

PAULO ARTUR PINTO DE OLIVEIRA LOPES

Análise de Cenas 3D com Visão e Laser

universidade de aveiro



DEPARTAMENTO DE FÍSICA

UNIVERSIDADE DE AVEIRO

1997

PAULO ARTUR PINTO DE OLIVEIRA LOPES

Análise de Cenas 3D com Visão e Laser

Dissertação apresentada à Universidade de Aveiro
para a obtenção do grau de Doutor em Física.

DEPARTAMENTO DE FÍSICA
UNIVERSIDADE DE AVEIRO

1997

*Aos meus Pais
e à minha Esposa*

AGRADECIMENTOS

Em primeiro lugar dirijo um agradecimento muito especial aos meus pais a quem muito devo e que sempre me serviram de referência ao longo da minha vida.

À minha esposa Ana Paula por todo o carinho, apoio psicológico, compreensão e ajuda incondicional que me prestou. Agradeço-lhe ainda a leitura integral desta Dissertação em ambas as fases da sua preparação com todo o trabalho volumoso de correcção que isso implicou.

Duma forma geral a toda a minha família e amigos que me deram incentivo.

Ao João Lemos Pinto pela amizade e pelo apoio pessoal em todo o processo de candidatura à defesa da minha Tese, e em particular durante o penoso folhetim da transferência da minha defesa de Tese do Porto para Aveiro.

Ao Carlos Ramos por ter acedido a ler e corrigir algumas partes da 1ª versão da dissertação na sua fase de preparação, e pela exaustiva e escrupulosa correcção que nela efectuou para a formulação da versão final, para a qual também contribuíram todos os seus conselhos e sugestões.

Gostaria também de agradecer a todos os amigos colegas do grupo de investigação do NIA&R (Núcleo de Inteligência Artificial & Robótica), ao Marcos, Carlos Ramos, Paulo Marinho, Conceição, Mouta, Benedita, Rui Camacho, Tozé, Feliz, Nuno, Alcínio e Ana Paula, pelo excelente espírito de companheirismo com que sempre me presentearam e também pelo excelente ambiente de trabalho que me proporcionaram.

Ao Isidro pelo permanente apoio material que me prestou e pela sua amizade.

Ao Carlos Ramos pela boa interacção que com ele tive quando ambos trabalhámos na área da Robótica.

Ao Jorge Alves pelas informações e sugestões dadas.

Aos elementos do meu Júri, Professores João de Lemos Pinto, Hélder Araújo, Carlos Ramos, João Sentieiro e Paulo Fiadeiro, que me enviaram sugestões preciosas para a elaboração da versão final da minha Dissertação.

Por fim, à instituição Faculdade de Engenharia da Universidade do Porto por me ter facultado os meios materiais necessários à execução do meu trabalho.

RESUMO

Neste trabalho procurou-se construir um Sistema de Visão completo, em regime de funcionamento autónomo, que permitisse analisar e estruturar o ambiente circundante de um robot de manipulação, por forma a que ele pudesse proceder a uma dada tarefa de montagem. Para que tal fosse possível foi desenvolvido todo um conjunto de programas que a partir da imagem adquirida por uma câmara conseguisse obter os dados fundamentais sobre aquele ambiente: o número de objectos visíveis, a sua localização e orientação no espaço e finalmente a sua identidade. Só assim seria possível estabelecer um plano, constituído por acções elementares de manipulação, que permitisse atingir a referida montagem final.

Para uma distribuição dispersa de objectos na cena, os primeiros dados referidos puderam ser obtidos por uma segmentação da imagem de intensidades binarizada e medida das características geométricas das figuras encontradas. A identificação foi também possível utilizando quer algumas das características medidas directamente quer grandezas compostas.

Em alternativa a este processo de identificação foi desenvolvida uma abordagem alterada da Transformada de Hough. Este novo processo que efectua a descrição de segmentos de fronteira dos objectos, em oposição à análise anterior da superfície, é um desenvolvimento melhorado da Transformada de Hough Clássica e é particularmente eficiente quando se possui imagens com ruído. Ela pode também ser utilizada em alguns casos de oclusão parcial, situação para a qual não pode ser utilizada o primeiro tipo de análise apresentado. Foi ainda incluído um algoritmo de pós-processamento que evita a determinação de um limiar de aceitação das células na tabela de acumulação assim como procedimentos de salientação de máximos. Uma vez conseguida uma lista completa de todos os segmentos existentes na imagem, eles são agrupados por objecto e sequenciados. Para cada conjunto são então determinados os vértices que permitem extrapolar os segmentos para o seu verdadeiro comprimento. São ainda calculados todos os ângulos internos formados pelos segmentos. Os valores de comprimento e ângulo alternados definem códigos que caracterizam os objectos. Quando comparados com códigos semelhantes referentes a modelos, registados em bases de dados, eles permitem obter a identidade dos objectos.

Quando ocorrem oclusões parciais ou mesmo totais, este processo pode não permitir obter a identidade de ambos os objectos, enquanto que o primeiro apresentado é totalmente inadequado. Para estes casos foi desenvolvido um outro processo a 3D que se baseia na análise dos histogramas 3D extraídos de imagens de distância. Para que tal fosse possível foi construído um sistema de varrimento laser que mede a altura a que vários pontos de cena se encontram da base de suporte, com várias resoluções. Para varrimentos na resolução máxima e utilizando vários filtros é então possível obter imagens de distância nas quais todos os pontos da imagem possuem uma altura associada que é codificada em nível de cinzento. Após uma segmentação primária é possível construir histogramas 3D, que, no caso dos objectos em estado de sobreposição obedecerem a certas condições, permitem a sua separação. Utilizando valores numéricos resultantes da parametrização daqueles histogramas 3D é ainda possível obter a identificação do objecto superior, após comparação com os registados previamente num ficheiro de modelos.

RÉSUMÉ

Cette Thèse cherche à construire un système de Vision complet, sous un régime de fonctionnement autonome, permettant d'analyser et structurer l'ambiance environnante d'un robot de manipulation pour l'accomplissement d'une tâche de montage. Afin que ça soit possible, tout un ensemble de programmes a été développé à partir de l'image obtenue par une caméra pour arriver à atteindre les données fondamentales concernant cette ambiance-là: le nombre d'objets visibles, leur localisation et orientation dans l'espace et finalement leur identité. À ce compte-là ce serait possible d'établir un planning, constitué par des actions élémentaires de manipulation, qui permettait d'accomplir la sus-dite montage finale.

Pour une distribution dispersive des objets en scène, les premières données citées ont été obtenues par une segmentation de l'image d'intensité rendue binaire et mesurée des caractéristiques géométriques des figures trouvées. L'identification a été aussi possible grâce à l'utilisation soit de quelques caractéristiques mesurées directement soit des grandeurs composées.

Une abordage modifiée de la Transformée de Hough a été développée comme alternative à ce procédé d'identification. Ce nouveau processus qui fait la description de segments de frontières des objets, en opposition à l'analyse précédente de la surface, c'est un développement perfectionné de la Transformée de Hough classique et il est particulièrement efficace quand on a des images avec du bruit. Elle peut être utilisée davantage dans certains cas d'occlusion partielle, situation dans laquelle on ne peut pas faire l'usage du premier genre d'analyse présenté. On a encore ajouté un algorithme de post-instruction d'un procès qui évite la détermination d'un seuil d'acceptation des cellules dans le tableau d'accumulation de même que des procédures de distinction de maximums. Ayant accomplie une liste complète de tous les segments existants dans l'image, ils son regroupés objet par objet et séquencés. Pour chaque ensemble, les sommets qui permettent d'extrapoler les segments pour leur longueur réelle sont alors déterminés. En outre, les angles internes formés par les segments sont encore chiffrés. Les valeurs de longueur et d'angle tour à tour déterminent des codes qui caractérisent les objets. Alors qu'elles sont comparées à des codes similaires concernant des modèles enregistrés dans des bases de données, elles permettent d'obtenir l'identité des objets.

Quand il y a des occlusions partiales ou même entières, ca procédé risque ne pas permettre d'atteindre l'identité de tous les deux tandis que le premier cas présenté en haut est inadéquat. Pour faire face à ces cas-là un autre processus à 3D a été développé, qui se fonde sur l'analyse des histogrammes 3D issues des images d'écartement. Afin que cela soit possible, on a construit un système de balayage laser qui mesure la hauteur de plusieurs points de scène par rapport à la plaque d'appui, ayant de différentes résolutions. Pour des balayages concernant une résolution au maximum, en se servant d'un ensemble de filtres, il est donc possible d'obtenir des images d'écartement dans lesquelles tous les points de l'image ont une hauteur associée qui est codifiée en niveau gris. Après une segmentation primaire il est possible de construire des histogrammes 3D qui permettent leur classement si les objets superposés sont soumis à certaines conditions. En utilisant des valeurs numériques, résultantes du paramétrage des histogrammes-là 3D, on peut obtenir encore l'identification de l'objet supérieur après la comparaison avec ceux qui ont été enregistrés préalablement dans un fichier de modèles.

ABSTRACT

In this work, a complete and autonomous Vision System has been built, which by itself analyses the surrounding environment of a robot, so that a certain predefined assembly task may take place. A complete set of computer programs have been written that, after processing an image acquired by a camera, manages to obtain some important data about that environment: the number of visible objects, their location and orientation in space, and finally, their identity. Only in possession of those results, a plan consisting of elementary robotic actions could be drawn, which led to a final assembly.

For images in which all the objects are separate, the above mentioned data were computed by segmenting the thresholded intensity image, achieved by scanning it and measuring geometrical features of all found blobs. Identification was also possible either by using those directly measured features or by using composite features.

Alternatively, another algorithm has been developed for identification, based on the Hough Transform. This new process that detects and measures all the segments present in the image, in opposition to the previous approach which analyses area, consists of an improved version of the classical Hough Transform and is particularly efficient in the case of noisy images. It may also be used in cases of object partial occlusion, situation in which the first analysis is useless. A post-processing algorithm has also been added to it which avoids choosing a threshold level for the accumulator array as well as peak enhancement procedures. Once a complete list of segments in the image is obtained, these are conveniently grouped and chained sequentially. For each group associated to an object, vertice coordinates are computed in order to extrapolate the segments to their real extent. Internal angles are also computed. The alternate values of length and angle define chain codes which after matching to similar chain codes associated to object models and previously stored in a models database, permit achieving the object identity.

Whenever partial or even complete occlusion occurs, this approach may not be able to recognise the objects, while the first one presented is completely inadequate. For those cases, then, another algorithm has been developed based on the 3D histograms built from range images. In order to get them, a special scanning laser system has been designed and built, which measures the height of points in the scene, for several resolutions. If the maximum resolution is used, a set of image filters can be applied in order to finally get the range image. In those images, every single point has got a height associated with it which is stored as a grey-level value. After a first segmentation, 3D histograms for all objects are built. When the objects is the situation of partial occlusion meet certain conditions, data belonging to each one of them can be separated. Under those conditions, numerical values obtained by parameterizing the 3D histograms, may be used for object identification: if similar values belonging to object models are previously stored in a models database, then, they can be matched with those obtained for each detected object.

ÍNDICE

Lista de Figuras	xxiii
------------------------	-------

INTRODUÇÃO

0.1 Motivação e Objectivos	1
0.2 Os Sistemas de Visão	2
0.3 Temas Abordados	6
0.4 Originalidades	9
0.5 Estrutura da Tese	10

Capítulo 1 A CALIBRAÇÃO DA CÂMARA

1.1 Introdução	13
1.2 A Conversão a 2D	15
1.3 A Conversão a 3D: O Fundamento Teórico	16
1.3.1 A Matriz Rotação	16
1.3.2 A Matriz Translação	18
1.3.3 As Coordenadas Homogéneas	18
1.3.4 A Transformação de Perspectiva	20
1.3.5 A Calibração da Câmara	23
1.3.6 Obtenção dos Elementos a_{ij} da Matriz de Transformação	24
1.3.7 Justificação do Resultado $\bar{x} = (A^T A)^{-1} A^T b$	27
1.3.8 Cálculo da Matriz Inversa de A	28
1.3.8.1 Influência da 3ª linha na Inversão de A	29
1.3.9 Obtenção de (X,Y) a partir de (x,y, Z) - 1º processo	30
1.3.9.1 Obtenção de (X,Y) a partir de (x,y, Z) usando cálculo matricial (2º processo)	30

1.4	A Montagem Experimental	32
1.5	Cálculo Final da Matriz A	34
1.6	Testes de Conversão com a Matriz A	35
1.7	Conclusão	38

Capítulo 2 APRESENTAÇÃO DO AGENTE VISÃO

2.1	Introdução	39
2.2	Explicação Breve sobre a Abordagem a Utilizar no Processo de Identificação a 2D	39
2.3	Descrição do Sistema do Agente Visão	41
2.4	Conclusão	45

Capítulo 3 A DETECÇÃO DE OBJECTOS NA IMAGEM E A MEDIDA DE PARÂMETROS GEOMÉTRICOS

3.1	Introdução	47
3.2	A Binarização das Imagens de Intensidade	47
3.3	A Construção dos "Blobs"	50
3.3.1	A Função de Leitura de uma Linha da Imagem	51
3.3.2	As Funções Detectoras de Transições $state_i()$	51
3.4	A Actualização dos Parâmetros Geométricos	56
3.4.1	Perímetro	57
3.4.2	Área	58
3.4.3	Somatório dos Valores de j (Σj)	58
3.4.4	Somatório dos Quadrados de j (Σj^2)	58
3.4.5	Somatório dos Valores de i (Σi)	58
3.4.6	Somatório dos Valores de $i \times j$ ($\Sigma i \times j$)	59
3.4.7	Somatório dos Valores de i^2 (Σi^2)	59
3.5	O Cálculo do Ângulo de Orientação	62
3.6	O Momento de Inércia Geométrico Relativamente ao CM	65
3.7	O Parâmetro Característico p_{sq_a}	67

3.8	A Selecção dos "Blobs" Válidos	68
3.9	Apresentação de Alguns Resultados	68
3.10	Conclusão	72

Capítulo 4 A IDENTIFICAÇÃO DE OBJECTOS A 2D USANDO CARACTERÍSTICAS GEOMÉTRICAS

4.1	Introdução	73
4.2	A Estrutura do Ficheiro de Modelos	73
4.3	A Consulta e Gestão do Conteúdo de 'Models.dat' ('R-Model.c')	75
4.3.1	A Consulta Genérica da Base de Dados ("lm")	75
4.3.2	A Consulta do Conteúdo de um dado Modelo ("gm i")	76
4.3.3	A Eliminação de Modelos ("rm i")	77
4.3.4	A Edição de um Modelo ("em i")	77
4.4	Construção do Ficheiro de Modelos: o programa 'M-Model.c'	78
4.5	A Fiabilidade dos Parâmetros Identificadores	79
4.6	A Estrutura de Funções de 'Identify.c' e a sua Execução	81
4.7	O Cálculo do Coeficiente de Comparação	84
4.8	Experiências de Identificação	86
4.8.1	Identificação da "balança" numa posição diferente à usada para registo de dados no Ficheiro de Modelos	87
4.8.2	Identificação do Objecto "nova_caixa"	89
4.8.3	Identificação do Objecto "fita_adesiva"	90
4.9	Conclusão	92

Capítulo 5 A TRANSFORMADA DE HOUGH MODIFICADA COM PÓS-PROCESSAMENTO

5.1	Introdução	95
5.2	A Descrição Matemática da Transformada de Hough	95
5.3	Os Primeiros Melhoramentos	98
5.3.1	O Registo Actualizado das Coordenadas dos Pontos Extremos	

de cada Segmento	98
5.3.2 A Escolha de Novos Parâmetros	98
5.3.3 Os Operadores de Sobel	100
5.4 Proposta de Novos Melhoramentos	101
5.5 Dificuldades Remanescentes na Selecção dos Máximos	104
5.6 Apresentação de Resultados Experimentais	105
5.7 Solução Encontrada (modificação da abordagem de Hough)	113
5.8 Proposta de um Algoritmo de Pós-Processamento	115
5.9 O Software Desenvolvido	117
5.10 Resultados Experimentais	118
5.10.1 Imagem nº 1	118
5.10.2 Imagem nº 2	121
5.11 Conclusão e Trabalho Futuro	125

Capítulo 6 A SEQUENCIAÇÃO DE SEGMENTOS CODIFICADA POR PARES COMPRIMENTO– –ÂNGULO

6.1 Introdução	127
6.2 O Agrupamento dos Segmentos	127
6.3 A Extrapolação dos Segmentos	129
6.4 Resultados Experimentais	133
6.4.1 1º Conjunto de Objectos	133
6.4.2 2ª Imagem: 1 Objecto	138
6.5 A Ligação entre Grupos de Segmentos	140
6.5.1 O Algoritmo Utilizado	140
6.5.2 Ilustração das Várias Fases de Ligação	141
6.6 A Identificação de Objectos	146
6.7 A Separação de Cadeias Pertencentes a Objectos diferentes	147
6.8 Conclusão	149

Capítulo 7 APRESENTAÇÃO DO AGENTE LASER

7.1	Introdução	151
7.2	Explicação Breve sobre a Abordagem a Utilizar no Processo de Identificação a 3D	151
7.3	Descrição do Programa Principal do Agente Laser	154
7.3.1	A Recepção de Pedidos e o Envio de Resultados	155
7.3.2	As Opções de Funcionamento do Agente Laser	157
7.4	Descrição dos Programas Envolvidos e Apresentação dos Capítulos Seguintes	160

Capítulo 8 O SISTEMA DE VARRIMENTO LASER PARA AQUISIÇÃO DE ALTURAS

8.1	Introdução	161
8.2	O Estado da Arte nos Sensores de Distância	162
8.2.1	Triangulação Simples	162
8.2.2	Triangulação por Visão Binocular (Visão Passiva)	163
8.2.3	Medição das Distâncias Usando Desfasamento	164
8.2.4	Medição da Distância por Tempo de Vôo	166
8.2.5	Medição da Distância com um "Array" Linear de Fotodíodos (Triangulação)	168
8.2.6	Medição da Distância através da Projecção de n Linhas Luminosas Paralelas (Triangulação)	170
8.2.7	Medição da Distância através de um Gradiente de Intensidade Luminosa	170
8.2.8	Medição da Distância através da Projecção de uma Linha Luminosa com Varrimento (Triangulação)	171
8.2.9	A Discriminação de Pontos por Variação do Plano de Observação da Imagem	171
8.2.10	Análise de Textura	172
8.3	O Díodo Laser TOLD 9200	172
8.3.1	Descrição do Funcionamento do Circuito de Alimentação	172
8.3.2	A Viabilidade do Uso do Díodo Laser TOLD 9200	176
8.4	O Sistema Óptico	176
8.4.1	A Luz Estruturada	176
8.4.2	O Sistema Óptico Implementado	177

8.4.3	A Geometria da Zona de Trabalho e suas Consequências	178
8.5	O Sistema Mecânico	179
8.5.1	O "Scanner" e a Opção do Motor Passo-a-passo	179
8.5.2	O Motor Passo-a-passo, a Caixa de Desmultiplicação e o Controlador SAA1027	180
8.5.3	O Circuito Associado ao Controlador; os Sinais de Saída e de Entrada	181
8.5.4	A Comunicação com o Computador: A Placa de I/O Digital de 24 bits, PIO-12	185
8.6	O Software de Controle	187
8.6.1	O Controle de Baixo Nível	187
8.6.2	A Obtenção de Imagens de Distância com Densidade de Pontos Constante	189
8.6.3	A Construção de Novos Comandos de Alto Nível	193
8.7	Resultados Experimentais	196
8.7.1	Testes de Posicionamento da Linha Laser	196
8.7.1.1	Varrimento com retorno; voltas completas	196
8.7.1.2	Varrimento e retorno, várias vezes	199
8.7.2	Uso das Funções de Deslocamento Horizontal Constante, Avaliação do Rigor de Posicionamento (sem dependência no caminho) e Repetibilidade	200
8.8	Conclusão	203

Capítulo 9 A AQUISIÇÃO DOS VALORES DE ALTURA

9.1	Introdução	205
9.2	A Obtenção da Linha Laser	205
9.2.1	A Binarização em Tempo Real	206
9.2.2	A Importância do Contraste: O Filtro Vermelho	207
9.2.3	A Filtragem por Software da Linha Laser	212
9.3	A Medida das Alturas Z	213
9.3.1	O Registo em Memória acima dos 64KB	215
9.3.2	O Registo em Disco do Vector de Estruturas	216
9.3.3	A Codificação das Alturas no "Display", por Cômputador	217
9.4	O Efeito de Perspectiva: A Correção da Fórmula de Z	219

9.5	A Distorção da Imagem e a Avaliação do Factor Variável <i>pixel_to_mm</i>	221
9.5.1	Tabela de Correspondências em $y=0$ e Derivadas Locais	228
9.6	A calibração do Sistema Laser: A Avaliação dos Parâmetros Mecânicos e Geométricos	233
9.6.1	Experiências de Calibração	235
9.6.2	Tabela Final de Resultados	243
9.6.3	Obtenção dos Parâmetros Geométricos e Mecânicos	243
9.6.4	Avaliação Aproximada dos Erros em Z	244
9.6.5	O Programa de Calibração	246
9.7	Apresentação de Amostragens de Altura com Várias Resoluções	247
9.8	A Recepção de Pedidos para Varrimento da Cena e os Tempos de Execução	250
9.9	A Utilização do Vector de Estruturas	251
9.9.1	O Algoritmo de Atribuição de Alturas a Pontos Aleatórios	252
9.9.2	O Algoritmo de Atribuição de Alturas a Pontos Pertencentes a Segmentos	252
9.9.3	A Função Comum <i>get_Z()</i>	253
9.9.4	O Efeito da Altura dos Objectos na Procura do Pixel (i,j)	256
9.10	A Imagem de Distâncias	258
9.10.1	Sequência de Aplicação dos Filtros num Caso Real	261
9.10.2	O Registo em Ficheiro da Imagem de Alturas	264
9.11	Conclusão	265

Capítulo 10 A PARAMETRIZAÇÃO DOS MODELOS 3D

10.1	Introdução	267
10.2	A Construção das "Assinaturas"	267
10.2.1	Poliedros Sobrepostos e nos seus Estados Estáveis	269
10.2.2	1º Poliedro em Sobreposição num Estado "Não Estável" e o 2º Sobreposto e Parcialmente Visível	271
10.2.3	A Rotação no Espaço do Objecto em Sobreposição	273
10.2.3.1	As Primitivas de Detecção Primária de Picos e Bandas	274
10.2.3.2	As 4 Funções de Maior Nível	274
10.2.4	Resumo da Codificação da Informação nos 3 casos: Ilustração Gráfica e Resultados	276

10.2.5	A Filtragem dos Histogramas: A Nova Abordagem e o Processo de Filtragem Associado	280
10.2.5.1	As Primeiras Versões do Filtro de Histogramas 3D	283
10.2.5.2	O Programa Interactivo para Determinação do Melhor Filtro de Histogramas 3D	283
10.2.6	Outros Programas Auxiliares: A Extracção de Valores para a Representação Bi e Tridimensional	291
10.3	A Fiabilidade dos Parâmetros Identificadores	293
10.3.1	Translação com um Poliedro Simples	294
10.3.2	Rotação com um Poliedro Simples	295
10.3.3	Poliedro Simples em Posições Aleatórias	296
10.4	O Ficheiro de Modelos 3D	297
10.5	Conclusão	298

Capítulo 11 A SEGMENTAÇÃO DAS IMAGENS DE ALTURA

11.1	Introdução	301
11.2	A Gestão da Segmentação	301
11.2.1	Localização do Objecto e Medição da Respectiva "Assinatura"	305
11.3	A Segmentação Primária da Imagem de Alturas	306
11.3.1	Descrição Breve do Programa de Varrimento	306
11.3.2	A Construção das Figuras ou Manchas	307
11.3.3	A Actualização dos Parâmetros Geométricos	307
11.3.4	As Funções de Manipulação de Histogramas 3D	309
11.4	O Programa de Sub-Segmentação	310
11.4.1	Apresentação do Programa	310
11.4.2	A Acção do Algoritmo	310
11.4.3	1º Exemplo: Caso Real de dois Poliedros Sobrepostos e nos seus Estados Estáveis	313
11.4.4	A Separação dos dois Objectos	314
11.4.5	Resultados Experimentais da Separação dos dois Objectos	315
11.4.6	2º Exemplo: Sobreposição de dois Poliedros com Rotação no Espaço do que se Encontra Visível	318
11.4.7	Resultados Experimentais da Separação dos dois Objectos	319
11.5	Conclusão	322

Capítulo 12	O ALGORITMO DE IDENTIFICAÇÃO A 3D	
12.1	Introdução	323
12.2	Descrição das Várias Funções Envolvidas	326
12.2.1	Leitura do Ficheiro em Disco	326
12.2.2	Obtenção da Informação do Modelo	326
12.2.3	Comparação da Informação do Modelo com a do Objecto	327
12.2.4	A Procura do Melhor Modelo	328
12.2.5	A Função Principal de Identificação a 3D	331
12.3	Descrição Esquemática do Programa de Identificação	331
12.4	Conclusão	332
	Conclusões e Trabalho Futuro	333
Anexo A		
A.1	O Método Usado na Inversão da Matriz de Transformação A	339
A.2	A Descrição do Software para Calibração da Câmara	340
A.2.1	Os Programas.....	340
A.2.2	Esquemas de Chamadas do Programa de Calibração.....	341
A.2.3	Descrição das Várias Funções	341
A.2.4	Descrição dos Programas de Teste "T-?.c".....	343
Anexo B	Esquema de Chamadas para as Funções Principais e Secundárias na Análise Bidimensional	
B.1	Esquema de Chamadas para as Funções Principais e Secundárias na Análise Bidimensional	345

Anexo C

C.1	O Significado das Funções no Programa	347
C.2	Demonstração do Teorema de Steiner para Massas Pontuais	348
C.3	Demonstração do Resultado $XY = \text{sumij}/\text{area} = \text{sumij}/\text{area} - \langle x \rangle \langle y \rangle$	349

Anexo D

D.1	Esquema de Chamadas no Software Desenvolvido para a Transformada de Hough Modificada com Pós-Processamento	351
D.2	Descrição das Funções	352

Anexo E

E.1	Fluxograma do Algoritmo de Agrupamento e Encadeamento de Segmentos	355
E.2	Fluxograma do Algoritmo de Extrapolação de Segmentos	356
E.3	Teste de Ligação entre 2 Grupos de Segmentos	357

Anexo F

F.1	Esquema Geral de Chamadas no Agente Laser	359
-----	---	-----

Anexo G

G.1	Estrutura de Dados no Vector de Objectos	361
G.2	Ficheiros de Comandos para FTP	364
G.3	O Esquema de Chamadas da Função de Filtragem	365

Anexo H

H.1	Esquema de Chamadas do Programa de Análise de "Assinaturas"	367
H.2	Fluxograma do Programa de Análise de "Assinaturas"	367
H.3	A Gestão do Ficheiro de Modelos 3D	369
H.3.1	A Linha de Comando	369
H.3.2	O Modo de Acesso aos Modelos	370
H.3.3	Funções Auxiliares	370
H.3.4	Funções Principais	372

Anexo I

I.1 a)	Descrição Esquemática do Programa Principal de Gestão do Agente Laser	387
I.1 b)	Esquema de Chamadas do Programa Principal de Gestão do Agente Laser	389
I.2 a)	Esquema de Chamadas do Programa de Segmentação a 3D	390
I.2 b)	O Significado das Funções no Programa de Segmentação a 3D	391
I.3 a)	Descrição Esquemática do Programa de Sub-Segmentação	393
I.3 b)	Esquema de Chamadas do Programa de Sub-Segmentação	392

Anexo J

J.1	Esquema de chamadas do Programa de Identificação a 3D	395
J.2	Descrição Esquemática da Função de Procura do Melhor Modelo	396

Referências	399
-------------------	-----

LISTA DE FIGURAS

Capítulo 1

Fig. 1	Grelha de pontos escolhidos para constarem do ficheiro e conversão	15
Fig. 2	Cálculo das coordenadas X e Y do ponto consoante a sua proximidade a cada um dos 4 pontos delimitadores	15
Fig. 3	A rotação de um sistema de referência e a obtenção das novas coordenadas	17
Fig. 4	A translação de um sistema de referência e a obtenção das novas coordenadas	18
Fig. 5	O conjunto de matrizes que permitem a transformação de coordenadas	20
Fig. 6	Mapeamento dos pontos (X,Y,Z) no plano xy	21
Fig. 7	Projecção do vector 2º membro no sub-espaco das colunas da matriz de 3×2	28
Fig. 8	Desenho da grelha utilizada com indicação dos limites quer da acção do robot quer da imagem	33
Fig. 9	Imagem da grelha numa das suas posições de calibração	33
Fig. 10	Experiência de calibração da câmara: a fig. mostra as várias posições da grelha	34
Fig. 11	Sobreposição dos pontos de referência com os obtidos por conversão	36
Fig. 12	Sobreposição dos pontos de referência com os obtidos por conversão	36

Capítulo 2

Fig. 1	Apresentação de todos os estados estáveis de um paralelepípedo	40
Fig. 2	Objecto com alguma complexidade utilizado em experiências de identificação	40
Fig. 3	Imagem binarizada de um conjunto de objectos, a maior parte dos quais de uso corrente	41
Fig. 4	Esquema de transferência de ficheiros de e para a zona de acesso comum	42

Capítulo 3

Fig. 1	Histograma de uma imagem de intensidade no qual é visível o registo dos pixels do fundo em torno do nível ~ 49	48
Fig. 2	O histograma ampliado para melhor visualização do registo referente aos pixels dos objectos com níveis de cinzento superior a ~ 70	49
Fig. 3	Resultado da suavização do histograma anterior, no qual são bem visíveis as duas elevações correspondentes ao fundo e aos objectos	49
Fig. 4	Ampliação do histograma suavizado	50
Fig. 5	O novo conjunto de pixels consecutivos pertencem afinal a um outro "blob"	54
Fig. 6	O "blob" associado ao pixel d foi completamente circundado	55
Fig. 7	A lista completa dos buracos do objecto poderá ser obtida seguindo	

	os conteúdos dos vários campos consecutivos do 2º tipo	55
Fig. 8	Se o nº do "blob" associado ao pixel c for diferente do de b , então ambos os "blobs" etiquetados como 1 e 2 devem ser fundidos	56
Fig. 9	Para cada estado existe sempre um outro, obtido da troca dos pixeis 1 e 0	56
Fig. 10	O acréscimo de mais uma linha à figura já existente, incrementaria o perímetro, em $\sqrt{2}$.	58
Fig. 11	Desvio do ponto experimental relativamente à recta ideal, medido na vertical	62
Fig. 12	Objecto sujeito à análise geométrica	68
Fig. 13	1.ªs resultados da análise geométrica registados em disco	69
Fig. 14	Registo de informação seleccionada onde se evidenciam as relações de parentesco entre "blobs"	70
Fig. 15	Desenho no monitor dos centros de massa e de linhas que definem a orientação dos vários objectos	71

Capítulo 4

Fig. 1	Registo sequencial e em modo binário da informação dos modelos	74
Fig. 2	Estrutura de cada registo no ficheiro de modelos	74
Fig. 3	Efeito da rotação do objecto na medida do perímetro	79
Fig. 4	Variação do factor que afecta o perímetro em função do ângulo de rotação α	80
Fig. 5	Formato e medida do pixel na imagem	81
Fig. 6	Algoritmo de identificação	82
Fig. 7	Variação do coeficiente de comparação c_i com a razão entre δ_i e σ_i	86
Fig. 8	Objectos acrescentados ao ficheiro de modelos e usados nas experiências de identificação	86
Fig. 9	Objecto de teste nº 1: "balança"	87
Fig. 10	Objecto de teste nº 2: "nova_caixa"	89
Fig. 11	Objecto de teste nº 3: "fita_adesiva"	91

Capítulo 5

Fig. 1	A representação de uma recta no espaço dos parâmetros	96
Fig. 2	A representação de um ponto no espaço dos parâmetros	96
Fig. 3	O "ponto" de intersecção das várias "rectas" no espaço dos parâmetros	97
Fig. 4	Ilustração de uma possível tabela de acumulação na forma de histograma bidimensional	97
Fig. 5	Obtenção dos novos parâmetros (ρ, θ) de uma recta	98
Fig. 6	Exemplo de uma tabela de acumulação no caso da abordagem clássica	99
Fig. 7	Os operadores de Sobel utilizados na nova abordagem	100
Fig. 8	Esquematisação da imagem sujeita à aplicação da Transformada de Hough com os operadores de Sobel	101

Fig. 9	Tabela de acumulação obtida por aplicação dos operadores de Sobel	102
Fig. 10	O uso das novas máscaras de 5×5 permitem obter valores de (ρ, θ) mais correctos	101
Fig. 11	Justificação do efeito de espalhamento dos incrementos ao longo de ρ	103
Fig. 12	Máscaras de 5×5 adoptadas para substituir os operadores de Sobel	103
Fig. 13	Novas máscaras para detecção de fronteira em imagens binarizadas	104
Fig. 14	Espalhamento dos incrementos ao longo de ρ	105
Fig. 15	Tabela de acumulação com 3 células incrementadas	106
Fig. 16	A sobreposição de aglomerados	107
Fig. 17	A escolha de 1 célula por aglomerado	108
Fig. 18	A escolha de um limiar de incrementos	109
Fig. 19	A escolha de 1 célula por aglomerado	110
Fig. 20	A sobreposição parcial de 2 aglomerados	111
Fig. 21	A sobreposição parcial de 2 aglomerados	112
Fig. 22	Presença de um 2º segmento com o mesmo (ρ, θ)	113
Fig. 23	Presença de ruído que poderá originar incrementos na mesma célula	113
Fig. 24	Situação de quase colinearidade e também quase horizontalidade	114
Fig. 25	Representação da tabela de acumulação de onde emergem as sub-células	114
Fig. 26	Fusão de sub-células na lista ligada	115
Fig. 27	Partes de um segmento a serem fundidos	115
Fig. 28	2º teste de colinearidade dos segmentos	116
Fig. 29	Verificação da distância entre dois segmentos colineares	117
Fig. 30	Imagem binarizada do conjunto de objectos	118
Fig. 31	Pontos de fronteira sujeitos à Transformada de Hough	119
Fig. 32	Segmentos seleccionados da tabela de acumulação	119
Fig. 33	Segmentos resultantes do algoritmo de fusão	120
Fig. 34	Imagem final dos segmentos reconstruídos	120
Fig. 35	Imagem binarizada do conjunto de objectos	121
Fig. 36	Pontos de fronteira sujeitos à Transformada de Hough	122
Fig. 37	Segmentos seleccionados da tabela de acumulação	122
Fig. 38	Segmentos resultantes do algoritmo de fusão	123
Fig. 39	Imagem final dos segmentos reconstruídos	123
Fig. 40	Registo textual da lista final dos segmentos	124

Capítulo 6

Fig. 1	Verificação da proximidade do 2º segmento: sucesso e rejeição	128
Fig. 2	Actualização das coordenadas do fim da cadeia de segmentos	128
Fig. 3	Determinação dos vértices	129

Fig. 4	Determinação de θ em função de θ_1	130
Fig. 5	Determinação de θ em função de θ_1	131
Fig. 6	Determinação de θ em função de θ_1	131
Fig. 7	Pontos de fronteira pertencentes a 4 objectos de teste	134
Fig. 8	Segmentos após o algoritmo de fusão	134
Fig. 9	Segmentos após o algoritmo de fusão, numerados e orientados	135
Fig. 10	Segmentos após o algoritmo de extrapolação	137
Fig. 11	Pontos de fronteira pertencentes a 4 objectos de teste	138
Fig. 12	Resultado após a fusão e eliminação de segmentos pequenos	139
Fig. 13	Encadeamento final dos segmentos	140
Fig. 14	Descrição do programa principal de ligação entre grupos	141
Fig. 15	Resultado após a 1ª iteração	142
Fig. 16	Resultado após a 2ª iteração	143
Fig. 17	Resultado após a 3ª e última iteração	144
Fig. 18	Apresentação final dos segmentos após a ligação ente grupos e a extrapolação	145
Fig. 19	Objecto sujeito ao processo de identificação	146
Fig. 20	Esquema geral de comparação de cada cadeia obtida da imagem com qualquer modelo da base de dados	148
Fig. 21	Análise de cadeias para um caso simples de sobreposição parcial	148

Capítulo 7

Fig. 1	Exemplo de um histograma de alturas de duas faces planas paralelas ao plano horizontal	152
Fig. 2	Apresentação de todos os estados estáveis de um paralelepípedo	153
Fig. 3	Sobreposição parcial de dois objectos poliédricos nos seus estados estáveis	154
Fig. 4	Objecto num dos seus estados estáveis	154
Fig. 5	Dois objectos em estado de sobreposição parcial, encontrando-se o objecto superior numa posição considerada como não estável	154
Fig. 6	Exemplo de envio de um pedido pelo Agente Visão	156
Fig. 7	Ficheiros de comandos destinados ao programa FTP	157
Fig. 8	O envio dos resultados após execução do pedido	157

Capítulo 8

Fig. 1	Triangulação simples	162
Fig. 2	Visão stereo com 2 câmaras	163
Fig. 3	Medição do desfasamento entre 2 ondas	164
Fig. 4	Modo de funcionamento do sistema óptico	168

Fig. 5	Esquema do sistema mecânico responsável pelo varrimento do feixe	168
Fig. 6	Esquema do alimentador e estabilizador do díodo laser	174
Fig. 7	Rectificação de onda completa na alimentação do díodo laser	173
Fig. 8	Montagem final da fonte de alimentação do díodo laser	175
Fig. 9	Montagem robusta do díodo laser na placa dissipadora	175
Fig. 10	Desenho esquemático do sistema óptico implementado	177
Fig. 11	O dispositivo responsável pelo varrimento ("scanner"); perspectiva do motor passo-a-passo	178
Fig. 12	O dispositivo responsável pelo varrimento ("scanner"); perspectiva da lente cilíndrica e do espelho rotativo	178
Fig. 13	A câmara utilizada e o respectivo suporte orientável	179
Fig. 14	Espelho orientável em 3D	179
Fig. 15	Esquema de ligações do controlador SAA1027	182
Fig. 16	Esquema e "pin-out" de um acoplador óptico 4N33	181
Fig. 17	Esquema de ligações dos 4 c.i. 4N33 utilizados	184
Fig. 18	Fotografia dos acoplamentos ópticos	181
Fig. 19	Comparador com um 741 para filtrar o sinal de "clock"	183
Fig. 20	Os contadores 4029, os conversores bin-7segm e os "displays" respectivos	186
Fig. 21	Os contadores 4029, os conversores bin-7segm e os "displays" respectivos	185
Fig. 22	A montagem final de todo o circuito de controle	185
Fig. 23	O esquema de ligações da tomada de 25 pinos da placa PIO-12	187
Fig. 24	Esquema completo de circuitos e ligações entre o computador e o motor passo-a-passo	188
Fig. 25	Estrutura hierárquica de funções de controle	190
Fig. 26	Deslocamento da linha laser em dx como resultado da rotação $d\theta$ do espelho	191
Fig. 27	O valor do deslocamento dx como resultado da rotação $d\theta$ do espelho depende do ângulo de incidência θ	191
Fig. 28	Obtenção do nº de impulsos a enviar ao motor	194
Fig. 29	Obtenção do nº total de impulsos a enviar ao motor	195
Fig. 30	A nova função <i>shifts()</i> calcula o nº de impulsos a enviar ao motor	196

Capítulo 9

Fig. 1	Imagem binarizada da cena que inclui a linha laser projectada	208
Fig. 2	Histograma da imagem sem uso do filtro vermelho	209
Fig. 3	Uma ampliação do histograma	209
Fig. 4	Aquisição binarizada da imagem quando o filtro não é usado	210
Fig. 5	Histograma da imagem quando o filtro vermelho é usado	210
Fig. 6	Histograma ampliado com uso do filtro vermelho	211

Fig. 7	Aquisição binarizada da imagem quando o filtro é usado	211
Fig. 8	Sequência de filtragem da linha laser	213
Fig. 9	Avaliação da altura Z através da medida de <i>shift</i>	213
Fig. 10	Indicação das posições relativas entre o espelho rotativo e o espaço de trabalho	214
Fig. 11	A amostragem de alturas Z	215
Fig. 12	Registo das estruturas associadas aos Z s sequencialmente em memória RAM	215
Fig. 13	Registo em sequência dos conjuntos de estruturas associados a cada linha laser, na memória RAM	215
Fig. 14	Registo sequencial das estruturas com uso de memória acima dos 64KB	216
Fig. 15	Extracção do Segmento e Offset de um apontador	216
Fig. 16	Avaliação (interna) de um endereço a partir do Segmento e do Offset	216
Fig. 17	Gráficos de conversão da LUT de saída 6 para as 3 cores	219
Fig. 18	A incorrecção na medida de Z devido ao efeito de perspectiva	220
Fig. 19	Crescimento de mm e $pixel$ com o n° de medidas	222
Fig. 20	Representação de mm em função de $x \equiv pixel$	222
Fig. 21	Representação das derivadas locais para uma vizinhança de cálculo de 70mm	224
Fig. 22	Representação das derivadas locais para uma vizinhança de cálculo de 150mm	224
Fig. 23	Simetrização do gráfico em torno de 385mm (vizinhança de 70mm)	225
Fig. 24	Simetrização do gráfico em torno de 385mm (vizinhança de 150mm)	225
Fig. 25	Ajuste da melhor função aos pontos experimentais	227
Fig. 26	Gráfico das derivadas locais simetrizadas com ampliação (vizinhança de 150mm), para $y=0$	230
Fig. 27	Curva de ajuste para $pixel_to_mm$ referente à linha superior da imagem ($y=0$)	230
Fig. 28	Função teórica para $pixel_to_mm$ (com $y_factor=1,194$)	231
Fig. 29	Representação de $pixel_to_mm$ como função de x e y	232
Fig. 30	Representação de Y em função de X	236
Fig. 31	Representação de Y em função de X	237
Fig. 32	Representação de Y em função de X	239
Fig. 33	Representação de Y em função de X	240
Fig. 34	Representação de Y em função de X	241
Fig. 35	Representação de Y em função de X	243
Fig. 36	Perspectiva tridimensional dos objectos presentes na cena	248
Fig. 37	Representação do vector de estruturas a 2D	248
Fig. 38	Amostragem das alturas com $res=3$	249
Fig. 39	Amostragem das alturas com $res=2$	249
Fig. 40	Amostragem das alturas com $res=1$	249
Fig. 41	Situação real que requer a intervenção do algoritmo de atribuição de alturas	252
Fig. 42	Numa imagem de alturas todos os pixels possuem uma altura associada	253

Fig. 43	Exibição em monitor do ficheiro de amostragem de alturas	254
Fig. 44	Procura do canto inferior esquerdo do "quadrado" delimitador do CM	255
Fig. 45	Avaliação de Z a associar ao CM dado o "quadrado" delimitador	256
Fig. 46	Varrimento laser para um objecto baixo	257
Fig. 47	Varrimento laser para um objecto alto	257
Fig. 48	Representação tridimensional dos objectos	261
Fig. 49	Figura original resultante de um varrimento laser com $res=1$	262
Fig. 50	Figura obtida após a aplicação do filtro referido em d)	262
Fig. 51	Figura obtida após a aplicação do filtro referido em c)	263
Fig. 52	Figura obtida após a aplicação do filtro referido em b)	263
Fig. 53	Figura final obtida após a aplicação do filtro referido em a)	264
Fig. 54	Registo em disco de apenas 5 dos 8 blocos do ficheiro de imagem	265

Capítulo 10

Fig. 1	Medição dos parâmetros relativos às faces planas paralelas ao plano horizontal	269
Fig. 2	Representação tridimensional de dois objectos em situação de sobreposição parcial ...	270
Fig. 3	Histograma 3D respeitante à situação da figura anterior	270
Fig. 4	Representação tridimensional de dois objectos poliédricos	272
Fig. 5	Histograma 3D referente ao caso da figura anterior	272
Fig. 6	Parametrização do histograma 3D e registo na "Assinatura" dos valores referentes a 2 picos	277
Fig. 7	Parametrização do histograma 3D e registo na "Assinatura" dos valores referentes a 1 faixa	278
Fig. 8	Parametrização do histograma 3D e registo na "Assinatura" dos valores referentes a 1 pico + 1 faixa	279
Fig. 9	Histograma 3D não filtrado (2 picos)	281
Fig. 10	Histograma 3D não filtrado (1 faixa)	281
Fig. 11	Histograma suavizado referente a 1 faixa	282
Fig. 12	Histograma 3D referente a 1 pico + 1 faixa, não filtrado e filtrado	282
Fig. 13	Ilustração comparativa do resultado de 3 filtragens diferentes para 2 picos	286
Fig. 14	Ilustração comparativa do resultado de 4 filtragens diferentes para 1 faixa	288
Fig. 15	Ilustração comparativa do resultado de 4 filtragens diferentes para 1 pico + 1 faixa	290
Fig. 16	Representação bidimensional de uma imagem de alturas	292
Fig. 17	Representação bidimensional de uma amostragem de alturas na resolução máxima	292
Fig. 18	Representação tridimensional em duas perspectivas de uma situação de sobreposição .	293
Fig. 19	Representação bidimensional de uma amostragem de alturas na resolução máxima	293
Fig. 20	Registo em modo binário com comprimentos variáveis para cada modelo	298
Fig. 21	Estrutura interna de cada modelo 3D	298

Capítulo 11

Fig. 1	Máscaras de convolução referentes aos dois operadores de Sobel	303
Fig. 2	Os valores de posição e orientação a registar	304
Fig. 3	Separação da informação referente aos dois objectos	311
Fig. 4	Caso simples de uma sobreposição de dois poliedros	313
Fig. 5	Imagem de alturas referente a dois poliedros em sobreposição	313
Fig. 6	Representação tridimensional dos mesmos objectos	314
Fig. 7	Vista dos objectos pela câmara	314
Fig. 8	O novo varrimento ordenado para a sub-segmentação contempla apenas o objecto assinalado que está associado ao 2º pico do histograma 3D	315
Fig. 9	Esta imagem de alturas representa apenas o objecto visível	315
Fig. 10	Representação tridimensional do objecto completamente visível	316
Fig. 11	Esta imagem de alturas representa apenas o objecto parcialmente oculto	316
Fig. 12	Representação tridimensional do objecto parcialmente oculto	316
Fig. 13	Registo em texto dos dados relativos a ambos os objectos	317
Fig. 14	Imagem de alturas referente a dois poliedros em sobreposição	318
Fig. 15	Representação tridimensional dos mesmos objectos	318
Fig. 16	Esta imagem de alturas representa apenas o objecto visível	319
Fig. 17	Representação tridimensional do objecto completamente visível	319
Fig. 18	Esta imagem de alturas representa apenas o objecto parcialmente oculto	320
Fig. 19	Representação tridimensional do objecto parcialmente oculto	320
Fig. 20	Registo em texto dos dados relativos a ambos os objectos	321

Capítulo 12

Fig. 1	Representação da distância Euclidiana no espaço multidimensional dos parâmetros identificadores	323
Fig. 2	Quadro de resultados experimentais para a tolerância dos parâmetros	325
Fig. 3	Ciclo de procura dos dois melhores modelos	329
Fig. 4	Procedimento quando é encontrado um modelo melhor que o nº1	330
Fig. 5	Procedimento quando é encontrado um modelo melhor que o nº2	330
Fig. 6	Registo final dos dois melhores modelos encontrados na base de dados	332
Fig. 7	Descrição esquemática da função principal de identificação a 3D	331

Anexo A

Fig. 1	Esquema de chamadas do programa de calibração	341
--------	---	-----

Anexo B

Fig. 1	Esquema de chamadas no Agente visão	345
--------	---	-----

Anexo C

Fig. 1	Momentos de inércia de uma massa discreta m_i relativamente a dois eixos paralelos, calculados a partir dos respectivos vectores de posição: r_i e r_{io}	348
Fig. 2	Representação do sistema de referência centrado no centro de massa do objecto	349

Anexo D

Fig. 1	Esquema geral de chamadas no algoritmo de Hough modificado com pós-processamento	351
--------	--	-----

Anexo E

Fig. 1	Descrição do algoritmo de encadeamento de segmentos	355
Fig. 2	Descrição do algoritmo de extrapolação de segmentos	356
Fig. 3	Descrição da função <i>one_iteration()</i> contida pelo algoritmo de ligação entre grupos	357

Anexo F

Fig. 1	Esquema de chamadas a programas e funções no Agente Laser	359
--------	---	-----

Anexo G

Fig. 1	Estrutura de cada elemento de entrada do vector de objectos registado em 'objects.dat'	361
Fig. 2	Envio de pedido pelo Agente Visão assim como do ficheiro de objectos para a área de acesso comum e recepção destes pelo Agente Laser	361
Fig. 3	Envio de resultados do Agente Laser para a área de acesso comum e recepção destes pelo Agente Visão	364
Fig. 4	Estrutura de cada elemento de entrada do vector de posições registado em <i>fCoord.dat</i>	365
Fig. 5	Esquema de chamadas dos vários filtros para a obtenção da imagem de alturas	365

Anexo H

Fig. 1	Esquema de chamadas do programa de análise de "Assinaturas", 'Signatur.c'	367
Fig. 2	Fluxograma do programa de análise de "Assinaturas"	368
Fig. 3	Cópia integral do ficheiro de modelos 3D para memória RAM	370
Fig. 4	Listagem dos nomes de todos os modelos da base de dados	373
Fig. 5	Localização dos apontadores quando da remoção de um modelo da base de dados ...	374
Fig. 6	Remoção do modelo da base de dados	374
Fig. 7	Actualização em disco do ficheiro de modelos	374
Fig. 8	A estrutura do registo de um modelo no ficheiro e a sua leitura para variáveis conhecidas	377
Fig. 9	A edição de um modelo	378
Fig. 10	Representação dos vários estados estáveis de um poliedro	383

Anexo I

Fig. 1	Descrição esquemática do programa de gestão do Agente Laser	388
Fig. 2	Esquema de chamadas do programa de gestão do Agente Laser	389
Fig. 3	Esquema de chamadas do programa de segmentação a 3D	390
Fig. 4	Descrição esquemática do programa de sub-segmentação	393
Fig. 5	Esquema de chamadas do programa de sub-segmentação	392

Anexo J

Fig. 1	Esquema de chamadas do programa de identificação a 3D	395
Fig. 2	Descrição esquemática da função de procura do melhor modelo 3D	396

Nesta dissertação foram utilizados inúmeros termos ingleses, de uso corrente na gíria informática e que foram colocados entre aspas. O seu emprego deveu-se em alguns casos à inexistência de equivalentes na língua portuguesa e noutros ao seu uso generalizado e que possuem significados bastante específicos. A sua tradução forçada fá-los-ia perder esse significado específico pelo que optei pelo uso da terminologia original.

INTRODUÇÃO

1. Motivação e Objectivos

O desenvolvimento de todo este sistema de visão veio na sequência dos meus trabalhos iniciais na área da robótica, que consistiram na programação de acções de manipulação automática em robótica de manipulação, e cujo conteúdo não se encontra apresentado nesta dissertação.

O Sistema de Visão é suposto funcionar como ligação entre o mundo real constituído por um ambiente robótico não completamente estruturado e um planeador de acções de alto nível (que foi desenvolvido no âmbito de outra Tese de Doutoramento). Mais concretamente, conhecendo-se uma determinada montagem final que constitui o objectivo de uma célula robótica de manipulação, há que conhecer com rigor o posicionamento de cada uma das partes dessa montagem dada a sua posição e orientação tridimensional na imagem dada por uma câmara, assim como a correcta atribuição de identidades. Só desse modo aquele planeador conseguirá traçar um plano de acção, composto por várias acções discretas de manuseamento robótico das peças, por forma a atingir o estado final pretendido.

O sistema que desenvolvi e que funciona de um modo completamente automático e autónomo (pretende-se que não haja intervenção externa por parte do utilizador), permite precisamente atingir aqueles objectivos.

Todo o software desenvolvido destina-se a ser corrido em dois computadores pessoais (PC) separados, ou alternativamente em duas janelas de um ambiente Windows num mesmo PC. Esta separação que irá dar origem a dois sistemas funcionando em paralelo, vai de encontro à filosofia adoptada de distribuição da carga computacional por vários agentes assim como de especialização de tarefas. De facto, para além do sistema que recebe a codificação das acções discretas a efectuar em termos para ele compreensíveis, e que constitui o agente Robot instalado também num PC, foram também criados numa estação de trabalho vários processos Unix e atribuídos aos vários agentes especializados. Todos eles formam uma comunidade de agentes que trocam informação entre si, enviando pedidos de execução e recebendo resultados. Dos agentes nela instalados e que foram desenvolvidos na Tese acima referida, destaca-se o planeador de alto nível (HLP - "High Level Planner") cuja tarefa é a elaboração do plano de acções intercalares que levam à montagem final, conhecido o "estado do mundo"

definido pelo sistema de Visão apresentado nesta dissertação, e o executor de baixo nível (LLE - "Low Level Executer") que traduz aquelas acções em comandos compreensíveis pela linguagem de programação utilizada no agente Robot, com os respectivos parâmetros numéricos.

Os dois conjuntos de software desenvolvidos nesta Tese irão constituir os Agentes Visão e Laser. O primeiro efectua análises a 2D enquanto que o segundo faz uso de um sistema de varrimento laser para obter medidas de altura e portanto analisa a cena robótica a 3D. Estes dois agentes funcionam separadamente e em paralelo, possuindo a restrição única de não poderem aceder simultaneamente à placa de aquisição de imagens. Embora eles possam funcionar em regime de complementaridade, no qual o Agente Laser fornece medidas de altura para complementar os algoritmos executados no Agente Visão, eles podem funcionar em regime concorrencial. Ou seja, ambos efectuam o mesmo tipo de análise, sendo escolhido o resultado daquele que o apresenta primeiro, ou que apresenta maior credibilidade em função dos condicionalismos particulares detectados na cena.

2. Os Sistemas de Visão

Por forma a poder-se ter uma atitude crítica em relação ao sistema aqui desenvolvido, será útil a apresentação de outros sistemas de visão [Fan89] que constituam o "estado da arte" em processos de identificação (embora este sistema execute muitas outras acções, formando um "pacote" completo de software).

Em [Horn84a] [Horn84b] [Ikeuchi81] são utilizados modelos EGI ("Extended Gaussian Image") em multi-perspectiva para reconhecer objectos a 3D. Cada modelo é representado através do seu mapeamento na esfera Gaussiana e cada objecto da cena é também representado por um EGI. Os EGIs dos objectos da cena são então comparados com os das várias perspectivas do modelo. Para restringir o domínio de procura, os dois EGIs são primeiramente alinhados ao longo das direcções de valor mínimo para a massa inercial. Então é calculada uma medida de sobreposição através da comparação de similaridade das respectivas distribuições de massa. O modelo que maximiza aquela medida é escolhido como melhor modelo.

Este método tem a vantagem do EGI poder ser calculado directamente sem qualquer fase complexa de descrição. A principal desvantagem reside no facto do EGI ser sensível à oclusão sendo única apenas para objectos convexos. Para além disso, quando a cena contém múltiplos objectos, é necessário segmentar o EGI em várias regiões cada uma correspondente a um objecto, tarefa que não é fácil de executar excepto para objectos de forma muito simples.

Outros trabalhos baseados na representação do modelo através de multi-perspectiva bidimensional poderão ser encontrados em [Korn87], [Lowe87], [Lamdan88], [Dhome89], [Breuel90], [Clemens91], [Stein91], [Flynn92] e [Roux94].

Em [Boshra96] é apresentada uma técnica para verificação de hipóteses relativas à identidade dos objectos 3D presentes numa cena, a partir da imagem numa única perspectiva. Em primeiro lugar, é construída uma imagem do modelo a experimentar

constituída pelas suas arestas visíveis. A imagem assim construída é então sobreposta à imagem da cena após a detecção das arestas nela contidas. A validade da hipótese colocada é avaliada pela correcta correspondência dos pixeis nas duas imagens. A incerteza em estimar a localização dos pixeis das arestas tanto do objecto como do modelo é avaliada após dilatação das arestas na imagem da cena. É ainda apresentado um método analítico para determinar a extensão dessa dilatação, assumindo um certo majorante para o erro na posição e orientação do objecto.

Em [Horaud84] [Bolles86] é apresentado o sistema 3DPO para reconhecimento e localização tridimensional em imagens de altura ("range images"). O modelo consiste em duas partes: um modelo CAD aumentado e uma rede de classificação de características. O modelo CAD descreve arestas, superfícies, vértices e outras relações geométricas. A rede de classificação de características classifica as diversas características observáveis por tipo e tamanho. O sistema reconhece objectos desconhecidos através da procura de características que se justaponham às de algum modelo; por exemplo, uma curva cilíndrica com um dado raio. São então lançadas hipóteses sobre a identidade dos objectos determinando se um par de segmentos observados é consistente com uma dada característica do modelo. Este sistema mostrou-se eficiente no reconhecimento de objectos em cenas bastante complexas, mas com um número bastante reduzido de modelos. Uma vez que os modelos são representados a 3 dimensões, torna-se muito difícil criá-los automaticamente pelo que se faz uso de um modelo CAD que requer a intervenção de um utilizador. Para além disso este sistema baseia-se fortemente na detecção de arcos de circunferência e arestas, pelo que o tipo de formas que ele consegue reconhecer é bastante restrito.

Em [Grimson84] é discutida a forma como medidas locais de posições 3D e de normais à superfície podem ser utilizadas para identificar e localizar objectos dentro de um conjunto de objectos desconhecidos. Os modelos consistem em objectos poliédricos representados pelas suas faces planas. A informação sobre essas superfícies (tais como as suas equações) e as relações entre faces (como a distância) são também calculadas. Informação de distância ou táctil, embora pouco densa, de objectos 3D é utilizada como características da cena. O processo de comparação é composto por dois passos. No primeiro é feita uma interpretação dos dados sensoriais obtidos. As interpretações consistem no emparelhamento de cada ponto obtido do sensor com uma superfície de um dos modelos. Aquelas que forem inconsistentes com restrições locais são ignoradas. No segundo passo, as interpretações aceitáveis são verificadas através de um teste de transformação. Uma dada interpretação é aceite se puder ser usada numa transformação que coloque cada ponto obtido sensorialmente numa superfície do modelo. Só objectos poliédricos ou objectos com um número suficiente de superfícies planas podem ser usados neste sistema.

Em [Ikeuchi87] é desenvolvido um método para reconhecimento de objectos em tarefas de "bin-picking" (apanha de objectos de uma amálgama). Os modelos dos objectos são gerados de acordo com várias direcções de observação, após o que as formas são classificadas em grupos. Uma vez que este sistema foi desenvolvido para tarefas de "bin-picking", um único tipo de objecto, que coincide com o do modelo, aparece na cena. As mesmas características de superfície usadas nos modelos são extraídas e classificadas com ajuda do modelo. No processo de reconhecimento uma

"árvore de interpretação" é gerada de acordo com as várias perspectivas do modelo. A orientação e localização do objecto na cena é então decidida comparando as suas características de superfície e classificada pela "árvore de interpretação". Este não é exactamente um sistema de reconhecimento mas antes um sistema que identifica um dado objecto a 3D de modo a ser recolhido do conjunto.

Em [Faugeras86] é desenvolvido um sistema de reconhecimento e localização de objectos no espaço tridimensional. Os modelos são representados em termos de características tais como pontos, linhas e planos. As mesmas características tais como pontos, linhas e planos significativos, são utilizadas para descrever objectos numa cena. Este sistema usa restrições para guiar o processo de comparação em primeiro lugar, e possíveis emparelhamentos entre características do modelo e do objecto são estabelecidos. Depois, justaposições posteriores são previstas e verificadas por aquelas restrições. Este sistema mostrou-se capaz de reconhecer objectos bastante complexos, mas na condição de não ocorrerem oclusões. Uma vez que a segmentação de objectos resulta num largo número de superfícies elementares e que esse processo não é guiado, é de prever que ele seja criticamente sensível a pequenas modificações nas superfícies.

Em [Oshima83] é apresentado um sistema de reconhecimento para objectos com superfícies planas e curvas. Cada modelo é representado por um grafo contendo características cujos nodos representam superfícies planas ou suavemente curvas enquanto que as ligações representam relações entre superfícies adjacentes. A comparação é efectuada através de uma combinação de processos de procura orientadas pela informação adquirida e pelo modelo. Inicialmente nodos "kernel", que consistem em superfícies planas largas sem oclusão, são extraídas. De seguida é efectuada uma procura exaustiva de todos os grafos dos modelos, e aqueles contendo regiões que estejam de acordo com esses nodos "kernel" são seleccionados como candidatos. Finalmente, uma procura com prioridade na profundidade (do tipo "depth-first") é aplicada para construir as correspondências para as restantes superfícies. Uma vez que apenas uma perspectiva é utilizada para cada modelo, se os objectos puderem ser vistos de múltiplas direcções então um grafo relacional separado deve ser construído para cada perspectiva, estes modelos devem ser tratados independentemente pelo processo de comparação. Para além disso não é permitida a ocorrência de oclusão para superfícies curvas.

Em [Nevatia77] é apresentado um sistema que utiliza cones como representação básica. A informação utilizada é a de distâncias. A aquisição do modelo e a descrição da cena são efectuadas da mesma forma porque os modelos podem ser adquiridos automaticamente este sistema também lida com a articulação de partes do objecto. Ele é testado em objectos com a complexidade de uma boneca ou de um cavalo. Este sistema utiliza também um esquema de indexação em características medidas por grosso de modo a que o reconhecimento não requeira uma procura por todos os modelos na base de dados. A principal limitação deste sistema advém da capacidade de gerar descrições apropriadas através de cones para objectos complexos.

Em [Brooks83], é apresentado um sistema de interpretação de imagem chamado ACRONYM que utiliza cilindros para descrever modelos e objectos. Os modelos são representados por grafos hierárquicos. O utilizador constroi uma árvore para cada

objecto sempre que os nodos contenham partes dos objectos representados por cilindros e as ligações representem as relações entre as suas sub-partes. Faixas e elipses são utilizados para descrever os objectos da cena. As descrições são finalmente representadas por um grafo de informação cujos nodos contêm descrições de faixas e elipses, enquanto que as ligações especificam relações espaciais entre nodos. A comparação é efectuada em dois níveis. Em primeiro lugar, faixas previstas devem justapôr-se a faixas da imagem e em segundo lugar, estas justaposições locais devem ser globalmente consistentes. A principal limitação do ACRONYM, reside no facto de modelos e grafos de restrições serem construídos pelo utilizador, o que torna o processo extenso e restringe a possibilidade de construção automática de modelos. Para além disso, uma vez que tanto os modelos como as cenas são representadas por cilindros que consistem em faixas e elipses a forma dos modelos e dos objectos é limitada. A direcção de visualização supõe-se também aproximadamente conhecida.

Em [Parvin92] é apresentada um método para produzir uma descrição integrada de um objecto, dadas várias imagens de distância ou "range" de diferentes pontos de observação e medida. A representação de vista múltipla é feita na forma de representação de fronteira. Cada vista do objecto é descrita como um grafo cujos nodos são compostos por parcelas de superfície, e as ligações representam as relações entre elas. Quaisquer dois grafos, cada um representando uma dada vista do objecto, são justapostos e comparados, e a transformação correspondente ao movimento rígido relativo é calculado. A estratégia básica para a integração das perspectivas múltiplas é composta por dois aspectos: em primeiro lugar é construído um grafo composto que contém as superfícies de fronteira e os respectivos atributos; seguidamente intersectam-se estas superfícies de modo a que as arestas e vértices correspondentes à descrição da representação de fronteira possam ser identificados. Este método é aplicável tanto a objectos de superfície poliédrica como curva desde que seja quadrática.

Em [Grau96] é apresentado o sistema AIDA que combina técnicas de reconstrução de superfícies com o reconhecimento de objectos para a geração de modelos 3D. Este sistema permite que sejam inseridas, de forma fácil, restrições geométricas tais como ângulos específicos entre superfícies. O conjunto das restrições extraídas vai constituir uma base de conhecimento para reconhecimento de objectos. A aplicabilidade daquelas restrições é provada através do teste de hipóteses numa primeira fase: a da interpretação. Em caso de rejeição, uma restrição mais geral ou modelo é seleccionado. As potencialidades deste método são demonstradas em particular em informação de contorno, e de profundidade extraída a partir de imagens stereo.

Em [Umeda96] são apresentados métodos para a fusão de imagens de "range" e de intensidade para o reconhecimento de figuras tridimensionais. Da imagem de "range" são extraídas regiões planas e cilíndricas através da associação de pixels de "range". Da imagem de intensidade são extraídas as arestas das várias figuras presentes. A necessidade em usar esta segunda imagem deriva do facto de esta ser densa, contrariamente à primeira que pela morosidade na sua obtenção é geralmente pouco densa. As arestas tridimensionais, no caso das regiões planas, são então determinadas a partir da fusão dessas regiões com as arestas bidimensionais encontradas. No caso dos cilindros, os parâmetros medidos unicamente através da análise da imagem de "range" são actualizados e rectificadas através da fusão com arestas bidimensionais. Obtêm-se

assim figuras planas e cilíndricas correctamente parametrizadas que são posteriormente identificadas através da respectiva representação EGI (Extended Gaussian Image).

Trabalhos similares neste tipo de abordagem podem ser encontrados em [Kadono91], [Devy94], [Nakazawa94], [Umeda94].

Em [Fayek96] é apresentado um mecanismo para modelização e reconhecimento de objectos 3D a partir de imagens de "range" densas. Em primeiro lugar a imagem de "range" é segmentada em pequenas superfícies triangulares. A partir destas unidades elementares de superfície, são detectadas as descontinuidades e os pontos extremos das concavidades. Seguidamente aquelas unidades são fundidas de acordo com as suas características, em várias iterações. Esta acção é levada a cabo mantendo sempre intacta a informação das descontinuidades e das concavidades. Neste processo é também removida toda a informação sensorial redundante. É então extraída informação simbólica que leve à modelização dos objectos presentes. Basicamente, este método explora as características topográficas 3D para gerar modelos tridimensionais.

Este método é particularmente adequado à modelização CAD automática de objectos reais complexos, cuja geração manual se apresente difícil. É também adequado quando os dados sensoriais disponíveis sejam imagens de "range" densas e com ruído.

3. Temas Abordados

Como se terá oportunidade de verificar, esta Tese foi bastante abrangente, tendo abordado inúmeros assuntos, tal como o exige um sistema que se pretende completo. De facto ela teve que versar todo o desenvolvimento matemático assim como todas as elaborações experimentais imprescindíveis ao funcionamento correcto e autónomo de um sistema de visão ligado a uma célula robótica de manipulação.

A abrangência dos trabalhos não impediu, contudo, que tivesse alcançado e mesmo nalguns casos ultrapassado, o estado da arte actual. De facto consegui introduzir aperfeiçoamentos em abordagens já existentes, nalguns casos, enquanto que noutros consegui desenvolvimentos originais.

Embora não faça parte desta Dissertação, não posso deixar de referir alguns trabalhos desenvolvidos relacionados com esta Tese.

Refiro-me, por exemplo, à programação em linguagem de alto nível do robot da Renault que me foi posto à disposição. Com ela foi-me possível construir uma estrutura hierárquica de programas que permitia executar uma série de operações de empilhamento de objectos poliédricos, apenas com instruções genéricas do tipo *Pega(obj_A)*, *ColocaA_emB(obj_A, obj_B)*, *ColocaA_naMesa(obj_A)*. Para que tal fosse possível, era mantida constantemente actualizada (automaticamente) uma base de dados que continha a localização de todos os objectos utilizados. Era ainda acedida uma outra base de dados que continha a informação das dimensões desses objectos, essencial para o seu correcto manuseamento assim como para actualizar aquele primeiro ficheiro. Qualquer um daqueles comandos genéricos chamava outros de menor nível que utilizava não só aquela informação como as funções próprias da linguagem. Esta continha para além de comandos, funções e variáveis normais para uma linguagem de

alto nível, outras próprias do ambiente a que se destinava: comandos de inicialização do robot, de accionamento directo dos motores e da garra, funções de leitura de sensores que também integrei no sistema, e finalmente o uso de variáveis e funções especiais que permitiam a execução de movimentos do robot através da estruturação do espaço – refiro-me à introdução de referenciais no espaço definíveis a partir do referencial absoluto do robot e obtidas por aplicação de matrizes de rotação e translação e à introdução destas últimas operações. Um qualquer movimento com o objectivo de apreender um dado objecto seria conseguido ordenando ao referencial da garra que coincidissem com um outro previamente definido pelo utilizador, ou construído conhecida a posição e dimensões desse objecto.

Um outro trabalho também desenvolvido no âmbito da Tese mas a que só farei referência aqui, diz respeito ao desenvolvimento de filtros a 2D. Estes programas destinavam-se essencialmente a eliminar ruídos de determinada espécie, na imagem obtida pela câmara, mas foram também desenvolvidos para obter determinados efeitos especiais. De entre os vários que desenvolvi saliento o de binarização que é utilizado previamente pelo Agente Visão antes de qualquer uma das suas análises de imagem (embora no Agente Laser viesse a ser substituído por uma binarização em tempo real através da manipulação no funcionamento da placa de aquisição de imagem).

Tendo como objectivos genéricos a localização, a medida da orientação e a identificação dos objectos na cena, foram desenvolvidos algoritmos que os permitiam atingir. Refiro-me por exemplo à análise e segmentação a 2D da imagem através da detecção de figuras separadas e medição das respectivas características geométricas. Estes valores, para além de fornecerem o posicionamento e orientação dos objectos, permitiam também calcular valores característicos que levavam à atribuição da sua identidade.

Em alternativa a esta análise de superfície foi também desenvolvida a análise de fronteira baseada na Transformada de Hough. Nesta abordagem procurou-se detectar e parametrizar os segmentos de recta contidos nas linhas que limitavam os objectos. Embora a orientação de algum segmento mais proeminente de um objecto pudesse dar a informação da própria orientação do objecto, este algoritmo destinou-se essencialmente à sua identificação. Para tal foi ainda desenvolvido um outro algoritmo que media com rigor quer o comprimento quer a orientação dos vários segmentos detectados e os encadeava consoante a sua proximidade relativa. O conjunto de informações de comprimento e ângulo relativo ao longo da fronteira de um objecto constituiu então uma estrutura identificativa do objecto em análise.

Tendo em consideração que a análise bidimensional da imagem não era suficiente para a correcta manipulação robótica automática dos objectos, foi desenvolvido um sistema completo com o objectivo de adquirir e analisar a informação da 3ª dimensão. Este sistema veio a constituir o Agente Laser, cuja acção pode conduzir a complementação dos algoritmos dos Agente Visão através do fornecimento de valores de altura, mas que também produz por si só resultados que podem concorrer com os obtidos por este. Mais concretamente pode reconhecer objectos existentes na cena com especial relevância para aqueles que se encontram em estado de sobreposição parcial ou total. De facto, esta situação que não é resolúvel a 2D (com a única excepção da análise de segmentos de fronteira que pode detectar no objecto parcialmente oculto algum

segmento ou ângulo interno suficientemente distinto de todos os outros) pode ocorrer com alguma frequência e requer na maior parte dos casos uma análise a 3D.

Para que fosse possível efectuar análises tridimensionais foi necessário desenhar, construir e aperfeiçoar um sistema de medida de alturas que utilizava uma fonte laser. O seu feixe, devidamente expandido de forma a produzir um plano de luz, devia atingir sucessivamente várias posições da cena de forma a que pudesse fornecer valores de altura para toda ela. A medida das alturas baseou-se no processo de triangulação: o feixe de luz que era direccionado obliquamente em direcção à cena, quando atingia um objecto produzia um desvio da linha laser vista pela câmara. A medida deste desvio, conjuntamente com o conhecimento da inclinação do feixe e de parâmetros relacionados quer com a geometria do sistema quer com a deformação radial da imagem, permitia obter os pretendidos valores de altura.

Foram também desenvolvidas funções de posicionamento e controle da linha laser por forma a permitir efectuar o seu varrimento de uma forma precisa. A sua estruturação cuidada em vários níveis permitiu obter também um controle de alto nível que o tornou "transparente" para o programador, facilitando assim quer as tarefas de calibração quer as de medida das alturas. Ela permitiu ainda obter mapas de altura com densidade uniforme contrariando a tendência dos pontos sujeitos a medida se afastarem com a diminuição do ângulo de inclinação do feixe laser, consequência do seu varrimento através da cena.

Para que aqueles valores de altura fossem o mais correctos possível, efectuou-se a medida de parâmetros vários, mecânicos e geométricos, através da calibração do sistema laser e fazendo uso daquele controle de alto nível da linha laser.

A medida da deformação da imagem foi também alvo do meu estudo, já que ela influenciava as medidas de altura. Conhecendo a função de deformação, esta podia então ser devidamente compensada naquele cálculo.

De referir ainda o estudo da conversão de coordenadas da imagem para as do sistema de referência do robot, que constitui um passo essencial para transmitir ao sistema que o controla a informação numérica imprescindível ao manuseamento robótico dos objectos. Esta conversão é efectuada conhecida a matriz de transformação, que para além de considerar translações, rotações e factores de escala, utiliza também a informação sobre o efeito de perspectiva cuja parametrização numérica depende de parâmetros intrínsecos e extrínsecos da câmara utilizada para adquirir as imagens. Estes valores não são, contudo, calculados separadamente, sendo em vez disso calculada aquela matriz global de transformação. Para que aquela conversão seja possível, é essencial o fornecimento de pelo menos uma das 3 coordenadas no sistema de referência do robot: por exemplo a 3ª dimensão, mensurável precisamente pelo sistema laser descrito acima.

Uma vez conseguida a aquisição de valores de altura, foram desenvolvidos filtros especiais que permitiam obter imagens de altura, ou seja, imagens dos objectos em que cada pixel continha, não o valor de intensidade medido pelo sistema de aquisição de imagem, mas um outro que consistia na da altura codificada.

A partir da segmentação dessas imagens e da parametrização das figuras encontradas, foi possível não só resolver o problema da oclusão parcial de objectos como obter um modo alternativo e concorrencial de os identificar.

4. Originalidades

Antes de mais quero chamar a atenção de que todo o sistema desenvolvido foi construído de raiz. Neste trabalho obteve-se um sistema completo que envolve processamentos em vários níveis de complexidade como também comunicação e ainda a montagem de diversos dispositivos mecânicos e electrónicos.

Foram também introduzidas contribuições originais em aspectos específicos que irei de seguida descrever.

4.1 Foi criado um sistema óptico de varrimento laser com controle electrónico para medição de alturas, apropriado à geometria e condicionalismos do espaço circundante do robot, e ainda obedecendo a requisitos e necessidades próprias da análise 3D a ser efectuada posteriormente. Para além do grande rigor obtido no posicionamento da linha laser, é de salientar como virtude deste sistema, a sua grande imunidade à luminosidade ambiente.

4.2 A formulação matemática desenvolvida para lidar com diversas situações de medida e de calibração é também uma inovação.

4.3 É também de referir os melhoramentos introduzidos na Transformada de Hough que é utilizada para detectar a presença de segmentos na linha de fronteira dos objectos. Para além deles, foi ainda desenvolvido um processamento complementar que eleva a eficácia naquela detecção para resultados muitíssimo acima do normalmente obtido com aquela transformada.

4.4 Finalmente refiro a abordagem na resolução do problema da oclusão parcial que utiliza histogramas de alturas construídos a partir das imagens também de alturas. O método baseia-se na detecção de picos e faixas cujo posicionamento relativo naqueles histogramas permite obter um valor de altura que defina uma separação em termos daquela medida (altura Z). Essa separação, por seu lado, irá permitir uma 2ª segmentação da figura correspondente ao conjunto de dois objectos em estado de sobreposição, de que resulta não só a parametrização do objecto superior para posterior identificação, como também o cálculo de parâmetros essenciais ao seu manuseamento robótico caso se pretenda a sua separação física efectiva.

5. Estrutura da Tese

Os vários temas descritos no ponto 3 encontram-se separados em vários capítulos e estruturados da seguinte forma:

5.1 No capítulo 1 é apresentada a formulação matemática e a descrição do processo de medida que levou à calibração da câmara utilizada. Esta calibração permitiu efectuar conversões de coordenadas do referencial da imagem para o referencial do robot. São ainda apresentados resultados experimentais que levaram ao cálculo da matriz de transformação assim como à avaliação dos erros cometidos nas conversões efectuadas.

5.2 No capítulo 2 é feita a apresentação do Agente Visão que controla todos os algoritmos de análise a 2D. Nele é feita referência ao processo de comunicação entre máquinas no que diz respeito à recepção de pedidos e ao envio de resultados.

5.3 No capítulo 3 é feita a descrição de todo o processo de segmentação da imagem a 2D que envolve a detecção de figuras distintas e na medição das respectivas características geométricas. Essas características destinam-se não só à localização e medida da orientação dos vários objectos detectados como também ao cálculo de parâmetros caracterizadores que permitam a sua identificação.

5.4 No capítulo 4 é feita a apresentação do processo de identificação a 2D que utiliza esse parâmetros geométricos. É ainda descrito o registo desses parâmetros na base de dados que irá funcionar como ficheiro de modelos, assim como um programa que efectua a sua gestão. São finalmente apresentados resultados experimentais que atestam a eficácia deste método de identificação.

5.5 No capítulo 5 é descrita a Transformada de Hough com os respectivos melhoramentos assim como o algoritmo de pós-processamento. Este conjunto permite obter com um elevado grau de eficácia, a descrição e parametrização dos segmentos existentes na imagem, contidos na linha de fronteira dos objectos.

5.6 No capítulo 6 é apresentado o algoritmo de encadeamento daqueles segmentos e separação em grupos por objecto. É feita a detecção de vértices que permitem obter os comprimentos reais dos segmentos e medidos correctamente os ângulos de inclinação. Com base nestes valores e na sequência dos segmentos para cada objecto, são construídas sequências numéricas de comprimento– ângulo que os caracterizem. Essas sequências numéricas são então utilizadas na identificação dos objectos pelo que são também reunidas numa base de dados.

5.7 No capítulo 7 é feita a apresentação do Agente Laser assim como do seu programa principal. Este programa efectua a comunicação com outros agentes, nomeadamente recebe pedidos e envia resultados, e efectua o controle de chamadas dos vários algoritmos que constituem o Agente Laser, de acordo com aqueles pedidos. Faz também a gestão das restrições geométricas que vêm incluídas naqueles pedidos.

5.8 No capítulo 8 é feita a descrição de todo o sistema óptico, mecânico e electrónico, que constituem o sistema de varrimento laser para medida de alturas. São

referidos os primeiros desenvolvimentos e feita a descrição do protótipo final tal como se encontra na actualidade.

5.9 No capítulo 9 é apresentada a formulação matemática que permite obter os valores de altura. É ainda descrito o processo de aquisição da linha laser para cada posição daquela na cena, a sua filtragem e medida dos desvios detectados na imagem binarizada. São também apresentados os vários modos de registo e leitura em disco do ficheiro de amostragem de alturas.

É também apresentado um estudo da deformação da imagem essencial à correcta avaliação das alturas. O resultado final desse estudo é a obtenção de uma função bidimensional (e portanto aplicável a toda a imagem) de conversão da dimensão do pixel para *mm*.

É também incluído a medição de vários parâmetros mecânicos e geométricos através da calibração do sistema laser.

É também incluída uma utilização dos ficheiros de amostragem de alturas, nomeadamente na tarefa de complementar os algoritmos 2D do Agente Visão com medidas de altura.

Finalmente é apresentado o processo de filtragem desses ficheiros de amostragem que levam à obtenção das imagens de altura, nas quais todos os pixels dos objectos têm um valor de altura definido.

5.10 No capítulo 10 é descrita a parametrização dos modelos 3D. Mais concretamente são apresentados os histogramas de altura para várias situações de interesse e a respectiva parametrização de que resulta aquilo que convencionei chamar de "Assinaturas".

É feita a descrição dos programas que as obtêm e apresentados alguns resultados experimentais.

É também apresentado um estudo de fiabilidade dos vários parâmetros que constituem a "Assinatura".

É ainda apresentado o ficheiro de modelos 3D que contém o registo das "Assinaturas" dos modelos conhecidos e utilizado para a identificação 3D dos objectos detectados na imagem de alturas.

5.11 No capítulo 11 é apresentado o programa que efectua a segmentação das imagens de altura e ordena a sub-segmentação sempre que é detectada uma situação de sobreposição.

É também descrito todo o processo de segmentação primária dessas imagens assim como o de sub-segmentação.

No final são apresentados resultados experimentais com objectos reais em situação de sobreposição.

5.12 No capítulo 12 é apresentado o algoritmo de identificação a 3D que utiliza as "Assinaturas". É descrita toda a estrutura desse algoritmo com especial destaque para o processo de comparação entre objecto e modelo e cálculo do respectivo coeficiente que se diferencia do anteriormente utilizado no processo a 2D.

5.13 Finalmente são apresentadas as conclusões e perspectivas de trabalho futuro.

1.1 Introdução

Este capítulo é dedicado fundamentalmente à esquematização de processos de conversão de coordenadas da imagem adquirida pela câmara, em unidades de pixel, para coordenadas no mundo real. De uma forma equivalente pode-se afirmar que se trata de um problema de calibração da câmara utilizada para adquirir imagens da cena robótica. Este passo é essencial para que o robot tenha "conhecimento" das localizações exactas dos objectos a manipular por forma a executar as tarefas que lhe são impostas pelo planeador de acções.

Em geral, contudo, os problemas de calibração destinam-se a calcular por vários processos, todos os parâmetros individuais que caracterizam quer o sistema óptico da câmara quer as transformações geométricas que ligam os dois sistemas de referência, como o vector de posição que liga as duas origens ou ainda ângulos de rotação em torno de determinados eixos. Nesse sentido foram desenvolvidos vários métodos [Grattoni91] que apresentarei sucintamente para referência.

Em [Tsai86], apresenta-se um método que permite calcular tanto os parâmetros intrínsecos como extrínsecos da câmara, considerando também a distorção provocada pela lente. O método também permite o cálculo de coeficientes de correcção para as dimensões do pixel relativamente ao centro de distorção quando ela existe.

Em [Faugeras87] não é considerada a distorção provocada pela lente. A câmara é descrita por quatro parâmetros intrínsecos: duas distâncias focais, uma para cada eixo do plano da imagem de forma a considerar as diferentes resoluções assim como o centro da imagem. Este modelo não requer o conhecimento das dimensões do pixel.

Em [Martins81] não é feito uso explícito de modelos de câmara. Este método fornece a recta da linha de vista, em coordenadas do mundo real, para cada ponto a partir das imagens de pontos de calibração pertencentes a dois planos: um na parte da frente e o outro na parte de trás do espaço de trabalho. Dados aqueles pontos de calibração no espaço e as respectivas imagens dadas pela câmara, são construídos dois mapas por interpolação, de tal modo que para cada ponto na imagem, são definidos dois pontos (um na frente e outro no plano de trás) que fornecem o vector pretendido da linha de vista.

Em [Pollastri89] também não é considerada a distorção da imagem provocada pela lente. O objectivo deste método é determinar o centro da imagem e a distância focal f independentemente de outros parâmetros da câmara. Este método é baseado na relação

geométrica entre o centro de projecção O e os focos de expansão. O uso desta relação permite separar o problema em duas partes. Na primeira, é calculado um número adequado N de focos de expansão produzidos por translações conhecidas de um objecto rígido. Na segunda, os focos de expansão são usados como pontos de referência para, por triangulação, obter os parâmetros pretendidos.

Em [Caprile90], que também não considera a distorção provocada pela lente, são consideradas as propriedades dos pontos de fuga na imagem: se forem fornecidos os pontos de fuga associados às três direcções perpendiculares do espaço, então o centro do triângulo com vértices naqueles três pontos corresponderá ao centro da imagem. Os parâmetros intrínsecos da câmara podem ser calculados a partir da construção de três conjuntos mutuamente perpendiculares de linhas paralelas desenhadas na faces de um cubo. Os pontos de fuga associados a cada conjunto são usados para calcular o centro da imagem e a distância focal f .

Finalmente em [Batista93] é apresentado um novo método de calibração baseado numa computação linear e utilizando um grupo coplanar de pontos de calibração. Este novo método suprime a restrição apresentada pelo modelo de calibração RAC [Tsai87] de existir um ângulo de incidência entre o eixo óptico e o plano de calibração de pelo menos 30° . A maior contribuição deste novo método relaciona-se com o cálculo da matriz de rotação. O princípio básico que leva ao cálculo dessa matriz foi apresentado por Haralick e é baseado no facto de haver informação de perspectiva suficiente na projecção bidimensional de um rectângulo para determinar os parâmetros do ângulo de visão da câmara. Com o conhecimento da matriz de rotação, o vector de translação e todos os parâmetros intrínsecos podem ser obtidos facilmente. Os parâmetros são obtidos com base num cálculo em várias etapas, e a precisão dos parâmetros é drasticamente aumentada quando se usa informação de retorno num processo de calibração com várias iterações.

Embora o cálculo individual, em alguns destes trabalhos, dos vários parâmetros (intrínsecos e extrínsecos) possa ser de grande interesse, ele é perfeitamente dispensável no caso de se pretender a calibração global do sistema de visão, como se verá mais adiante. De facto, é suficiente para os objectivos traçados acima, calcular apenas uma matriz transformação de 4×4 que traduza todas as operações geométricas necessárias para fazer coincidir os dois sistemas de referência, ou o que é equivalente, que transforme um tipo de coordenadas no outro.

Foi então desenvolvido todo um suporte matemático, baseado em cálculo matricial, que permitisse aquela transformação.

A obtenção daquela matriz de transformação foi efectuada experimentalmente a partir da leitura de coordenadas de imagem de pontos pertencentes a uma grelha de calibração, cujas coordenadas reais eram pré-conhecidas e para alturas diferentes relativamente ao plano $Z=0$. Neste procedimento, a câmara foi utilizada como instrumento de medida.

Os sub-capítulos seguintes apresentam detalhadamente todos estes desenvolvimentos.

1.2 A Conversão a 2D

Este processo consiste na escolha de pontos chave para os quais é feita simultaneamente a medida das respectivas coordenadas no referencial do robot e no referencial da imagem. Qualquer outro ponto para o qual se pretenda conhecer aquelas primeiras coordenadas conhecidas as segundas, deve ser confinado a um "quadrado" de pontos chave e as coordenadas pretendidas calculadas em função da proximidade a qualquer um deles.

Os pontos a ser escolhidos devem cobrir todo o espaço simultaneamente visível pela câmara e acedível pelo braço de robot e o espaçamento deve ser tal que permita obter boas precisões (erros de posicionamento inferiores a 2mm). Tendo em conta o atrás exposto, foi escolhido um espaçamento de 50mm no rectângulo definido pelos pontos extremos da sua diagonal ($x=-100; y=-300$) e ($x=250; y=350$) (Fig.1).

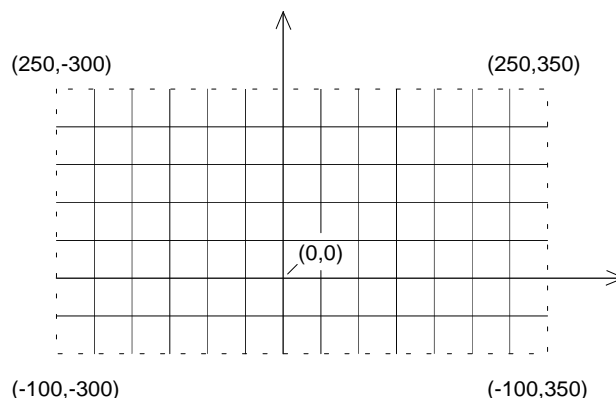


Fig. 1 - Grelha de pontos escolhidos para constarem do ficheiro de conversão.

Uma vez seleccionados os pontos a ser associados a coordenadas do referencial do robot, resta adquirir a imagem desses pontos e recolher as respectivas coordenadas na imagem. Essa tarefa foi feita com o apoio de um programa auxiliar que funciona de um modo interactivo.

Então, sempre que se pretenda conhecer as coordenadas de um determinado ponto da imagem (x,y), procuram-se os 4 pontos que o delimitam com coordenadas do referencial do robot, conhecidas. Atribui-se então uns valores X e Y consoante os 4 pares (X,Y) conhecidos e a proximidade do referido ponto a cada um deles (Fig.2).

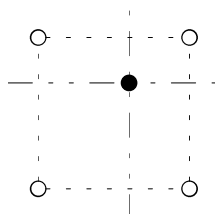


Fig. 2 - Cálculo das coordenadas X e Y do ponto consoante a sua proximidade a cada um dos 4 pontos delimitadores.

O sistema desenvolvido contempla ainda a possibilidade do ponto se encontrar não no interior do "quadrado" mas num dos seus lados e ainda a de se encontrar coincidente com um dos pontos chave. Estes últimos casos são triviais.

Este processo a 2D apresenta algumas vantagens sobre outros. Todo o processo apresenta uma simplicidade inigualável, tanto no processo de aquisição de pontos como no cálculo, que associada a uma grande rapidez de conversão de coordenadas torna-o bastante atraente. Mais ainda, ele contempla a possibilidade do espaço cartesiano do robot se encontrar afectado com erros de posicionamento, o que se verificou ser verdade para o robot de que dispunha. É claro que para outro robot seria necessário repetir todo o processo de calibração, pois ele decerto apresentaria erros de posicionamento diferentes ou mesmo erros não mensuráveis.

Por outro lado, para que ele seja válido há que garantir que os objectos possuam alturas muito inferiores à sua distância à câmara, já que as imagens dos objectos são projectadas no plano $Z=0$ e daí o tratamento apenas a 2D.

Várias experiências foram realizadas que mostraram resultados satisfatórios.

Em geral e para outras situações que não contemplem aquelas condições, há que escolher outro método que utilize os valores da 3ª dimensão dos objectos como aquele que apresento a seguir e que foi adoptado pelos algoritmos de medida a 3D.

1.3 A Conversão a 3D: O Fundamento Teórico

Dados dois sistemas de referência diferentes, como é o caso da imagem dada pela câmara e o do robot, é sempre possível construir uma matriz que traduza um tipo de coordenadas para outro conhecidas as operações geométricas necessárias para fazer coincidir os dois sistemas. A transformação de coordenadas será dada por uma operação matricial dessa matriz sobre o vector de coordenadas a converter.

Apresento de seguida todos os tipos de matrizes de transformação possíveis consoante a operação geométrica efectuada assim como uma forma de incorporar todas elas numa só matriz.

Mais adiante será apresentada uma forma de obter essa matriz global de transformação e ainda um modo sistemático de calcular as coordenadas pretendidas a partir dela.

1.3.1 A Matriz Rotação

Consideremos primeiramente uma rotação do sistema de referência e procuremos um modo de relacionar as antigas coordenadas com as novas.

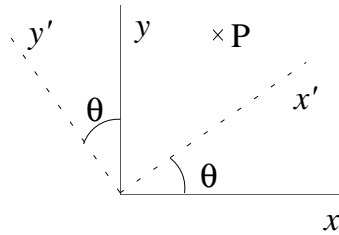


Fig. 3 - A rotação de um sistema de referência e a obtenção das novas coordenadas.

Após a rotação do referencial, os novos versores \hat{i}' e \hat{j}' poderão ser descritos da seguinte forma:

$$\begin{cases} \hat{i}' = \cos\theta\hat{i} + \text{sen}\theta\hat{j} \\ \hat{j}' = -\text{sen}\theta\hat{i} + \cos\theta\hat{j} \end{cases}$$

Um ponto P com coordenadas (x',y') terá no antigo referencial as seguintes coordenadas:

$$P: x'\hat{i}' + y'\hat{j}' = x'(\cos\theta\hat{i} + \text{sen}\theta\hat{j}) + y'(-\text{sen}\theta\hat{i} + \cos\theta\hat{j}) =$$

$$= (x'\cos\theta - y'\text{sen}\theta)\hat{i} + (x'\text{sen}\theta + y'\cos\theta)\hat{j} = \begin{bmatrix} \cos\theta & \text{sen}\theta \\ \text{sen}\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} (\hat{i}'|\hat{i}) & (\hat{i}'|\hat{j}) \\ (\hat{j}'|\hat{i}) & (\hat{j}'|\hat{j}) \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix} =$$

$$= \begin{bmatrix} \boxed{\phantom{\hat{i}'}} & \boxed{\phantom{\hat{j}'}} \\ \bullet & \bullet \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix}$$

novo versor \hat{i}' projectado
no antigo referencial

novo versor \hat{j}' projectado
no antigo referencial

Em geral, para referenciais a 3D, temos então a seguinte matriz rotação,

$$\begin{bmatrix} (\hat{i}'|\hat{i}) & (\hat{j}'|\hat{i}) & (\hat{k}'|\hat{i}) \\ (\hat{i}'|\hat{j}) & (\hat{j}'|\hat{j}) & (\hat{k}'|\hat{j}) \\ (\hat{i}'|\hat{k}) & (\hat{j}'|\hat{k}) & (\hat{k}'|\hat{k}) \end{bmatrix} = R^{-1}$$

que fornece as coordenadas no antigo referencial em função do novo vector (x',y',z') .

A matriz rotação, que fornece as novas coordenadas em função das antigas é então a seguinte:

$$R = \begin{bmatrix} (\hat{i}'|\hat{i}) & (\hat{i}'|\hat{j}) & (\hat{i}'|\hat{k}) \\ (\hat{j}'|\hat{i}) & (\hat{j}'|\hat{j}) & (\hat{j}'|\hat{k}) \\ (\hat{k}'|\hat{i}) & (\hat{k}'|\hat{j}) & (\hat{k}'|\hat{k}) \end{bmatrix}$$

1.3.2 A Matriz Translação

Se o referencial efectuar apenas uma translação, então as coordenadas no antigo sistema será dado pela soma do vector posição da nova origem:

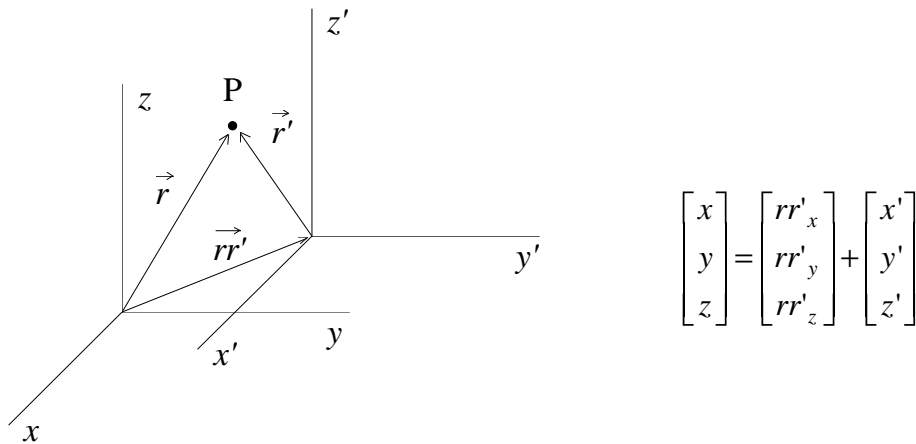


Fig. 4 - A translação de um sistema de referência e a obtenção das novas coordenadas.

onde (rr'_x, rr'_y, rr'_z) é o vector posição do centro do novo referencial em relação ao antigo. (x', y', z') pode também ser obtido de (x, y, z) :

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} -rr'_x \\ -rr'_y \\ -rr'_z \end{bmatrix} + \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

1.3.3 As Coordenadas Homogêneas

Como traduzir aquela soma anterior em termos de um produto matricial? Para isso tem-se que introduzir mais 1 dimensão àquela matriz:

$$T = \begin{bmatrix} 1 & 0 & 0 & -rr'_x \\ 0 & 1 & 0 & -rr'_y \\ 0 & 0 & 1 & -rr'_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Esta matriz quando aplicada a (x, y, z) origina a soma do vector posição:

$$T \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -rr'_x \\ 0 & 1 & 0 & -rr'_y \\ 0 & 0 & 1 & -rr'_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} (x - rr'_x) \\ (y - rr'_y) \\ (z - rr'_z) \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}$$

Da mesma forma, a matriz rotação pode adquirir a nova dimensão:

$$R = \begin{bmatrix} (\hat{i}'|\hat{i}) & (\hat{i}'|\hat{j}) & (\hat{i}'|\hat{k}) & 0 \\ (\hat{j}'|\hat{i}) & (\hat{j}'|\hat{j}) & (\hat{j}'|\hat{k}) & 0 \\ (\hat{k}'|\hat{i}) & (\hat{k}'|\hat{j}) & (\hat{k}'|\hat{k}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Uma rotação seguida de translação pode então ser descrita como um produto das duas matrizes mostradas acima:

$$T \cdot R = \begin{bmatrix} 1 & 0 & 0 & -rr'_x \\ 0 & 1 & 0 & -rr'_y \\ 0 & 0 & 1 & -rr'_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} (\hat{i}'|\hat{i}) & (\hat{i}'|\hat{j}) & (\hat{i}'|\hat{k}) & 0 \\ (\hat{j}'|\hat{i}) & (\hat{j}'|\hat{j}) & (\hat{j}'|\hat{k}) & 0 \\ (\hat{k}'|\hat{i}) & (\hat{k}'|\hat{j}) & (\hat{k}'|\hat{k}) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} (\hat{i}'|\hat{i}) & (\hat{i}'|\hat{j}) & (\hat{i}'|\hat{k}) & -rr'_x \\ (\hat{j}'|\hat{i}) & (\hat{j}'|\hat{j}) & (\hat{j}'|\hat{k}) & -rr'_y \\ (\hat{k}'|\hat{i}) & (\hat{k}'|\hat{j}) & (\hat{k}'|\hat{k}) & -rr'_z \\ 0 & 0 & 1 & 1 \end{bmatrix}$$

A matriz resultante contém, como se pode observar, toda a informação contida nas duas anteriores, com a vantagem de substituir um somatório (translação) por um produto matricial.

Outras operações geométricas que ocorram podem ser acrescentadas àquelas, formando uma sucessão de produtos matriciais devidamente ordenados. A matriz resultante de todos esses produtos constituirá a matriz transformação final que aplicada a qualquer vector (x,y,z) dá como resultado o novo vector (x',y',z') :

Os vectores a ser transformados por aquelas matrizes podem ainda ser escritos de uma forma mais geral:

$$c_h = \begin{bmatrix} c_x \\ c_y \\ c_z \\ k_c \end{bmatrix} = A \cdot \begin{bmatrix} w_x \\ w_y \\ w_z \\ k_w \end{bmatrix} = A \cdot w_h$$

onde w_h corresponde ao vector 1×4 no referencial robot ("world coordinates") e c_h corresponde ao vector 1×4 no referencial da imagem. As coordenadas neles contidas são designadas por homogêneas e relacionam-se com os valores medidos através das relações,

$$c = c_h / k_c = \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} c_x / k_c \\ c_y / k_c \\ c_z / k_c \end{bmatrix} \quad ; \quad w = w_h / k_w = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} w_x / k_w \\ w_y / k_w \\ w_z / k_w \end{bmatrix}$$

Se o antigo referencial for o do robot e o novo o da imagem, então deverá ser possível obter uma única matriz transformação de 4×4 , conhecidas todas as transformações geométricas, que transforme qualquer conjunto de coordenadas reais (X,Y,Z) nas correspondentes coordenadas da imagem $(x',y',0)$ (Fig.5):

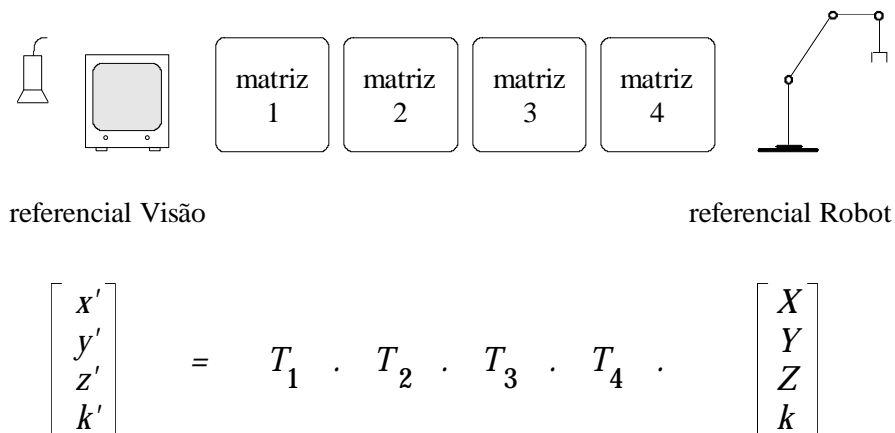


Fig. 5 - O conjunto de matrizes que permitem a transformação de coordenadas.

Essa matriz, que é um dos grandes objectivos deste capítulo, deverá ser medida experimentalmente para posterior uso na conversão inversa: de coordenadas da imagem para as correspondentes coordenadas a 3D, ou seja, no referencial do robot.

1.3.4 A Transformação de Perspectiva

Esta transformação consiste no mapeamento no plano xy da imagem, de todo um conjunto de pontos do mundo real, (X,Y,Z) . Utilizando o modelo "pin-hole" para a câmara vamos de facto conseguir aquele mapeamento, situando-se a abertura no ponto $(0,0,\lambda)$. Se as distâncias Z forem muito superiores a λ , então este valor vai corresponder à distância focal da câmara, que no caso concreto da câmara usada é de 12,5mm.

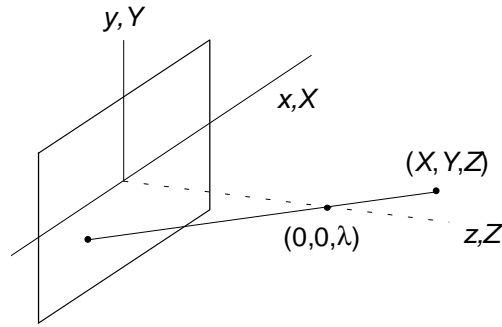


Fig. 6 - Mapeamento de pontos (X, Y, Z) no plano xy .

De acordo com a Fig.6 e usando a similaridade de triângulos rectângulos, obtêm-se as seguintes relações:

$$\begin{cases} \frac{x}{\lambda} = -\frac{X}{Z-\lambda} = \frac{X}{\lambda-Z} \\ \frac{y}{\lambda} = \frac{Y}{\lambda-Z} \end{cases} \quad \begin{cases} x = \frac{\lambda X}{\lambda-Z} \\ y = \frac{\lambda Y}{\lambda-Z} \end{cases}$$

Então, se se utilizar as coordenadas homogêneas do ponto (X, Y, Z) , $w_h = (kX, kY, kZ, k)$, com $k \neq 0$, obtêm-se a seguinte matriz de transformação:

$$P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & (-1/\lambda) & 1 \end{bmatrix}$$

Ou seja, quando aplicada àquele vector, obtêm-se as coordenadas homogêneas no espaço xyz :

$$P \cdot w_h = c_h = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & (-1/\lambda) & 1 \end{bmatrix} \begin{bmatrix} kX \\ kY \\ kZ \\ k \end{bmatrix} = \begin{bmatrix} kX \\ kY \\ kZ \\ -kZ/\lambda + k \end{bmatrix}$$

Os valores reais obtêm-se por divisão pela 4ª componente do vector:

$$c = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} kX / (k - kZ/\lambda) \\ kY / (k - kZ/\lambda) \\ kZ / (k - kZ/\lambda) \end{bmatrix} = \begin{bmatrix} \lambda X / (\lambda - Z) \\ \lambda Y / (\lambda - Z) \\ \lambda Z / (\lambda - Z) \end{bmatrix}$$

As duas primeiras componentes correspondem de facto às expressões atrás obtidas para (x, y) . A 3ª componente é por agora irrelevante: ela irá posteriormente funcionar como variável livre.

A transformação inversa, por outro lado, mapeia um ponto da imagem em coordenadas reais 3D:

$$w_h = P^{-1} \cdot c_h$$

onde,

$$P^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & (1/\lambda) & 1 \end{bmatrix}$$

Consideremos então agora um ponto da imagem com coordenadas (x_o, y_o) e as correspondentes coordenadas homogêneas:

$$c = (x_o, y_o, 0) \rightarrow c_h = (kx_o, ky_o, 0, k)$$

Por aplicação da matriz inversa de perspectiva, obtém-se:

$$w_h = P^{-1} \cdot c_h = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & (1/\lambda) & 1 \end{bmatrix} \begin{bmatrix} kx_o \\ ky_o \\ 0 \\ k \end{bmatrix} = \begin{bmatrix} kx_o \\ ky_o \\ 0 \\ k \end{bmatrix} \rightarrow w = \begin{bmatrix} x_o \\ y_o \\ 0 \end{bmatrix}$$

ou seja, $w = c$ com $Z=0$!

Este é um resultado trivial e apenas um dos muitos possíveis. De facto, os valores de (X, Y) podem situar-se nas rectas seguintes, de acordo com a Fig.6 :

$$\begin{cases} X = \frac{x_o}{\lambda}(\lambda - Z) \\ Y = \frac{y_o}{\lambda}(\lambda - Z) \end{cases}$$

Para que se possa mapear os pontos da imagem de volta às coordenadas reais 3D, é necessário conhecer pelo menos uma das 3 coordenadas; por exemplo Z .

A obtenção daquelas linhas como resultados gerais implica o uso de z como uma variável livre:

$$c = (\cancel{x_o}, \cancel{y_o}, 0) \longrightarrow c = (x_o, y_o, z)$$

embora (x_o, y_o) se situe no plano xy .

Ou seja, c_h vai ter as seguintes coordenadas:

$$c_h = \begin{bmatrix} kx_o \\ ky_o \\ kz \\ k \end{bmatrix}$$

e,

$$w_h = P^{-1} \cdot c_h = \begin{bmatrix} kx_o \\ ky_o \\ kz \\ \left(\frac{kz}{\lambda} + k\right) \end{bmatrix}$$

Donde se extraem as coordenadas cartesianas no mundo real:

$$w = \begin{bmatrix} \frac{kx_o}{\left(\frac{kz}{\lambda} + k\right)} \\ \frac{ky_o}{\left(\frac{kz}{\lambda} + k\right)} \\ \frac{kz}{\left(\frac{kz}{\lambda} + k\right)} \end{bmatrix} = \begin{bmatrix} \frac{x_o}{(z/\lambda + 1)} \\ \frac{y_o}{(z/\lambda + 1)} \\ \frac{z}{(z/\lambda + 1)} \end{bmatrix}$$

Tratando z como uma variável livre obtém-se:

$$\begin{cases} X = \frac{x_o}{(z/\lambda + 1)} \\ Y = \frac{y_o}{(z/\lambda + 1)} \\ Z = \frac{z}{(z/\lambda + 1)} \end{cases} \Leftrightarrow \begin{cases} X = \frac{x_o}{\lambda} (\lambda - Z) \\ Y = \frac{y_o}{\lambda} (\lambda - Z) \\ - \end{cases}$$

E este resultado está de acordo com o já obtido anteriormente e com a observação de que é necessário conhecer-se pelo menos uma das coordenadas reais do ponto.

1.3.5 A Calibração da Câmara

A matriz final de transformação pode ser obtida através de um conjunto de matrizes que traduzem todas as operações geométricas que levam o sistema de referência do robot a coincidir com o da câmara. Tendo em conta que é necessário efectuar sucessivamente uma translação, uma rotação em torno de z seguida de outra em torno de x (traduzível por uma única matriz de rotação) por uma segunda translação (se o eixo de rotação na câmara não coincidir com o centro da imagem) e finalmente por

uma transformação de perspectiva, então a matriz final será dada pelo seguinte produto matricial:

$$A = P \cdot C \cdot R \cdot G$$

Perspectiva outra Translação Rotação Translação

O resultado desta operação é sempre do tipo apresentado a seguir:

Nela estão contidas todas as operações de transformação de perspectiva, de rotação de translação e ainda de escala, cuja informação se encontra posicionada da seguinte forma:

$$A = \begin{bmatrix} \begin{matrix} 3 \times 3 \\ \text{matriz de} \\ \text{rotação} \end{matrix} & \begin{matrix} 3 \times 1 \\ \text{vector de} \\ \text{posição} \end{matrix} \\ \begin{matrix} \text{transformação} \\ \text{de perspectiva} \end{matrix} & \begin{matrix} \text{escala} \\ 1 \times 1 \end{matrix} \end{bmatrix}$$

Esta matriz de transformação encerra em si todos os parâmetros intrínsecos e extrínsecos de uma câmara. Para os conhecer podemos utilizar a própria câmara como instrumento de medida, adquirindo imagens de conjuntos de pontos cujas coordenadas reais são conhecidas antecipadamente. Este procedimento é chamado de calibração de uma câmara.

No meu trabalho, contudo, os valores daqueles parâmetros, que podem ser obtidos como função dos elementos a_{ij} , não são de particular interesse. Em vez disso interessa-me antes conhecer os valores desses mesmos elementos a_{ij} .

1.3.6 Obtenção dos Elementos a_{ij} da Matriz de Transformação

Se c_h e w_h forem as coordenadas homogêneas respectivamente da imagem e do mundo real e A for a matriz transformação, temos então:

$$c_h = A \cdot w_h$$

Se fizermos $k = 1$, ficará:

$$\begin{bmatrix} c_{h_1} \\ c_{h_2} \\ c_{h_3} \\ c_{h_4} \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Ainda, tendo em conta que, $\begin{cases} x = c_{h_1} / c_{h_4} \\ y = c_{h_2} / c_{h_4} \end{cases}$ e substituindo fica:

$$\begin{cases} xc_{h_4} = a_{11}X + a_{12}Y + a_{13}Z + a_{14} \\ yc_{h_4} = a_{21}X + a_{22}Y + a_{23}Z + a_{24} \\ c_{h_4} = a_{41}X + a_{42}Y + a_{43}Z + a_{44} \end{cases}$$

onde c_{h_3} foi ignorado uma vez que se relaciona com z .

Substituindo c_{h_4} nas duas primeiras equações obtém-se um sistema com 12 coeficientes:

$$\begin{cases} a_{11}X + a_{12}Y + a_{13}Z - a_{41}xX - a_{42}xY - a_{43}xZ - a_{44}x + a_{14} = 0 \\ a_{21}X + a_{22}Y + a_{23}Z - a_{41}yX - a_{42}yY - a_{43}yZ - a_{44}y + a_{24} = 0 \end{cases}$$

Estes coeficientes só podem ser obtidos se forem conhecidas as coordenadas (x, y, X, Y, Z) de pelo menos 6 pontos reais, ou seja, a partir de 6×2 equações a 12 incógnitas [Noble69].

No caso presente, pretende-se utilizar mais do que aqueles 6 pontos por forma a ajustar a calibração para o maior número de pontos do espaço utilizado pelo robot. Vai-se por isso ter um sistema sobredimensionado que requer uma atenção especial [Strang88].

Acrescentando os termos nulos referentes às variáveis em falta em cada uma das equações, obtém-se:

$$\begin{cases} a_{11}X_1 + a_{12}Y_1 + a_{13}Z_1 + a_{14} \cdot 1 + a_{21} \cdot 0 + a_{22} \cdot 0 + a_{23} \cdot 0 + a_{24} \cdot 0 - a_{41}x_1X_1 - a_{42}x_1Y_1 - a_{43}x_1Z_1 - a_{44}x_1 = 0 \\ a_{11} \cdot 0 + a_{12} \cdot 0 + a_{13} \cdot 0 + a_{14} \cdot 0 + a_{21}X_1 + a_{22}Y_1 + a_{23}Z_1 + a_{24} \cdot 1 - a_{41}y_1X_1 - a_{42}y_1Y_1 - a_{43}y_1Z_1 - a_{44}y_1 = 0 \end{cases}$$

ficando assim a figurar em ambas as equações a totalidade dos 12 coeficientes desconhecidos. Para os conseguir calcular são necessários pelo menos 6 pontos de calibração que devem ser bem escolhidos: a sua localização deve ser tal que forneça informação quer dos factores de escala e de distorção geométrica, quer da translação, rotação e perspectiva. Mas uma forma de contornar esse cuidado e simultaneamente dotar as medidas de maior precisão é utilizar um número redundante de pontos (de preferência elevado) conjuntamente com o método dos mínimos quadrados. Neste método os coeficientes são escolhidos de forma a minimizar os 2.^{os} membros, cujos valores deveriam ser nulos de acordo com as duas equações apresentadas atrás.

De forma a evitar a solução trivial $a_{11}=a_{12}=\dots=a_{44}=0$, é conveniente arbitrar o valor de um dos coeficientes. Por exemplo se $a_{44} = 1$, obtém-se:

$$\begin{cases} a_{11}X_1 + a_{12}Y_1 + a_{13}Z_1 + a_{14}a_{21} \cdot 0 + a_{22} \cdot 0 + a_{23} \cdot 0 + a_{24} \cdot 0 - a_{41}x_1X_1 - a_{42}x_1Y_1 - a_{43}x_1Z_1 = x_1 \\ a_{11} \cdot 0 + a_{12} \cdot 0 + a_{13} \cdot 0 + a_{14}a_{21}X_1 + a_{22}Y_1 + a_{23}Z_1 + a_{24} \cdot 1 - a_{41}y_1X_1 - a_{42}y_1Y_1 - a_{43}y_1Z_1 = y_1 \end{cases}$$

ficando assim o sistema com um 2º membro não nulo.

Esta escolha vem afectar os valores finais da matriz de transformação. Contudo, isto apenas originará coordenadas homogéneas diferentes consoante a matriz final obtida, já que o termo k (4ª coordenada homogénea) também será uma função desses coeficientes e portanto variável. As coordenadas reais resultantes da divisão de w_h por esse termo k deverão ser então invariáveis.

Se tiver N pontos, serão $2 \times N$ equações a 11 incógnitas e a matriz será de $2N \times 11$:

$$\begin{bmatrix} b_{1\ 1} & b_{1\ 2} & \dots & b_{1\ 11} \\ b_{2\ 1} & b_{2\ 2} & \dots & b_{2\ 11} \\ \vdots & \vdots & & \vdots \\ b_{2N\ 1} & b_{2N\ 2} & \dots & b_{2N\ 11} \end{bmatrix} \begin{array}{l} \rightarrow \text{ponto 1} \\ \\ \\ \rightarrow \text{ponto N} \end{array}$$

em que,

$$\begin{cases} b_{11} = X_1; b_{12} = Y_1; b_{13} = Z_1; b_{14} = 1; b_{15} = 0; b_{16} = 0; b_{17} = 0; b_{18} = 0; b_{19} = -x_1 X_1; b_{1\ 10} = -x_1 Y_1; b_{1\ 11} = -x_1 Z_1 \\ b_{21} = 0; b_{22} = 0; b_{23} = 0; b_{24} = 0; b_{25} = X_1; b_{26} = Y_1; b_{27} = Z_1; b_{28} = 1; b_{29} = -y_1 X_1; b_{2\ 10} = y_1 Y_1; b_{2\ 11} = -y_1 Z_1 \end{cases}$$

relativas ao ponto 1.

O vector sobre o qual a matriz opera é:

$$\begin{bmatrix} a_{11} \\ a_{12} \\ \vdots \\ a_{42} \\ a_{43} \end{bmatrix}$$

ou seja, os 11 coeficientes desconhecidos ($a_{44}=1$).

e o resultado (2º membro) é:

$$\begin{bmatrix} x_1 \\ y_1 \\ \vdots \\ x_N \\ y_N \end{bmatrix}$$

A partir de $N \geq 6$ pontos e usando o método dos mínimos quadrados [Strang88], é possível obter os 12 coeficientes (dos quais 1 já é conhecido: $a_{44}=1$) e com os quais se constrói a matriz A de transformação de dimensão 3×4 (a 3ª linha é ignorada). Alternativamente, pode ser obtida uma outra matriz M de dimensão 2×2 , dependente da coordenada Z e que permite a transformação directa $(x,y) \leftarrow \rightarrow (X,Y)$, como se verá adiante.

A operação matricial fica então com a seguinte forma:

$$\begin{bmatrix} X_1 & Y_1 & Z_1 & 1 & 0 & 0 & 0 & 0 & -x_1 X_1 & -x_1 Y_1 & -x_1 Z_1 \\ 0 & 0 & 0 & 0 & X_1 & Y_1 & Z_1 & 1 & -y_1 X_1 & -y_1 Y_1 & -y_1 Z_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ X_N & Y_N & Z_N & 1 & 0 & 0 & 0 & 0 & -x_N X_N & -x_N Y_N & -x_N Z_N \\ 0 & 0 & 0 & 0 & X_N & Y_N & Z_N & 1 & -y_N X_N & -y_N Y_N & -y_N Z_N \end{bmatrix} \cdot \begin{bmatrix} a_{11} \\ a_{12} \\ a_{13} \\ a_{14} \\ a_{21} \\ a_{22} \\ a_{23} \\ a_{24} \\ a_{41} \\ a_{42} \\ a_{43} \end{bmatrix} = \begin{bmatrix} x_1 \\ y_1 \\ \vdots \\ \vdots \\ \vdots \\ x_N \\ y_N \end{bmatrix}$$

com $N \geq 6$ (nº de pontos de calibração).

De acordo com o referido método dos mínimos quadrados, a solução para um sistema sobredimensionado $A \cdot x = b$, com m equações em n incógnitas é:

$$A^T A \bar{x} = A^T b$$

onde A^T representa a matriz transposta de A . Se as colunas de A (nova matriz de $m \times n$) forem linearmente independentes, então $(A^T A)$ é invertível e a solução é dada por:

$$\bar{x} = (A^T A)^{-1} A^T b$$

Foi então desenvolvido um conjunto de funções que executam estas operações.

O vector resultante com dimensão 11×1 é o vector contendo todos os coeficientes desconhecidos:

$$(a_{11}, a_{12}, a_{13}, a_{14}, a_{21}, a_{22}, a_{23}, a_{24}, a_{41}, a_{42}, a_{43})$$

e que vão formar a matriz A de transformação pretendida:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ \cdot & \cdot & \cdot & \cdot \\ a_{41} & a_{42} & a_{43} & 1 \end{bmatrix}$$

1.3.7 Justificação do Resultado $\bar{x} = (A^T A)^{-1} A^T b$

Seja A a matriz de dimensão $m \times n$ e x a solução com dimensão n . Sendo $n < m$, o sistema representado é sobredimensionado a que corresponde a um número de

observações m superior ao número de variáveis n . É de esperar portanto que $Ax=b$ seja inconsistente, ou seja, que não exista uma solução que obedeça a todas as condições. Deste modo b não poderá ser posto como uma combinação linear das colunas de A .

O problema que se me depara é então o de encontrar uma solução \bar{x} aproximada que minimize o erro: por aplicação de A a \bar{x} , obtém-se um vector no sub-espaço das colunas de A ; esse vector deve ser tal que a sua distância a b , $\|A\bar{x} - b\|$, seja mínima. Isso só acontece quando ele constitui a projecção de b naquele sub-espaço, ou o que é equivalente, quando o vector distância é perpendicular a todos os vectores coluna que compõem A (Fig.7).

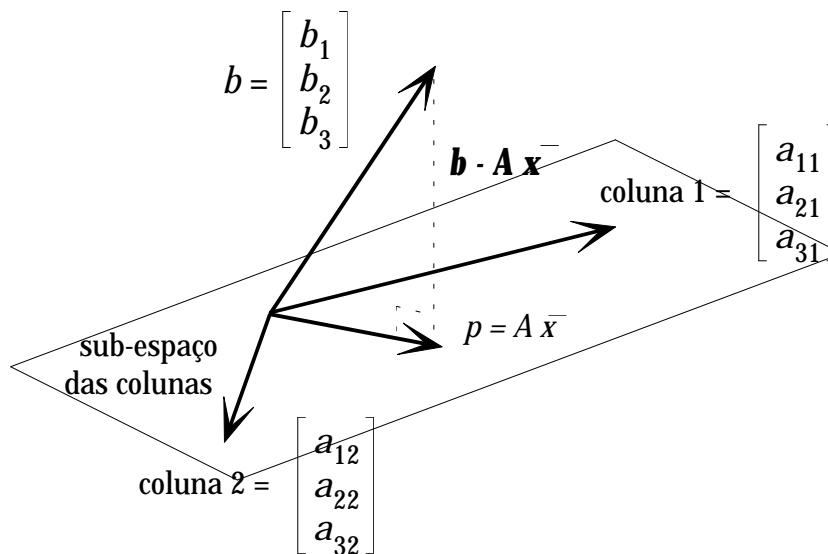


Fig. 7 - Projecção do vector 2º membro no sub-espaço das colunas de uma matriz de 3×2 .

Se o vector erro é perpendicular a todos o vectores coluna, então pode-se escrever:

$$\begin{cases} a_1^T(b - A\bar{x}) = 0 \\ \vdots \\ a_n^T(b - A\bar{x}) = 0 \end{cases} \quad \text{ou seja,} \quad \begin{bmatrix} a_1^T \\ \vdots \\ a_n^T \end{bmatrix} \begin{bmatrix} b - A\bar{x} \end{bmatrix} = 0$$

Este resultado é equivalente a: $A^T(b - A\bar{x}) = 0 \Leftrightarrow A^T A\bar{x} = A^T b$

O que dá o resultado final: $\bar{x} = (A^T A)^{-1} A^T b$

1.3.8 Cálculo da Matriz Inversa de A

Esta secção é válida tanto para a matriz de transformação (embora, como se verá adiante, a sua inversa não seja forçosamente necessária para o cálculo de (X, Y)) como para a matriz de calibração de $N \times 11$ elementos.

No primeiro caso, uma vez conhecida a matriz A de transformação (obtida experimentalmente), então ela terá que ser invertida por forma a conhecer-se (X,Y) em função de (x,y, Z) . Mas a 3ª linha é desconhecida pelo que terá que ser arbitrada.

1.3.8.1 Influência da 3ª linha na inversão de A

Consideremos a seguinte matriz A , cuja 3ª linha é composta por 4 algarismos 1:

$$A = \begin{bmatrix} 9 & -1 & 4 & 3 \\ 4 & 3 & 7 & 5 \\ 1 & 1 & 1 & 1 \\ 2 & 5 & 8 & 2 \end{bmatrix} \Rightarrow A^{-1} = \begin{bmatrix} 0,13 & -0,20 & 0,51 & 0,04 \\ -0,04 & -0,27 & 1,27 & 0,10 \\ 0,02 & 0,13 & -0,97 & 0,12 \\ -0,11 & 0,33 & 0,18 & -0,26 \end{bmatrix}$$

Consideremos outra matriz cuja 3ª linha é agora composta por algarismos 2:

$$A = \begin{bmatrix} 9 & -1 & 4 & 3 \\ 4 & 3 & 7 & 5 \\ 2 & 2 & 2 & 2 \\ 2 & 5 & 8 & 2 \end{bmatrix} \Rightarrow A^{-1} = \begin{bmatrix} 0,13 & -0,20 & 0,26 & 0,04 \\ -0,04 & -0,27 & 0,64 & 0,10 \\ 0,02 & 0,13 & -0,48 & 0,12 \\ -0,11 & 0,33 & 0,09 & -0,26 \end{bmatrix}$$

A 3ª linha só tem efeito, como se pode observar, no resultado da 3ª coluna. E o resultado da duplicação dos elementos constituintes da 3ª linha é precisamente a obtenção da metade dos anteriores elementos da coluna.

A 3ª coluna é precisamente aquela que vai actuar sobre a coordenada z da imagem que é a variável livre:

$$\begin{bmatrix} \cdot & \cdot & * & \cdot \\ \cdot & \cdot & * & \cdot \\ \cdot & \cdot & * & \cdot \\ \cdot & \cdot & * & \cdot \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Ora, os valores resultantes, kX e kY , terão que ser divididos por k de modo a obter-se (X,Y) . Mas k depende também dos valores dessa 3ª coluna pelo que a arbitrariedade na escolha da 3ª linha de A fica assim "anulada".

No Anexo A.1 é apresentado o método utilizado para inverter a matriz A .

1.3.9 Obtenção de (X, Y) a partir de (x, y, Z) - 1º processo

$$w_h = A^{-1} \cdot c_h$$

\swarrow
 real

\searrow
 câmara

Uma vez calculada a matriz inversa $B=A^{-1}$, as coordenadas X e Y podem finalmente ser obtidas:

$$B = A^{-1} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{bmatrix} \quad w_h = \begin{bmatrix} kX \\ kY \\ kZ \\ k \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = A^{-1} \cdot c_h$$

ou seja,

$$\begin{cases} kX = b_{11}x + b_{12}y + b_{13}z + b_{14} \\ kY = b_{21}x + b_{22}y + b_{23}z + b_{24} \\ kZ = b_{31}x + b_{32}y + b_{33}z + b_{34} \\ k = b_{41}x + b_{42}y + b_{43}z + b_{44} \end{cases} \quad \begin{cases} X = (b_{11}x + b_{12}y + b_{13}z + b_{14}) / (b_{41}x + b_{42}y + b_{43}z + b_{44}) \\ Y = (b_{21}x + b_{22}y + b_{23}z + b_{24}) / (b_{41}x + b_{42}y + b_{43}z + b_{44}) \\ Z = (b_{31}x + b_{32}y + b_{33}z + b_{34}) / (b_{41}x + b_{42}y + b_{43}z + b_{44}) \end{cases}$$

Todos estes valores vêm em função da variável livre z . Resolvendo a 3ª equação em função de z obtém-se:

$$Z = \frac{b_{31}x + b_{32}y + b_{33}z + b_{34}}{b_{41}x + b_{42}y + b_{43}z + b_{44}} \Leftrightarrow z = \frac{x(b_{31} - Zb_{41}) + y(b_{32} - Zb_{42}) + (b_{34} - Zb_{44})}{(Zb_{43} - b_{33})}$$

z é uma função de x , y e Z . Se a sua expressão calculada acima for substituída nas restantes duas equações, obtém-se finalmente as coordenadas reais (X, Y) .

$$\begin{cases} X = \frac{b_{11}x + b_{12}y + b_{13}z + b_{14}}{b_{41}x + b_{42}y + b_{43}z + b_{44}} \\ Y = \frac{b_{21}x + b_{22}y + b_{23}z + b_{24}}{b_{41}x + b_{42}y + b_{43}z + b_{44}} \end{cases}$$

1.3.9.1 Obtenção de (X, Y) a partir de (x, y, Z) usando cálculo matricial (2º processo)

Relembrando o resultado da aplicação de A a w_h :

$$\begin{cases} a_{11}X + a_{12}Y + a_{13}Z + a_{14}a_{21} \cdot 0 + a_{22} \cdot 0 + a_{23} \cdot 0 + a_{24} \cdot 0 - a_{41}xX - a_{42}xY - a_{43}xZ = x \\ a_{11} \cdot 0 + a_{12} \cdot 0 + a_{13} \cdot 0 + a_{14}a_{21}X + a_{22}Y + a_{23}Z + a_{24} \cdot 1 - a_{41}yX - a_{42}yY - a_{43}yZ = y \end{cases}$$

Se x for posto em evidência na 1ª equação e y na 2ª, fica,

$$\begin{cases} x(a_{41}X + a_{42}Y + a_{43}Z + a_{44}) = a_{11}X + a_{12}Y + a_{13}Z + a_{14} \\ y(a_{41}X + a_{42}Y + a_{43}Z + a_{44}) = a_{21}X + a_{22}Y + a_{23}Z + a_{24} \end{cases}$$

donde se pode tirar (x,y) :

$$\begin{cases} x = \frac{a_{11}X + a_{12}Y + a_{13}Z + a_{14}}{a_{41}X + a_{42}Y + a_{43}Z + a_{44}} \\ y = \frac{a_{21}X + a_{22}Y + a_{23}Z + a_{24}}{a_{41}X + a_{42}Y + a_{43}Z + a_{44}} \end{cases}$$

mas, pondo em evidência X e Y , obtém-se:

$$\begin{cases} X(a_{11} - a_{41}x) + Y(a_{12} - a_{42}x) = (-a_{13}Z + a_{43}xZ + a_{44}x - a_{14}) \\ X(a_{21} - a_{41}y) + Y(a_{22} - a_{42}y) = (-a_{23}Z + a_{43}yZ + a_{44}y - a_{24}) \end{cases}$$

Ora, esta disposição sugere o uso de uma operação matricial:

$$\begin{bmatrix} (a_{11} - a_{41}x) & (a_{12} - a_{42}x) \\ (a_{21} - a_{41}y) & (a_{22} - a_{42}y) \end{bmatrix} \cdot \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} -a_{13}Z + a_{43}xZ + a_{44}x - a_{14} \\ -a_{23}Z + a_{43}yZ + a_{44}y - a_{24} \end{bmatrix}$$

Ou seja,

$$M \cdot \begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} u \\ v \end{bmatrix}$$

Esta matriz opera sobre (X,Y) quando na realidade são estas as coordenadas que se pretende conhecer; mas M é uma matriz quadrada de 2×2 cuja inversa é fácil de calcular, e sendo assim posso escrever de imediato:

$$\begin{bmatrix} X \\ Y \end{bmatrix} = M^{-1} \cdot \begin{bmatrix} u \\ v \end{bmatrix}$$

considerando, $M = \begin{bmatrix} \alpha & \beta \\ \gamma & \delta \end{bmatrix}$ então, $M^{-1} = \frac{1}{\Delta} \cdot \begin{bmatrix} \delta & -\beta \\ -\gamma & \alpha \end{bmatrix}$

onde Δ é o determinante de M : $\Delta = \alpha\delta - \beta\gamma$

É claro que só é possível estabelecer a relação pretendida $(x,y) \leftrightarrow (X,Y)$ quando:

- Z for conhecido
- o determinante Δ não for nulo
- todos os 12 coeficientes forem conhecidos e correctos

$$\text{ent\~{a}o,} \quad \begin{bmatrix} X \\ Y \end{bmatrix} = \frac{1}{\Delta} \cdot \begin{bmatrix} \delta & -\beta \\ -\gamma & \alpha \end{bmatrix} \cdot \begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{\Delta} \begin{bmatrix} (\delta u - \beta v) \\ (-\gamma u + \alpha v) \end{bmatrix} \Leftrightarrow \begin{cases} X = \frac{1}{\Delta}(\delta u - \beta v) \\ Y = \frac{1}{\Delta}(-\gamma u + \alpha v) \end{cases}$$

$$\text{onde,} \quad \begin{cases} \alpha = a_{11} - a_{41}x \\ \beta = a_{12} - a_{42}x \\ \gamma = a_{21} - a_{41}y \\ \delta = a_{22} - a_{42}y \\ u = -a_{13}Z + a_{43}xZ + a_{44}x - a_{14} \\ v = -a_{23}Z + a_{43}yZ + a_{44}y - a_{24} \\ \Delta = \alpha\delta - \beta\gamma \end{cases}$$

Conhecidas estas express\~{o}es (a_{ij} s\~{a}o elementos da matriz A de transforma\~{c}o e as coordenadas (x, y, Z) s\~{a}o conhecidas), o valor final das coordenadas (X, Y) no mundo real fica perfeitamente definido.

1.4 A Montagem Experimental

A componente laboratorial deste trabalho consistiu na constru\~{c}o de uma grelha de calibra\~{c}o composta por diversos c\~{i}rculos colocados de 50 em 50 mil\~{i}metros num rect\~{a}ngulo de acr\~{i}lico de 600mm×400mm num total de 117 pontos. Essa grelha foi colocada a v\~{a}rias alturas do plano $Z=0$ com uma precis\~{a}o de 0,0005mm. Para cada posi\~{c}o era adquirida uma imagem que ap\~{o}s ser binarizada era, ou registada em disco para posterior utiliza\~{c}o, ou alvo imediato de uma an\~{a}lise de figuras para determina\~{c}o do seu n\~{u}mero e avalia\~{c}o dos respectivos centros geom\~{e}tricos. Essas coordenadas seriam depois processadas pelo software desenvolvido com vista a calcular a matriz de transforma\~{c}o A .

A Fig.8 mostra um desenho da grelha constru\~{i}da e utilizada no processo de calibra\~{c}o. A Fig.9 corresponde \~{a} imagem dada pela c\~{a}mara da grelha numa das suas posi\~{c}oes.

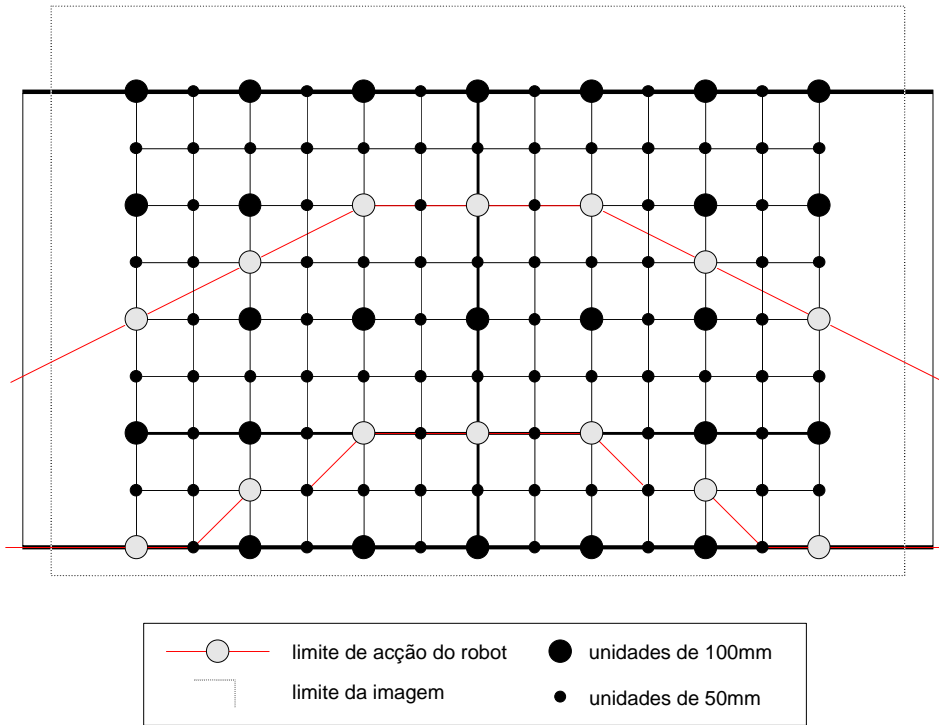


Fig. 8 - Desenho da grelha utilizada com indicação dos limites quer da acção do robot quer da imagem.

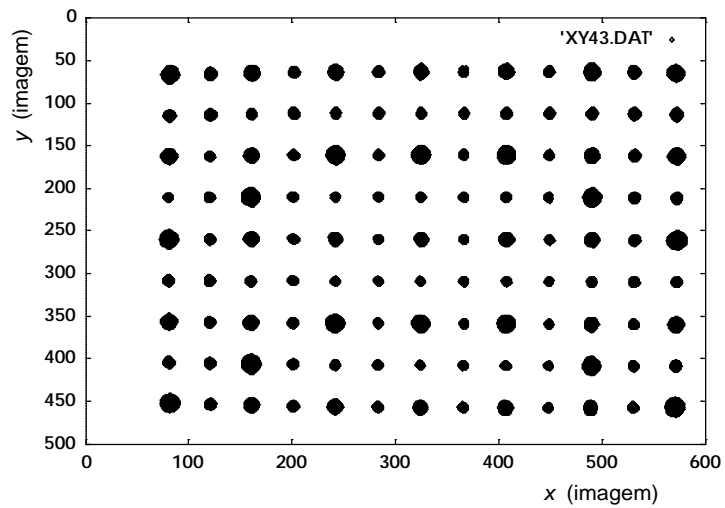


Fig. 9 - Imagem da grelha numa das suas posições de calibração.

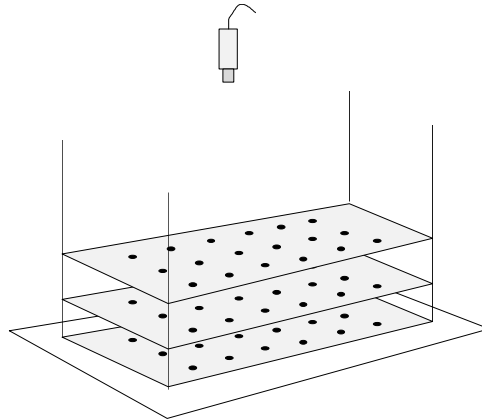


Fig. 10 - Experiência de calibração da câmara: a fig. mostra as várias posições da grelha.

O correcto posicionamento da grelha nas suas várias alturas requereu uma atenção muito especial já que um desvio sistemático nas coordenadas dos 117 pontos poderia originar uma avaliação incorrecta da matriz de transformação e assim comprometer todo o futuro processo de conversão de coordenadas. Foram portanto construídas "guias" de elevação que impediam a deslocação lateral da grelha. Estas peças metálicas, que não se encontram representadas na figura acima (Fig.10), permitiam a alteração na altura da grelha sem introduzir deslocamento na horizontal, dentro de uma tolerância de 1mm o que parece satisfatório.

No Anexo A.2 pode ser encontrada uma descrição do software desenvolvido.

1.5 Cálculo Final da Matriz A

Com o sistema implementado e fazendo uso de várias imagens binarizadas da grelha de calibração colocada a várias alturas, foi possível obter os 11 coeficientes desconhecidos da matriz A de transformação. Apresentam-se de seguida duas matrizes A . A primeira foi obtida após uma calibração normal com todos os pontos da grelha a várias alturas. A segunda matriz A foi obtida após uma calibração apenas com os 25 pontos centrais de cada grelha.

- matriz A de transformação após calibração da câmara com $3 \times 117 = 351$ pontos:

$$A = \begin{bmatrix} -0,00100747 & 0,80984232 & -0,25445335 & 326,36116263 \\ -0,98001218 & -0,00482131 & -0,03928230 & 357,17764492 \\ \cdot & \cdot & \cdot & \cdot \\ -0,00000549 & -0,00002639 & -0,00086536 & 1,00000000 \end{bmatrix}$$

- matriz A de transformação após calibração da câmara com $3 \times 25 = 57$ pontos centrais:

$$A = \begin{bmatrix} 0,00442838 & 0,82146792 & -0,32964606 & 324,79378016 \\ -0,98308730 & -0,00281919 & -0,20059689 & 358,09570302 \\ 0,00001035 & 0,00001495 & -0,00089563 & 1,00000000 \end{bmatrix}$$

1.6 Testes de Conversão com a Matriz A

Uma vez obtida a matriz de A de transformação, com a qual se obtém a conversão de coordenadas (x,y, Z) para (X,Y,Z) , houve que efectuar testes para verificar a correcta conversão daquelas. Para tal, melhor objecto não podia ser utilizado que a própria grelha de calibração colocada a uma dada altura do plano $Z=0$, já que as coordenadas reais são conhecidas antecipadamente com rigor e podem por isso ser comparadas de imediato às obtidas por conversão.

Apresento de seguida dois testes de conversão nos quais foram utilizadas duas matrizes de transformação obtidas de modos diferentes e com objectivos diferentes. Numa primeira situação, os pontos conhecidos foram comparados com os obtidos a partir de uma matriz A calculada com todos os pontos da grelha a 3 alturas diferentes, totalizando $3 \times 117 = 351$ pontos. Nesta abordagem a matriz foi calculada de forma a minimizar o erro de conversão em todos aqueles pontos pelo que os erros observados de conversão são sensivelmente os mesmos para todos eles como se pode observar na Fig.11.

Numa segunda situação, foram também utilizadas 3 alturas diferentes da grelha mas apenas os respectivos pontos centrais foram usados na calibração. O resultado foi uma matriz A que conseguia reduzir, ainda mais do que anteriormente, o erro no cálculo das coordenadas, mas apenas para aqueles pontos. A sobreposição dos pontos calculados com os reais evidencia então a distorção radial que caracteriza o sistema óptico utilizado na câmara, resultado da sua aberração esférica (Fig.12).

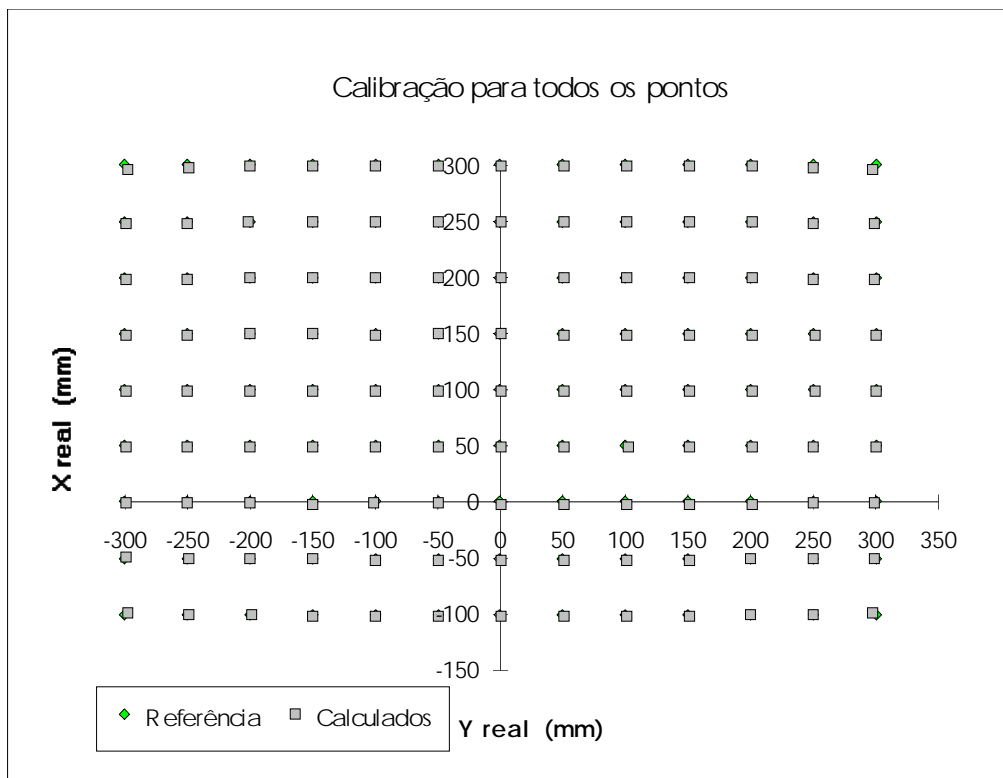


Fig. 11 - Sobreposição dos pontos de referência com os obtidos por conversão. (calibração executada para todos os pontos da grelha)

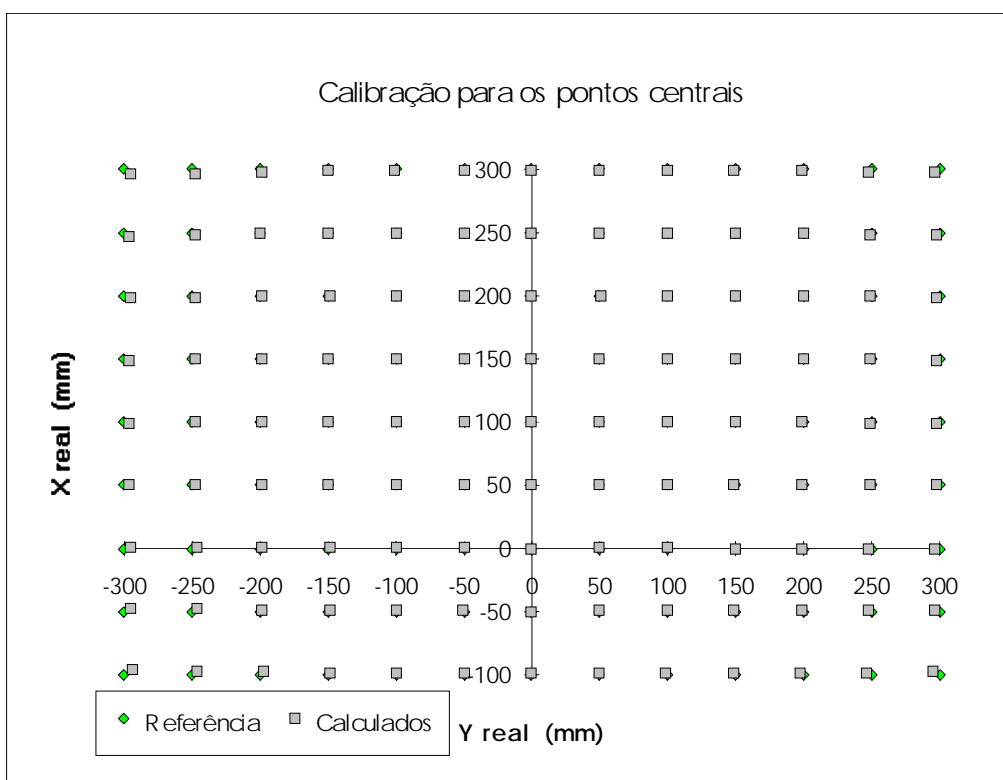


Fig. 12 - Sobreposição dos pontos de referência com os obtidos por conversão. (calibração executada apenas para os pontos centrais)

A seguinte tabela evidencia a redução referida no erro da conversão para pontos centrais:

		calibração normal		calibração para os pontos centrais	
X (mm)	Y (mm)	ΔX (mm)	ΔY (mm)	ΔX (mm)	ΔY (mm)
50	-50	1,3	-0,7	-0,2	-0,1
0	-50	1,6	-0,2	-0,3	-0,2
-50	-50	1,7	-0,6	-0,7	0,2
-50	0	1,9	-0,3	-0,2	0,5
-50	50	2,0	-0,8	-0,4	0,6
0	50	1,9	-0,9	-0,1	-0,2
50	50	1,4	-1,6	-0,1	0,0
50	0	1,1	-0,8	-0,5	0,2
0	0	1,7	-0,3	0,1	0,1

Como se pode observar, os erros de conversão deixam de ser da ordem dos milímetros e passam a ser da ordem das décimas de milímetro: uma ordem de grandeza inferiores.

Pelo contrário, os pontos mais afastados vão estar associados a erros maiores:

		calibração normal		calibração para os pontos centrais	
X (mm)	Y (mm)	ΔX (mm)	ΔY (mm)	ΔX (mm)	ΔY (mm)
300	-300	3,1	-2,2	3,7	-4,2
-100	-300	-1,3	-2,9	-4,5	-6,5
-100	300	-0,9	1,7	-2,6	4,5
300	300	3,3	1,7	2,4	3,1
300	0	0,7	-0,5	0,6	0,5

Esta calibração é contudo a mais vantajosa uma vez que a maioria dos objectos a manipular estarão mais localizados na zona central do que na periferia.

1.7 Conclusão

Com este trabalho experimental conseguiu-se obter um processo de conversão de coordenadas da imagem para coordenadas reais no sistema de referência do robot, com grande precisão: o erro obtido foi sempre inferior a 2mm, e no caso especial da calibração para pontos centrais esse erro foi reduzido para algumas décimas de milímetro, apresentando em geral valores inferiores a 0,2mm.

Embora o processo de conversão a 2D tivesse sido mantido na análise de figuras bidimensionais efectuado pelo agente Visão, o processo a 3D foi preferido sempre que a 3ª dimensão Z estava disponível, como é o caso das análises efectuadas pelo agente Laser.

2.1 Introdução

Tal como foi apresentado na introdução desta dissertação, toda a análise bidimensional da imagem que envolve medidas geométricas várias, quer relacionadas com as superfícies visíveis dos objectos quer das respectivas fronteiras, é suposta ser efectuada separadamente numa máquina à parte. Todos os algoritmos desenvolvidos foram então incorporados num pacote separado, sujeito a um esquema independente de pedidos e de envio de resultados e ao qual se resolveu designar Agente Visão. Em complementaridade a este, existe um outro que também possui um esquema separado de comunicação com a estação de trabalho e de recepção de pedidos de acção, que se designou por Agente Laser por envolver um dispositivo original construído pelo autor do presente trabalho no Laboratório de Robótica e Inteligência Artificial da Faculdade de Engenharia da Universidade do Porto, que utiliza uma fonte laser.

O Agente Visão escolhe os algoritmos a utilizar consoante as necessidades, assim como a sequência correcta das suas chamadas.

No Anexo B encontra-se a descrição dos programas desenvolvidos.

2.2 Explicação Breve sobre a Abordagem a Utilizar no Processo de Identificação a 2D

Mediante pedidos expressos feitos por qualquer dos restantes agentes, este agente efectua um de dois tipos de análise bidimensional. Naquela que se refere às superfícies visíveis dos objectos, será possível, para além da identificação, obter as respectivas localizações e orientações. Estes valores são dados a conhecer ao agente planeador de acções que conjuntamente com as respectivas identidades permitem construir um plano de acção, constituído por acções intercalares de manipulação, com o objectivo de atingir uma determinada montagem final. Para que os dados sejam correctos é obrigatório (pelo menos para esta 1ª opção do Agente Visão) que os objectos

se encontrem devidamente separados, sem sobreposições, e perfeitamente justapostos na mesa de suporte. Qualquer sobreposição originará uma imagem conjunta dos dois objectos em sobreposição que será interpretada como um só. Por outro lado, qualquer rotação no espaço fará com que a imagem venha alterada pelo que não será possível reconhecê-la. Impõe-se portanto que os objectos se encontrem também num dos seus estados estáveis (Fig.1).

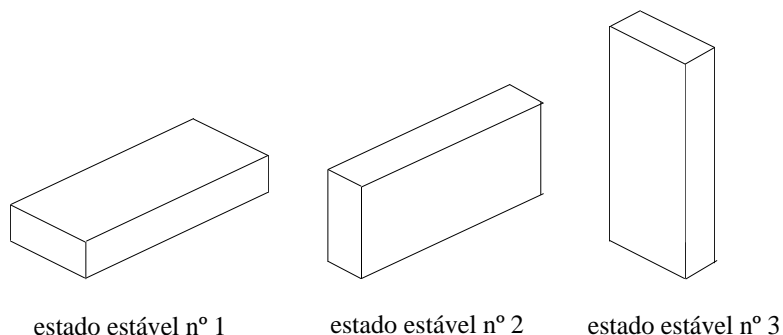


Fig. 1 - Apresentação de todos os estados estáveis de um paralelepípedo.

Garantidas estas condições, o algoritmo será capaz de processar imagens de objectos com qualquer forma e grau de complexidade (Fig.2).

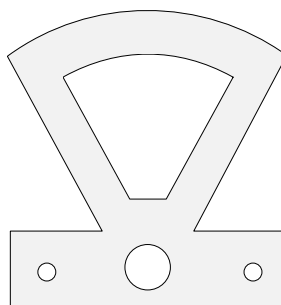


Fig. 2 - Objecto com alguma complexidade utilizado em experiências de identificação.

No 2º processo de identificação utilizado neste agente são já impostas algumas restrições uma vez que se baseia na análise de segmentos de fronteira. Sendo assim o objecto da Fig.2 seria processável por ele mas a maior parte dos da Fig.3 já não o seria. Num futuro desenvolvimento deste processo de análise que envolverá arcos de circunferência e de elipses, tal já será possível.

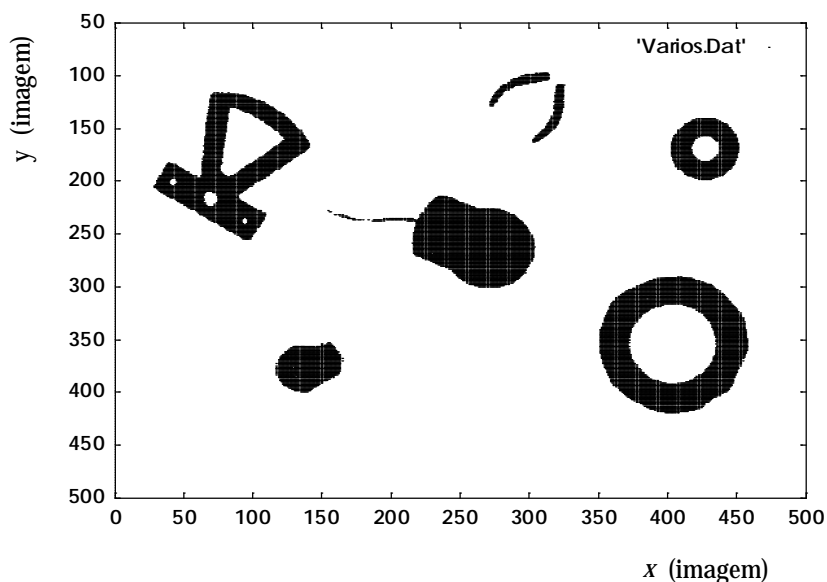


Fig. 3 - Imagem binarizada de um conjunto de objectos, a maior parte dos quais de uso corrente.

2.3 Descrição do Sistema do Agente Visão

Este é o programa principal do Agente Visão que gere todos os algoritmos nele contidos, desenvolvidos para efectuar análises da cena a duas dimensões (2D). Para tal dispõe de uma imagem adquirida por uma câmara vídeo Sony provida de uma objectiva com distância focal $f=12,5\text{mm}$. Essa imagem é amostrada espacialmente em intensidade e registada em memória na placa de aquisição de imagem da Imaging Technology, numa matriz de 640×512 pixels. O valor registado é o da intensidade luminosa recebida, codificada em nível de cinzento com 8 bits de resolução, ou seja, possuindo valores entre 0 e 255.

Atendendo a que aquela placa é suposta ser utilizada por dois agentes distintos, nomeadamente o Agente Visão a ser descrito e o Agente Laser, há que elaborar um processo de evitar o seu acesso simultâneo. Contemplando a possibilidade de ambos os agentes correrem na mesma máquina, mais concretamente em duas janelas distintas de um ambiente Windows num mesmo PC, foi procurada uma zona de memória RAM que pudesse ser visível por ambas e onde se pudessem registar bytes de controle. Foram então utilizadas duas posições de memória que foram internamente associadas a duas variáveis, que identificam qual o agente que se encontra a utilizar a placa:

flag_Vision : endereço $\rightarrow 0 \times 9\text{FFFE}$

flag_Laser : endereço $\rightarrow 0 \times 9\text{FFFF}$

A variável que exiba o valor 1 indica que o agente que lhe corresponde está a utilizar a placa. Essa variável é posta naquele valor pelo próprio agente, na condição da outra se

encontrar com o valor 0. Logo que o agente termine a sua execução, põe de imediato a variável que lhe corresponde a 0 por forma a disponibilizar a placa para o outro agente. No caso presente, a variável é *flag_Vision* que deve ser posta a 1 sempre que a execução se inicie. Esta, como já foi dito, só se inicia se *flag_Laser* for igual a 0. No fim, *flag_Vision* é reposta a 0.

Uma outra função deste programa é o "display" do ecrã de apresentação no qual é indicado o título do software, a sua autoria e o local em que ele foi desenvolvido. Num segundo ecrã que segue imediatamente o primeiro, são indicadas as duas vertentes possíveis do agente correspondente às duas opções *Blob* e *Hough*, uma janela para mensagens e o estado corrente do agente: em execução, em espera ou parado.

Este ecrã poderá ser apagado durante aquela execução por necessidade de utilizar todo o ecrã para o registo de mensagens e pedidos de escolha de opções várias, ou ainda para representações gráficas (como é o caso dos resultados de *Hough*), mas é sempre mostrado de novo logo que ela termine.

O grande objectivo do agente é contudo a localização e identificação a 2D de todos os objectos presentes na cena, e a sua função consiste então e principalmente no lançamento dos algoritmos desenvolvidos para esse fim. Para tal ele deve detectar a existência de um pedido por parte de um qualquer agente, recebê-lo e executar os programas correspondentes.

Os pedidos são registados num ficheiro que reside numa Estação de Trabalho DEC Station 5000/240 com sistema operativo Ultrix. Essa estação de trabalho encontra-se ligada a uma rede Ethernet à qual também se encontra ligado o PC onde funciona o Agente Visão. O ficheiro é acedido por FTP (Fig.4).

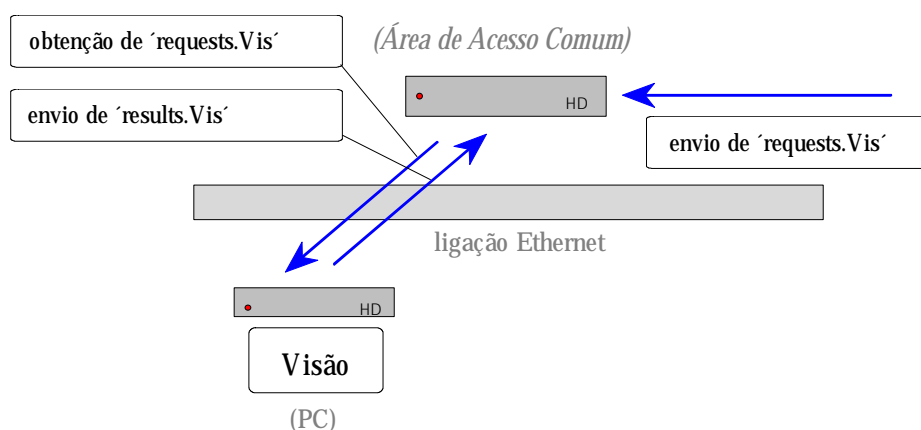


Fig. 4 - Esquema de transferência de ficheiros de e para a zona de acesso comum.

As opções neste agente são duas, constituindo a primeira delas a principal, com resultados mais completos e também mais utilizada (Tabela 1):

Tabela 1

Nº dos pedidos	Código	Significado	Parâmetros
1	Blob	Blob	(não possui)
2	Hough	Hough	(não possui)

As respectivas acções são descritas no quadro apresentado na Tabela 2:

Terminada a execução ele deve enviar os resultados da sua acção para a zona de acesso comum onde o agente responsável pelo pedido os irá recolher. O ficheiro onde os resultados são registados é enviado para a mesma Estação de Trabalho citada anteriormente.

Um exemplo daquele ficheiro poderia ser o apresentado abaixo:

ficheiro de resultados

```
position(balança , 14, 91, 135.6, 35.7, 0, 1.859456).
position(nova_caixa , 13, 80, 83.6, -26.9, 0, 0.823795).
position(fita_adesiva , 17, 91, 152.8, 23.9, 0, 3.113318).
```

onde os valores que são exibidos após o nome de cada modelo se referem respectivamente, ao nº do modelo na base de dados, à probabilidade em percentagem de corresponder ao objecto na cena, às coordenadas (X,Y,Z) do centro geométrico no sistema de referência do robot em mm, e finalmente ao ângulo, também naquele sistema, em radianos.

Tabela 2

Código	Acção
Blob	<p>Esta é a opção mais utilizada e consiste na análise das características geométricas das figuras ou "blobs" contidas na imagem bidimensional que se encontra registada na placa de aquisição de imagem. Para tal efectua uma sequência de acções que levarão à detecção daquelas figuras em termos informáticos, na medida das várias localizações, orientações, características geométricas várias e finalmente na sua identificação.</p> <p>A acção é iniciada pela construção de um histograma de intensidade (0~255), pela escolha automática de um nível intermédio entre as intensidades do fundo e dos objectos e pela binarização da imagem tendo como limiar aquele valor. Os valores registados nos pixeis passam a constar apenas de dois valores: 0 e 255.</p> <p>Logo de seguida é executado o algoritmo de análise geométrica de figuras, 'blob.c', de que resulta uma lista de objectos sob a forma de vector, que vão conter todos os parâmetros julgados necessários para a obtenção dos resultados acima referidos. Segue-se de imediato a tentativa de identificação desses objectos aos quais são atribuídos coeficientes de comparação ou "matching" como medida da certeza na sua identidade.</p> <p>Finalmente é efectuado um registo local com os resultados numa codificação compreensível para os demais agentes.</p>
Hough	<p>Nesta opção é chamada a execução da transformada de Hough modificada, com pós-processamento seguida do algoritmo de encadeamento de segmentos e ângulos internos que constituem aquilo a que designei como "chain-code".</p> <p>Esta análise consiste na detecção robusta de todos os segmentos de recta que definem a fronteira dos vários objectos (linhas curvas não são contempladas) existentes na cena. De seguida esses segmentos são agrupados por objecto, medidos, e encadeados, sendo também medidos os ângulos que eles formam entre si. As várias codificações de encadeamento vão servir para caracterizar os objectos sendo por isso construídas bases de dados. O objectivo desta opção é oferecer uma alternativa no processo de identificação, principalmente quando os objectos se encontrem encostados ou parcialmente sobrepostos. Neste último caso por exemplo, podem obter-se sequências de segmentos pertencentes à parte visível de cada objecto que se verifique estarem contidas em codificações segmento-ângulo como as referidas atrás, de determinados modelos pré-registados nas bases de dados. Nesse caso, podem obter identificações apesar de não existir informação completa dos objectos.</p>

2.4 Conclusão

Os resultados obtidos por este agente serão da maior importância para a elaboração do plano final de montagem num qualquer processo de robótica de manipulação e/ou de montagem, já que fornecem a localização, orientação e identidade dos objectos visíveis pela câmara e destinados a ser utilizados nessa tarefa. Apesar de impor a pré-condição da separação dos objectos para obter sucesso, este agente apresenta a grande vantagem da sua rápida execução, ao contrário do Agente Laser. Este último, apesar de dispensar aquela condição possui um tempo de execução que pode atingir os minutos, enquanto que a operação do Agente Visão não excede alguns segundos.

(Os tempos apresentados referem-se ao processador utilizado: um 80 486 da Intel; eles poderão vir a ser substancialmente reduzidos para o caso do Pentium/133 já disponível no mercado e a preço acessível à data da publicação desta Dissertação).

Os capítulos seguintes farão uma análise detalhada de qualquer dos algoritmos que compõem este agente.

3.1 Introdução

Esta é a 1ª das duas opções de execução do Agente Visão e consiste na análise geométrica das figuras visíveis na imagem (em oposição à 2ª opção referida em que é feita a análise das respectivas linhas de fronteira). A sua execução é iniciada logo que no ficheiro de pedidos seja lido o código correspondente: "Blob".

3.2 A Binarização das Imagens de Intensidade

A análise geométrica das figuras presentes numa imagem pressupõe, como se verá na secção seguinte, uma prévia binarização desta. Esta operação destina-se essencialmente a separar aquelas figuras do fundo da imagem, fazendo-as salientar por atribuição do nível máximo de cinzento: o 255. Em oposição, todos os pixels pertencentes ao fundo são postos a 0. Esta binarização prévia pode também, ser utilizada para seleccionar determinado ou determinados objectos pretendidos em detrimento de outros a ser ignorados. É o caso da análise das imagens de alturas pelo Agente Laser, quando aquelas codificam a altura a que os diversos pontos dos objectos se encontram da mesa de suporte, em que se pode pretender seleccionar apenas objectos situados acima de uma determinada altura. Na análise bidimensional que será feita aqui, contudo, não será desprezado nenhuns dos objectos presentes na imagem, mesmo quando a figura de que se dispõe resulte da sobreposição de dois deles.

No caso de ocorrerem aquelas sobreposições, será tentada uma outra abordagem ainda a 2D (e antes mesmo dos algoritmos a 3D) apresentada noutro capítulo mais adiante. Neste, pressupõe-se que todos os objectos se encontram devidamente separados de forma a ser possível medir as respectivas características geométricas, que posteriormente permitam conhecer quer a sua posição e orientação quer a sua identidade.

Por outro lado não é feita qualquer análise de reflectância ou de textura, pelo que as imagens adquiridas directamente em níveis de cinzento não serão alvo de qualquer análise.

Para se conseguir efectuar a binarização há que separar os níveis de cinzento associadas ao fundo e aos objectos. Isto pressupõe a utilização de um fundo de diferente tonalidade: por exemplo um fundo escuro. Desta forma, praticamente todos os objectos (com a evidente excepção de objectos igualmente escuros) ficarão salientados contra esse fundo pelo que será fácil conseguir a separação referida. Resta conhecer o limiar de nível de cinzento que efectua aquela separação, pelo que se deve proceder à análise do histograma da imagem, ou seja, contar o nº total de pixels associados a cada um dos 256 níveis de cinzento. Como se pode observar na Fig.1, a maior parte dos pixels possuem um nível baixo – vão corresponder ao fundo – enquanto que apenas alguns possuem níveis mais elevados – vão corresponder aos pixels dos objectos – formando duas elevações distinguíveis. Para esta análise foi utilizada uma imagem em níveis de cinzento cuja versão binarizada é mostrada na Fig.15. A Fig.2 apresenta uma ampliação do histograma que permite visualizar melhor estes últimos. O limiar de binarização poderá então ser qualquer nível entre estas duas elevações: todos os pixels de nível superior ao limiar serão postos a 255 enquanto que os restantes são postos a 0.

Para facilitar esta tarefa (que deverá ser executada automaticamente pelo programa sem a intervenção do utilizador) aquele histograma é ainda sujeito a uma suavização após o que se obtém a curva mostrada nas Figs. 3 e 4.

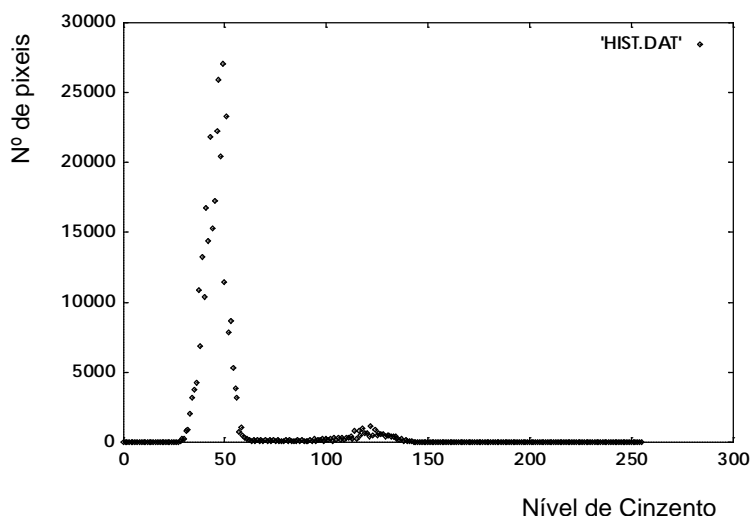


Fig. 1 - Histograma de uma imagem de intensidade no qual é visível o registo dos pixels do fundo em torno do nível ~49.

O programa de escolha do limiar apenas terá que detectar o ventre ladeado pelas duas elevações e escolher um valor intermédio.

Após execução do programa, aquele limiar ficou estabelecido em 74 (Fig.4), pelo que 308.899 pixels do fundo tomaram o valor 0 e 18.781 pixels pertencentes aos objectos tomaram o valor 255. Os pixels totalizam 327.680 o que confirma a dimensão da imagem: 512×640.

A imagem está agora pronta para análise e medida das características geométricas.

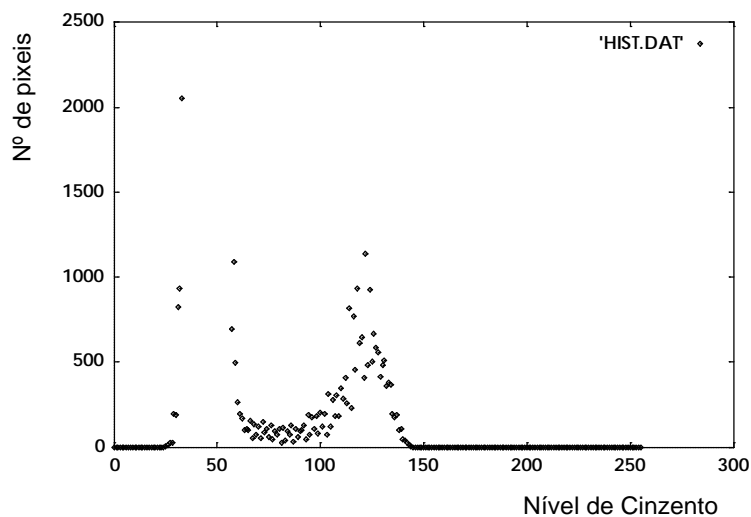


Fig. 2 - O histograma ampliado para melhor visualização do registo referente aos pixels dos objectos com níveis de cinzento superior a ~70.

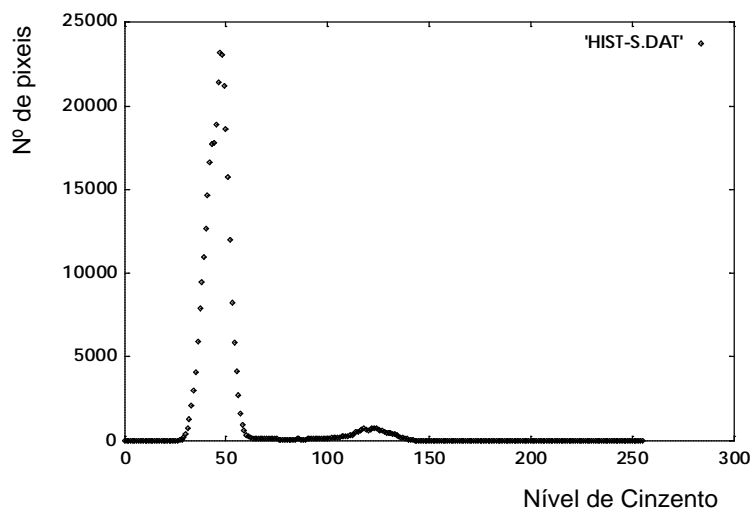


Fig. 3 - Resultado da suavização do histograma anterior, no qual são bem visíveis as duas elevações correspondentes ao fundo e aos objectos.

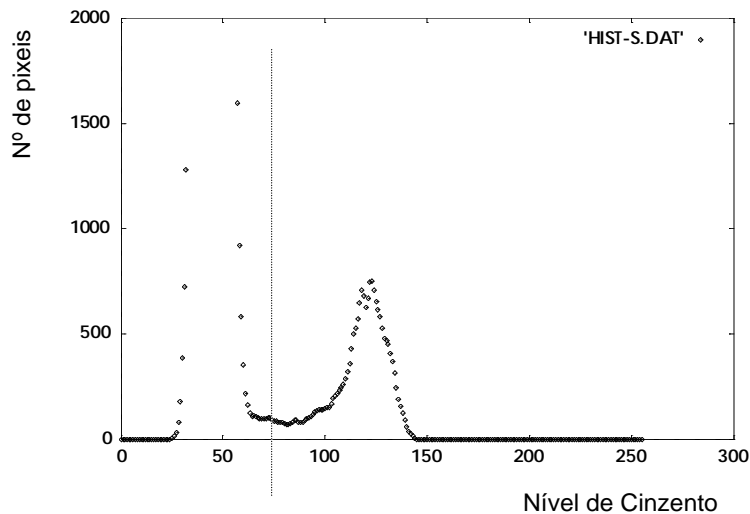


Fig. 4 - Ampliação do histograma suavizado. A linha tracejada indica o limiar de binarização calculado.

3.3 A Construção dos "Blobs"

Aqui pretende-se efectuar a detecção de figuras presentes numa imagem binarizada ("blobs") e medição das suas várias características geométricas [Cunningham81].

A análise da imagem é feita por varrimento da esquerda para a direita e de cima para baixo, durante a qual pixels do mesmo tipo (0 ou 1) na horizontal são agrupados em conjuntos e verificada a sua conexão com figuras já encontradas acima. Se qualquer um dos pixels desse conjunto tiver ligação com qualquer outro das linhas encontradas em cima, então é considerado como pertencendo-lhe e os valores geométricos são devidamente actualizados. Esses conjuntos, quando detectados, são parametrizados num vector de estruturas cujos campos são a abcissa do 1º pixel, o seu comprimento e o número da figura ("blob") à qual está associado.

Durante o varrimento são mantidas actualizadas duas variáveis que consistem em elementos de entrada para aquele vector e que indicam a posição (naquele vector) da informação quer do conjunto de pixels da linha imediatamente acima quer do de baixo. O objectivo é conhecer em qualquer instante qual o nº da figura que se encontra em cima (à medida que o vector vai sendo alargado com informação da linha que está a ser considerada, com a 2ª variável).

É então efectuado o varrimento da imagem, mantendo sempre o registo das 2 variáveis até que seja detectada uma transição na linha de cima ou na de baixo, altura em que é identificado o tipo dessa transição, sendo então efectuados os cálculos e actualizações necessários, em conformidade.

Essas transições correspondem à detecção do início ou fim dos conjuntos de pixels consecutivos e ela é efectuada com o auxílio de uma janela de 2x2 que percorre a imagem da forma já indicada atrás. Quando essa janela contém apenas 0's ou 1's isso significa que se está a efectuar o varrimento dentro de uma figura ou "blob", seja ela correspondente a um objecto, a um seu buraco ou à imagem do fundo. Nesse caso o

varrimento prossegue pois nada há a assinalar. Se ela contiver 1's em cima e 0's em baixo (ou o contrário), então o varrimento estará a ser efectuado através de uma fronteira entre figuras, mas não corresponde a nenhuma transição pelo que nenhuma acção é efectuada. Para outras combinações, porém, há que identificar o tipo de transição e chamar uma função adequada que desencadeie as acções necessárias de actualização de parâmetros geométricos e de alteração do conteúdo das várias variáveis de controle. As funções a que se faz referência são chamadas de *statei()*, sendo *i* o indicador da situação detectada, e serão descritas em pormenor na secção seguinte.

A cada figura encontrada estará associado um elemento de um outro vector. Nele são registados valores geométricos que vão sendo actualizados à medida que novos conjuntos de pixeis a ela pertencentes vão sendo considerados, assim como a relação de parentesco com outras figuras. Nessa estrutura estão também incluídos campos que são preenchidos apenas posteriormente pois são calculados a partir das características medidas. Os seus valores terão um significado especial que será apresentado mais adiante.

3.3.1 A Função de Leitura de uma Linha da Imagem

Esta função tem como objectivo a leitura para um "buffer" em memória do conteúdo de uma linha da imagem para posterior análise e agrupamento dos pixeis do mesmo tipo. De início são lidos os valores em nível de cinzento. Ora como a imagem é suposta ter sido binarizada previamente, pois o algoritmo só lida com esse tipo de imagem, os pixeis só poderão ter valores 0 ou 255. E sendo assim, para se verificar se o valor associado a um dado pixel é 1 ou 0, basta analisar o bit 7 do inteiro. Para tal utiliza-se uma máscara binária com o valor $1000\ 0000_b$. O pixel é então comparado com aquela máscara através da operação binária AND de que resulta um dos valores $1000\ 0000_b$ ou $0000\ 0000_b$ consoante o seu valor seja 255 ou 0. Logo de seguida o bit 7 é deslocado 7 posições para a direita por forma a obter-se o valor final 1 ou 0, após o que o resultado é registado.

O resultado final desta operação a toda a linha lida é a substituição dos valores 255 por 1 de forma a obter-se os conjuntos consecutivos de 1's e 0's que são alvo de análise no algoritmo.

Finalmente é de referir que a binarização da imagem com uso de um nível previamente escolhido de limiar poderia perfeitamente ser feita dentro da função de leitura da linha, mas o utilizador/programador não teria acesso ao aspecto da imagem binarizada; mais concretamente não teria possibilidades de verificar se essa binarização era a pretendida. Sendo assim é preferível efectuar essa binarização previamente, executando aquela função apenas para a leitura da linha e a transformação dos valores binarizados em 1's e 0's.

3.3.2 As Funções Detectoras de Transições *statei()*

Para se poder chamar a função adequada à situação detectada, houve que associar o conteúdo da janela de 2×2 a um algarismo *i* que fosse único para essa janela. Ora isso é

conseguido somando o conteúdo da janela, pesando devidamente os pixels consoante a sua posição dentro dela, como se mostrará já de seguida no ponto 0.

Os pontos seguintes (1, 2, ..., 6) descrevem as acções a tomar para cada uma das situações características detectadas. A escolha da acção será feita consoante o valor obtido para aquele algarismo.

0. O número associado à janela e que vai dar origem à chamada de uma das funções $state_i()$ é calculado com base no valor de cada um dos pixels da janela 2x2 como se mostra de seguida:

d	b
c	a

nº da função $state_i()$: $i = 8a + 4b + 2c + d$

De acordo com o resultado i é então escolhida a função a ser chamada.

1. $state7()$:

state 7

1	1
1	0

Neste caso trata-se do início de um novo "blob" pelo que na função que lhe corresponde, uma variável de controle é posta a 1. É também o início de uma nova fronteira horizontal assim como de um novo conjunto de pixels do mesmo tipo (0 ou 1); sendo assim a abcissa i é registada numa outra variável para uso posterior na função de actualização dos parâmetros geométricos.

2. $state4()$:

state 4

0	1
0	0

Embora não se trate de um novo conjunto de 1's ou 0's, esta situação marca o início de uma fronteira horizontal pelo que a 2ª variável referida acima é também igualada a i . Como há a necessidade de conhecer o nº do "blob" a que o pixel 1 pertence, ele

é procurado no vector que regista os conjuntos de 0's e 1's e registado numa 3ª variável para posterior uso.

3. *state3()*:

state 3

1	0
1	0

Detecção do fim de um conjunto de pixeis do mesmo tipo (0 ou 1). É chamada a função de actualização dos parâmetros geométricos. De seguida é procurado o nº do "blob" a que corresponde o 0 em **b** no vector que contém os conjuntos de 0's e 1's. Ele é devidamente registado noutra variável para posterior uso. Ainda, uma vez que ambos os zeros pertencem ao mesmo "blob", a variável que contém o nº do blob corrente (0 de **a**) é posta igual ao do "blob" de **b**.

4. *state2()*:

state 2

0	0
1	0

Mais uma vez é detectado o fim de um conjunto de 1's (ou 0's). Se a variável de controle referida em **2**. é igual a 1 então este é o 1º conjunto de pixeis detectados de um novo "blob", pelo que é considerada uma nova entrada no vector de "blobs" e registados os parâmetros geométricos desse conjunto. O nº dessa entrada é passada à variável que contém o nº do "blob" corrente, e o valor da variável de controle é repostado a zero. Uma vez que os pixeis de **c** e **d** são diferentes, o que implica que os dois conjuntos de pixeis em cima e em baixo são diferentes, há ainda que somar aos perímetros dos "blobs" correspondentes o contributo para o perímetro medido na horizontal que não é contemplado na função de actualização dos parâmetros geométricos. Por fim, atribui-se à variável que contém o nº do "blob" corrente, o valor do de cima (**b**) uma vez que o zero do conjunto seguinte de pixeis (em **a**) pertence ao mesmo "blob".

5. *state6()*:

state 6

0	1
1	0

Detecção de uma nova interrupção de pixels do mesmo tipo (linha de baixo). Se se tratar de um novo "blob" (variável de controle =1) então é considerada uma nova entrada no vector de "blobs" e ordenada a actualização dos parâmetros geométricos na sua estrutura. Mais uma vez são somados aos perímetros dos dois "blobs" (associados a **c** e **d**) o valor referente ao comprimento comum da fronteira horizontal entre eles. Atendendo à convenção de conectividade adoptada, o 0 em **a** é considerado como pertencente ao mesmo "blob" de **d** pelo que o nº do "blob" corrente toma o valor do de **d**. Mas o nº do "blob" acima muda pelo que é novamente procurado no vector de conjuntos de 1's e 0's e registado. De referir ainda que o nº encontrado pode já não ser válido por haver ocorrido uma fusão entre "blobs" pelo que antes desse registo ser efectuado é procurado o nº válido a que o antigo "blob" agora pertence com uma função própria (este procedimento é efectuado sempre que há uma consulta ao vector de 1's e 0's). A obtenção desse novo nº é sempre conseguida graças ao registo efectuado num outro vector auxiliar, sempre que ocorre uma fusão: se ela se verificar entre os "blobs" **p** e **q** e o primeiro for mantido (o que implica o abandono do segundo) então *vector_aux[q]* toma o valor de **-p**, indicando o sinal negativo que o "blob" **q** deixou de ser válido e que foi incorporado no "blob" **p**.

6. *state1()*:

state 1

1	0
0	0

Este caso é o mais complexo e contempla 3 situações diferentes possíveis. Se a variável de controle referida em 2. for 1 então o que foi originalmente considerado como um novo "blob" deve fundir com outro já existente (Fig.5).

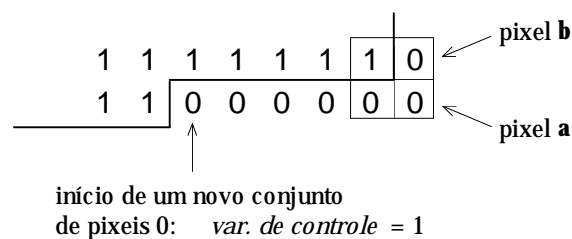


Fig. 5 - O novo conjunto de pixels consecutivos pertencem afinal a um outro "blob" já considerado na linha anterior. O nº do "blob" em **a** toma então o valor do "blob" em **b**.

No caso contrário (variável de controle =0) ou dois "blobs" diferentes devem-se fundir num só, ou o "blob" fechou-se sobre si mesmo englobando um outro a que pertence o pixel **d**. Mais concretamente, se o nº do "blob" associado ao pixel **b** for o mesmo daquele associado a **c**, então este cercou completamente o "blob" associado a **d**, ficando este a ser um buraco daquele (Fig.6). Em termos hierárquicos pode-se afirmar que ele passou a ser um "filho" daquele pelo que o campo na estrutura de dados associada ao "blob" recebe o seu número.

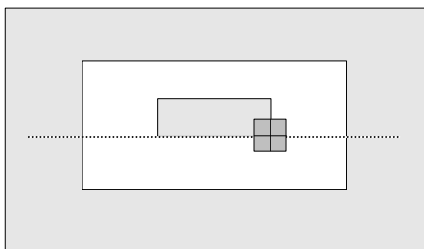


Fig. 6 - Quando o nº do "blob" associado ao pixel **c** é igual ao do pixel **b**, então o "blob" associado ao pixel **d** foi completamente circundado por aquele. Estabelece-se então uma relação de parentesco entre os dois "blobs".

Do mesmo modo, o "blob" acabado de circundar regista o nº do "blob" pai num outro campo. Atendendo a que o 1º "blob" a ser considerado é o do fundo da imagem, então todos os "blobs" presentes se encontram ligados por estas relações de parentesco, constituindo uma estrutura em árvore. Na situação mais simples em que não se detectam buracos nos vários objectos presentes na imagem, a árvore é constituída apenas por um "blob" pai (fundo da imagem) e um conjunto de filhos em nº igual ao dos objectos. Embora todos eles sejam "filhos" do "blob" nº1, apenas um dos "filhos" se encontra ligado a ele pelo registo no campo próprio. Todos os outros são relacionados com ele pelo estabelecimento de relações de "irmandade". Ou seja, uma vez encontrado o primeiro "filho" basta seguir os sucessivos conteúdos daquele novo campo para se conhecer o conjunto completo de "filhos". O último deles regista o valor -1.

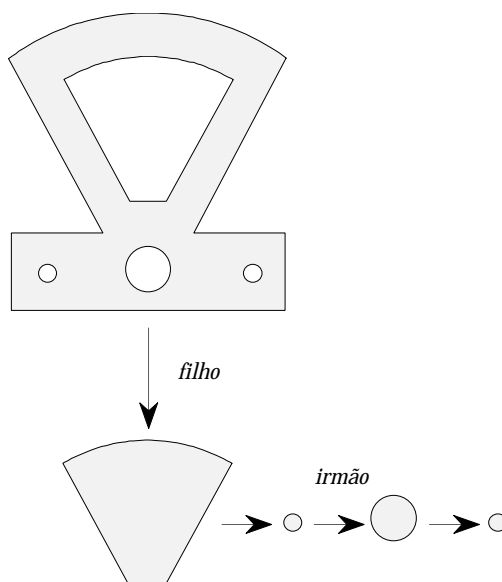


Fig. 7 - A estrutura correspondente ao objecto mostrado, possuirá num campo o nº do 1º "blob" filho. Este por sua vez terá num outro campo o nº de apenas 1 "blob" irmão. A lista completa dos buracos do objecto poderá ser obtida seguindo os conteúdos dos vários campos consecutivos do 2º tipo.

Se pelo contrário os números dos dois "blobs" forem diferentes, então o conjunto corrente de pixels 0 ou 1 liga-os, pelo que se deve fundi-los. Os parâmetros geométricos do "blob" corrente são então actualizados por acréscimo do parâmetros do de **b**, após o que este é invalidado através do classificador de "blobs" (vector auxiliar referido em 5.)

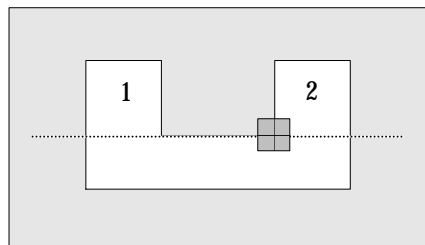


Fig. 8 - Se o nº do "blob" associado ao pixel **c** for diferente do de **b**, então ambos os "blobs" etiquetados como **1** e **2** devem ser fundidos.

Como é óbvio os dois estados para os quais as respectivas janelas 2×2 se obtêm uma da outra a partir da troca de pixels 0 e 1 (Fig.9), são equivalentes e por isso tratados na mesma função. É o caso dos pares de estados (0,15), (1,14), (2,13), (3,12), (4,11), (5,10), (6,9) e (7,8).

<i>state 4</i>	<i>state 11</i>								
<table border="1" style="border-collapse: collapse; width: 40px; height: 40px;"> <tr> <td style="text-align: center; padding: 5px;">0</td> <td style="text-align: center; padding: 5px;">1</td> </tr> <tr> <td style="text-align: center; padding: 5px;">0</td> <td style="text-align: center; padding: 5px;">0</td> </tr> </table>	0	1	0	0	<table border="1" style="border-collapse: collapse; width: 40px; height: 40px;"> <tr> <td style="text-align: center; padding: 5px;">1</td> <td style="text-align: center; padding: 5px;">0</td> </tr> <tr> <td style="text-align: center; padding: 5px;">1</td> <td style="text-align: center; padding: 5px;">1</td> </tr> </table>	1	0	1	1
0	1								
0	0								
1	0								
1	1								

Fig. 9 - Para cada estado existe sempre um outro, obtido da troca dos pixels 1 e 0, que lhe é equivalente e portanto tratado na mesma função *statei()*.

Os casos dos pares de estados (0,15) e (5,10) são triviais pois não correspondem a nenhuma transição. Sendo assim não dão origem a nenhuma acção nem possuem nenhuma função *statei()* associada.

3.4 A Actualização dos Parâmetros Geométricos

Quando se verifica que um dado conjunto consecutivo de pixels pertencente à linha lida do ficheiro de imagem, possui conexão com uma figura já registada no vector de "blobs", então há que actualizar os respectivos parâmetros geométricos. Esses parâmetros são os seguintes:

1. Perímetro
2. Área
3. Somatório dos valores de x
4. Somatório dos valores de y
5. Somatório dos quadrados de x
6. Somatório dos valores de $(x \times y)$
7. Somatório dos quadrados de y

Se a ordenada do conjunto de pixeis for j , a primeira abcissa desse conjunto for k e a última for i , então os valores daqueles somatórios serão:

1. Perim = 2
2. Área = $(i-k)$
3. Som. x = $(i-k) \times (k + (i-k-1)/2)$
4. Som. y = $(i-k) \times j$
5. Som. x^2 = $(i-k) \times (k^2 + (i-k-1) \times (k + (2 \times (i-k) - 1) / 6))$
6. Som. xy = $(i-k) \times (k + (i-k-1)/2) \times j$
7. Som. y^2 = $(i-k) \times j^2$

Estas contribuições são somadas ao conteúdo dos respectivos campos do elemento de entrada associado à figura, no vector de "blobs", sempre que é atingido o fim de um conjunto consecutivo de 1's ou 0's.

Os valores finais serão posteriormente utilizados para calcular parâmetros localizadores da figura assim como identificadores.

A justificação para aquele conjunto de resultados será apresentado já a seguir.

3.4.1 Perímetro

A inclusão de mais uma linha à figura já registada no vector de "blobs" origina o incremento do perímetro em duas unidades referentes aos pixeis extremos. O incremento do valor do perímetro na horizontal é efectuado dentro de algumas das funções *statei()* quando se completam conjuntos de pixeis 0 ou 1 (ver as funções *state2()*, *state6()* e *state1()*).

Como se depreende da descrição anterior, a medida final do perímetro de uma figura nunca vem correcto: as linhas inclinadas apresentam-se em escada formando-se sempre conjuntos sucessivos de triângulos rectângulos; os valores de perímetro são sempre medidos ao longo dos catetos quando eles deveriam na realidade ser medidos ao longo das hipotenusas. A situação mais crítica é aquela para a qual a fronteira se encontra a 45° (Fig.10). O factor de erro é neste caso de $\sqrt{2}$ (~14%).

Deste modo o valor do perímetro virá sempre com um erro apreciável de que resultará a baixa fiabilidade deste parâmetro.

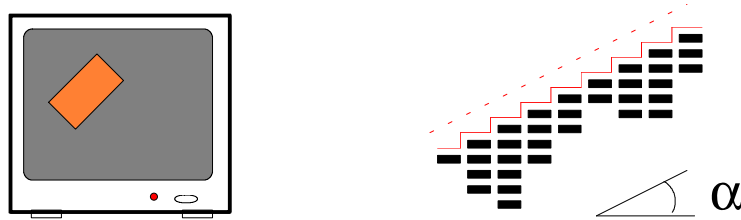


Fig. 10 - Na aproximação dos pixels serem quadrados (que na realidade são retangulares) $\alpha=45^\circ$, o acréscimo de mais uma linha à figura já existente, incrementaria o perímetro em $\sqrt{2}$ (considerando apenas uma das extremidades do conjunto de 1's ou 0's).

3.4.2 Área

Se o conjunto de pixels em linha for em número de $(i-k)$, então será este o incremento para o valor de área da figura. A área total virá então igual ao número total de pixels constituintes da figura.

3.4.3 Somatório dos Valores de j ($\sum j$)

Uma vez que todos os pixels pertencem à mesma linha, eles possuem o mesmo valor de ordenada j e o incremento no somatório dos j s será simplesmente $(i-k) \times j$.

3.4.4 Somatório dos Quadrados de j ($\sum j^2$)

Do mesmo modo o incremento deste parâmetro será de $(i-k) \times j^2$.

3.4.5 Somatório dos Valores de i ($\sum i$)

Considere-se o comprimento do conjunto de 1's ou 0's, $(i-k)$, como sendo L . Uma vez que os pixels a considerar são consecutivos, eles vão possuir valores crescentes de i com incrementos de apenas 1 unidade. Então o somatório vai corresponder à progressão aritmética de razão 1 entre k e $(k+L-1)$:

$$S = \sum_{i=k}^{k+L-1} i = k + (k+1) + (k+2) + \dots + (k+L-1) = L \cdot k + \sum_{i=0}^{L-1} i$$

O resultado daquele 2º somatório é simplesmente a soma de n parcelas todas iguais com o valor da soma entre o 1º e o último elemento (0 e $L-1$). Esse número de parcelas n é

igual a metade do nº de elementos da progressão ou seja, $(L-1-0 + 1)/2=L/2$. S toma então o seguinte valor final:

$$S = L.k + \frac{L.(L-1)}{2} = L.\left[k + \frac{(L-1)}{2}\right] = (i-k).\left[k + \frac{(i-k-1)}{2}\right] = (i-k).(i+k-1)/2$$

3.4.6 Somatório dos Valores de $i \times j$ ($\Sigma i \times j$)

Este somatório é igual ao resultado anterior multiplicado pela ordenada j (uma vez que é comum a todos os pixels considerados):

$$S = j.L.\left[k + \frac{(L-1)}{2}\right] = (i-k).\left[k + \frac{(i-k-1)}{2}\right] \times j$$

3.4.7 Somatório dos Valores de i^2 (Σi^2)

Este é o somatório dos quadrados dos números inteiros consecutivos entre k e i .

Vou contudo considerar inicialmente o somatório entre 0 e n :

$$S(n) = 1^2 + 2^2 + 3^2 + \dots + n^2$$

Este somatório pode ser obtido a partir do desenvolvimento de $(x + 1)^3$:

$$(x + 1)^3 = x^3 + 3x^2 + 3x + 1$$

Se x for substituído sucessivamente por 1, 2, ..., n , obtemos o seguinte conjunto de igualdades:

$$\begin{array}{ll} x = 1 & (1 + 1)^3 = 1^3 + 3(1)^2 + 3(1) + 1 \\ x = 2 & (2 + 1)^3 = 2^3 + 3(2)^2 + 3(2) + 1 \\ x = 3 & (3 + 1)^3 = 3^3 + 3(3)^2 + 3(3) + 1 \\ \dots & \dots \\ x = n & (n + 1)^3 = n^3 + 3n^2 + 3n + 1 \end{array}$$

Somando membro a membro e tendo em conta que,

$$\begin{array}{ll} (1 + 1)^3 & \text{corta com } 2^3 \\ (2 + 1)^3 & \text{corta com } 3^3 \\ \dots & \dots \\ (n-1 + 1)^3 & \text{corta com } n^3 \end{array}$$

Fica,
$$(n+1)^3 = 1 + 3 \sum_{i=1}^n i^2 + 3 \sum_{i=1}^n i + n$$

Ora, $\sum i$ corresponde à progressão aritmética de razão 1 entre 1 e n cujo resultado já foi apresentado atrás:

$$\sum_{i=1}^n i = \frac{1}{2}n(n+1)$$

Pelo que se obtém,
$$(n+1)^3 = 1 + 3 \sum_{i=1}^n i^2 + \frac{3}{2}n(n+1) + n$$

Rearranjando obtém-se finalmente,
$$S(n) = \sum_{i=0}^n i^2 = \frac{1}{6}n(n+1)(2n+1)$$

A prova da validade desta fórmula para qualquer n inteiro pode ser obtida por indução. Supondo então que ela é válida para n , então terá que o ser para $(n+1)$, ou seja,

$$S(n+1) = S(n) + (n+1)^2$$

O 1º membro é igual a,

$$S(n+1) = \frac{1}{6}(n+1)(n+2)(2n+3)$$

O 2º membro terá que dar o mesmo resultado:

$$\begin{aligned} S(n) + (n+1)^2 &= \\ &= \frac{1}{6}n(n+1)(2n+1) + (n+1)^2 = (n+1) \left[\frac{1}{6}n(2n+1) + (n+1) \right] = \frac{1}{6}(n+1)[2n^2 + 7n + 6] \end{aligned}$$

Verifiquemos que $n=-2$ é uma das raízes:

$$\begin{array}{r|l} 2n^2 + 7n + 6 & n + 2 \\ -2n^2 - 4n & 2n + 3 \\ \hline 3n + 6 & \\ -3n - 6 & \\ \hline 0 & \end{array}$$

É-o de facto e então,
$$(2n^2 + 7n + 6) = (n+2)(2n+3)$$

pelo que o resultado final é, $\frac{1}{6}(n+1)(n+2)(2n+3)$, ou seja $S(n+1)$ como se queria provar.

Em função deste resultado pode-se chegar ao valor do somatório de x^2 entre k e i :

$$S = k^2 + (k+1)^2 + \dots + i^2 = S(i) - S(k) + k^2,$$

(deve ser incluído k^2 uma vez que esta parcela pertencente a $S(i)$ é eliminada por $S(k)$).

$$S(i) - S(k) = \frac{1}{6}i(i+1)(2i+1) - \frac{1}{6}k(k+1)(2k+1)$$

Considerando apenas o 1º termo e o comprimento L do somatório:

$$i = k + (L-1)$$

Mas para simplificar o processo de cálculo considerara-se inicialmente $i = k + L$:

$$S(i) - S(k) = \frac{1}{6}(k+L)(k+L+1)(2k+2L+1) - \frac{1}{6}k(k+1)(2k+1)$$

Agora desenvolve-se a primeira parcela e procura-se o simétrico da 2ª para se eliminarem mutuamente:

$$\begin{aligned} \frac{1}{6}(k+L)(k+L+1)(2k+2L+1) &= \frac{1}{6}k(k+1)(2k+1) + \\ &+ \frac{1}{6}kL(2k+2L+1) + \frac{1}{6}L(k+L+1)(2k+2L+1) \end{aligned}$$

A primeira parcela é então eliminada e fica apenas a 2ª linha cujo desenvolvimento dá o seguinte resultado:

$$\frac{L}{6} \left[6k^2 + 2L^2 + 6kL + 6k + 3L + 1 \right] = L \left[(k^2 + kL + k) + \frac{1}{6}(2L^2 + 3L + 1) \right]$$

Na realidade onde está L deverá estar $(L-1)$:

$$\begin{aligned} (L-1) \left[k^2 + kL + \frac{1}{6}L(2L-1) \right] &= Lk^2 - k^2 + kL(L-1) + \frac{1}{6}L(L-1)(2L-1) = \\ &= L \cdot \left[k^2 + k(L-1) + \frac{1}{6}(L-1)(2L-1) \right] - k^2 \end{aligned}$$

Este é o resultado para $S(i) - S(k)$.

Mas S é dado por $S(i) - S(k) + k^2$, pelo que fica:

$$S = L \cdot \left[k^2 + k(L-1) + \frac{1}{6}(L-1)(2L-1) \right]$$

Sabendo que $L = (i-k)$ fica então:

$$S = (i-k) \cdot \left[k^2 + (i-k-1) \left(k + \frac{1}{6} (2(i-k)-1) \right) \right]$$

Após o completo varrimento da imagem binarizada nos quais aqueles parâmetros foram diversas vezes actualizados com as fórmulas apresentadas acima, fica completa a medida dos respectivos valores. Estes podem ser utilizados para avaliação de alguns valores importantes que irão ser utilizados não só para localizar os objectos presentes na imagem e medir a sua orientação, mas também para calcular parâmetros identificadores. De imediato pode-se calcular a posição do centro de massa (centro geométrico da figura):

$$\begin{cases} \langle x \rangle = \text{sum}i / \text{area} \\ \langle y \rangle = \text{sum}j / \text{area} \end{cases}$$

O ângulo de orientação relativamente ao eixo dos xx (no sistema de referência da imagem) requer ainda algumas operações com os momentos de 2ª ordem.

3.5 O Cálculo do Ângulo de Orientação

O ângulo de orientação será dado pela inclinação da recta que atravesse o objecto e minimize as distâncias entre ela e todos os pixeis a ele pertencentes.

Na aproximação (levantada mais adiante) da recta possuir uma inclinação pequena, então a distância de um dado pixel à recta pode ser medida na vertical tal como indicada na Fig.11.

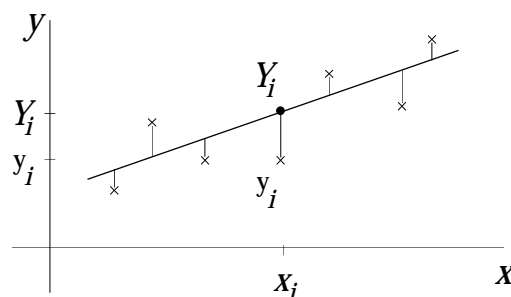


Fig. 11 - Desvio do ponto experimental relativamente à recta ideal, medido na vertical.

Se o pixel tiver coordenadas (x_i, y_i) e a ordenada da recta para $x=x_i$ for Y_i , então a distância será dada por,

$$d_i = | Y_i - y_i | = | m \times x_i + b - y_i |$$

onde m e b são respectivamente o declive e a ordenada na origem daquela recta, a determinar. O somatório de todos os d_i dá uma medida da "distância" D dos pontos à recta que se pretende minimizar:

$$D = \sum_i |d_i| \approx D = \sum_i (d_i)^2$$

A substituição do módulo pelo quadrado deve-se ao facto daquela primeira função não ser derivável. Fica então,

$$D = \sum_i (mx_i + b - y_i)^2$$

Para que a recta esteja orientada com o objecto é necessário que aquela "distância" seja minimizada. Por outras palavras, as suas derivadas parciais em ordem ao declive m e à ordenada na origem b devem ser nulas:

$$\begin{cases} \frac{\partial D}{\partial m} = 0 \\ \frac{\partial D}{\partial b} = 0 \end{cases} \quad \text{ou seja,} \quad \begin{cases} 2 \sum_i (mx_i + b - y_i)x_i = 0 \\ 2 \sum_i (mx_i + b - y_i) = 0 \end{cases} \Leftrightarrow \begin{cases} m \sum_i x_i^2 + b \sum_i x_i - \sum_i x_i y_i = 0 \\ m \sum_i x_i + \sum_i b - \sum_i y_i = 0 \end{cases}$$

$$\text{de onde se tira que,} \quad \begin{cases} m = \frac{N \sum_i x_i y_i - \sum_i x_i \sum_i y_i}{N \sum_i x_i^2 - (\sum_i x_i)^2} \\ b = \frac{\sum_i x_i^2 \sum_i y_i - \sum_i x_i y_i \sum_i x_i}{N \sum_i x_i^2 - (\sum_i x_i)^2} \end{cases} \Leftrightarrow \begin{cases} m = \frac{\text{area} \times \text{sum}ij - \text{sum}i \times \text{sum}j}{\text{area} \times \text{sum}i^2 - (\text{sum}i)^2} \\ b = \frac{\text{sum}i^2 \times \text{sum}j - \text{sum}ij \times \text{sum}i}{\text{area} \times \text{sum}i^2 - (\text{sum}i)^2} \end{cases}$$

Ou seja, de posse de todos aqueles parâmetros geométricos encontra-se de imediato o ângulo de orientação $\alpha = \text{arc tg}(m)$.

Aqueles resultados podem ainda ser postos noutra forma mais simples se tivermos em conta que os valores médios são dados por,

$$\begin{cases} \langle x \rangle = \frac{\sum_i x_i}{N} = \frac{\text{sum } i}{\text{area}} \\ \langle y \rangle = \frac{\sum_i y_i}{N} = \frac{\text{sum } j}{\text{area}} \\ \langle xy \rangle = \frac{\sum_i x_i y_i}{N} = \frac{\text{sum } ij}{\text{area}} \\ \langle x^2 \rangle = \frac{\sum_i x_i^2}{N} = \frac{\text{sum } i^2}{\text{area}} \\ \langle x \rangle^2 = \left(\frac{\sum_i x_i}{N} \right)^2 = \left(\frac{\text{sum } i}{N} \right)^2 \end{cases}$$

$$\begin{cases} m = \frac{\langle xy \rangle - \langle x \rangle \langle y \rangle}{\langle x^2 \rangle - \langle x \rangle^2} \\ b = \frac{\langle x^2 \rangle \langle y \rangle - \langle xy \rangle \langle x \rangle}{\langle x^2 \rangle - \langle x \rangle^2} \end{cases} \quad \text{pelo que, } \alpha = \arctg(m) = \arctg\left(\frac{\langle xy \rangle - \langle x \rangle \langle y \rangle}{\langle x^2 \rangle - \langle x \rangle^2}\right)$$

Consideram-se ainda situações em que o ângulo de inclinação α seja apreciável. Quando tal acontecer os cálculos acima apresentados podem ser repetidos considerando a função $x = f(y)$, ou seja, trocando os eixos. Então α' virá relativamente ao eixo y e será dado por:

$$\alpha' = \arctg(m') = \arctg\left(\frac{\langle xy \rangle - \langle x \rangle \langle y \rangle}{\langle y^2 \rangle - \langle y \rangle^2}\right)$$

$$\text{e, } \alpha = \pi/2 - \alpha'$$

Esta expressão poderá ser utilizada sempre que o denominador da anterior se aproxime de 0.

Ainda, e se considerar o teorema de Steiner que para uma figura de área de superfície S e eixo E em relação ao qual se calculam os momentos de 2ª ordem, distanciado de L de um outro paralelo que passe pelo centro de massa, tenho que,

$$I = I_0 + S.L^2$$

onde I_0 é o momento de inércia calculado relativamente ao eixo que passa pelo centro de massa. Aquela expressão aplicada por exemplo ao eixo dos xx fica:

$$\text{sum } i^2 = \text{sum } i^2_0 + \text{area} \cdot \langle x \rangle^2$$

e se a expressão for normalizada (dividida pela área) fica:

$$X2 = \text{sum } i^2_0 / \text{area} = \text{sum } i^2 / \text{area} - \langle x \rangle^2$$

e do mesmo modo para y :

$$Y2 = \text{sum } j^2_0 / \text{area} = \text{sum } j^2 / \text{area} - \langle y \rangle^2$$

para o parâmetro $\text{sum } ij$ pode obter-se uma relação semelhante:

$$XY = \text{sum } ij_0 / \text{area} = \text{sum } ij / \text{area} - \langle x \rangle \langle y \rangle$$

onde agora $\text{sum } ij_0$ refere-se à intersecção dos dois eixos paralelos aos eixos dos xx e yy que passam (ambos) pelo centro de massa; ou seja, refere-se precisamente ao centro de massa (ver demonstração mais adiante).

São estes valores, normalizados e calculados relativamente ao centro de massa que são registados nos campos respectivos da estrutura dos "blobs".

Com estes novos valores podem-se obter expressões para α ainda mais simples:

$$\alpha = \operatorname{arctg}\left(\frac{XY}{X2}\right) \quad \text{para } \alpha \text{ pequeno, ou}$$

$$\alpha = \frac{\pi}{2} - \operatorname{arctg}\left(\frac{XY}{Y2}\right) \quad \text{para } \alpha \text{ próximo de } \pi/2.$$

Para evitar a verificação de α pequeno pode ainda e finalmente utilizar-se a seguinte expressão:

$$\alpha = \frac{1}{2} \operatorname{arctg}\left(\frac{2 \cdot XY}{X2 - Y2}\right)$$

que deriva da seguinte formulação:

→ para α próximo de $\pi/2$ a expressão a utilizar é: $\operatorname{tg}\alpha' = \left(\frac{XY}{Y2}\right)$

onde $(\alpha' + \alpha) = \pi/2$; sendo assim, $\operatorname{tg}\alpha' = \frac{1}{\operatorname{tg}\alpha} = \left(\frac{XY}{Y2}\right) \Leftrightarrow \operatorname{tg}\alpha = \left(\frac{Y2}{XY}\right)$

→ do mesmo modo para α pequeno podemos escrever: $\operatorname{cotg}\alpha = \frac{1}{\operatorname{tg}\alpha} = \left(\frac{X2}{XY}\right)$

Ora o desenvolvimento de $1 / \operatorname{tg}(2\alpha)$ dá:

$$\frac{1}{\operatorname{tg}(2\alpha)} = \frac{\cos^2\alpha - \sin^2\alpha}{2\sin\alpha \cos\alpha} = \frac{1}{2} \operatorname{cotg}\alpha - \frac{1}{2} \operatorname{tg}\alpha = \frac{1}{2} \left(\frac{X2}{XY}\right) - \frac{1}{2} \left(\frac{Y2}{XY}\right) = \frac{1}{2} \left(\frac{X2 - Y2}{XY}\right)$$

Donde se tira finalmente que:

$$\operatorname{tg}(2\alpha) = \left(\frac{2 \cdot XY}{X2 - Y2}\right) \Leftrightarrow \alpha = \frac{1}{2} \operatorname{arctg}\left(\frac{2 \cdot XY}{X2 - Y2}\right)$$

Esta fórmula única substitui a constante verificação da proximidade de α a $\pi/2$.

3.6 O Momento de Inércia Geométrico Relativamente ao CM

Pode-se provar que a soma dos dois momentos de inércia relativamente aos eixos x_0 e y_0 , paralelos a x e y e a passarem pelo centro de massa (CM) é precisamente o momento de inércia calculado relativamente a CM:

$$\begin{cases} \sum i_o^2 = \sum i^2 - \text{area} \cdot \langle x \rangle^2 \\ \sum j_o^2 = \sum j^2 - \text{area} \cdot \langle y \rangle^2 \end{cases}$$

Somando membro a membro obtém-se,

$$\begin{aligned} I_{cm} &= (\sum i^2 + \sum j^2) - \text{area}(\langle x \rangle^2 + \langle y \rangle^2) = \\ &= (\sum x_i^2 + \sum y_i^2) - \text{area}(\langle x \rangle^2 + \langle y \rangle^2) = \\ &= \sum (x_i^2 + y_i^2) - \text{area}(R^2) = \\ &= \sum (r_i^2) - \text{area}(R^2) \end{aligned}$$

em que r_i é a distância de cada pixel ao centro dos eixos coordenados e R é a distância do CM àquele centro. Esta relação confirma o teorema de Steiner pelo que aquela 1ª relação se prova verdadeira. E sendo assim,

$$I_{cm} = \text{area} \times (X^2 + Y^2)$$

Os momentos de 2ª ordem vêm multiplicados pela área uma vez que se encontravam já normalizados.

Este valor I_{cm} é um parâmetro altamente dependente da forma da figura, caracterizando-a, pelo que será utilizado para identificação de objectos a 2D.

Alternativamente pode-se ainda considerar o momento de inércia geométrico relativamente ao centro de massa e normalizado I_{cm_a} , dado por:

$$I_{cm_a} = (X^2 + Y^2)$$

a importância deste valor reside na possibilidade de se identificar, para além do objecto em particular, o seu tipo. Aliás é este parâmetro que é registado num campo da estrutura do vector de "blobs" embora se utilize o anterior para identificação.

Para demonstrar esta potencialidade apresentam-se dois exemplos característicos: o círculo e o quadrado.

1. Círculo:

$$I_{cm_a} = \frac{\int_0^R (2\pi r)r^2 dr}{\int_0^R (2\pi r) dr} = \frac{2\pi \frac{R^4}{4}}{2\pi \frac{R^2}{2}} = \frac{R^2}{2}$$

2. Quadrado:

$$I_x = y \int_{-x/2}^{x/2} x^2 dx = y \left[\frac{x^3}{3} \right]_{-x/2}^{x/2} = \frac{2}{3} y \frac{x^2}{8} = \frac{x^2 y}{12} = \frac{a^3}{12}$$

Uma vez que a figura possui simetria de rotação a 90° , tenho que:

$$I_y = \frac{y^2 x}{12} = \frac{a^3}{12}$$

Ora, $I_{cm} = I_x + I_y = \frac{a^3}{6}$, pelo que, $I_{cm_a} = \frac{a^3}{6a^2} = \frac{a}{6}$

3.7 O Parâmetro Característico p_sq_a

A juntar a I_{cm_a} utiliza-se ainda um outro parâmetro, p_sq_a , que também permite caracterizar o tipo de objecto. Ele consiste simplesmente na razão entre o perímetro e a raiz quadrada da área, mas adquire valores característicos para determinadas formas:

1. Círculo:

A área é dada por πr^2 e o perímetro por $2\pi r$, donde que p_sq_a dará:

$$p_sq_a = \frac{2\pi r}{\sqrt{\pi r^2}} = 2\sqrt{\pi} \approx 3,545$$

Este valor é o mínimo possível qualquer que seja a forma: ele traduz o máximo de concentração de pixeis de que resulta a minimização do perímetro para uma dada área.

2. Quadrado:

A área é dada por a^2 e o perímetro por $4a$:

$$p_sq_a = \frac{4a}{\sqrt{a^2}} = 4$$

O resultado é, como seria de esperar, superior a 3,545. O valor 4 obtido de um cálculo não implica forçosamente que se trate de um quadrado podendo tratar-se de uma elipse ou de qualquer outra forma próxima do círculo. Este resultado serve só para confirmar que aquele valor 3,545 é de facto o mínimo possível de obter.

3.8 A Selecção dos "Blobs" Válidos

Por muitos elementos do vector de "blobs" que possam ter sido utilizados, apenas alguns sobrevivem ao processo de fusões sucessivas. A detecção destes "blobs" válidos poderia ser feita com o vector classificador já referido anteriormente, com o qual se procurariam os elementos de entrada para os quais o valor registado fosse positivo (se fosse negativo o valor absoluto indicaria o nº do "blob" ao qual o considerado teria sido fundido). Contudo, sabe-se que o elemento nº1 corresponde ao fundo da imagem, sendo os seus "filhos" precisamente os objectos presentes na imagem. Prevendo que haja "ruído" na imagem e que portanto possam aparecer falsos "blobs", seleccionam-se apenas aqueles que possuam uma área acima dum certo valor; por exemplo 20 pixeis.

3.9 Apresentação de Alguns Resultados

Da execução de uma função apropriada e após a análise de uma figura com 4 buracos (Figs. 7 e 12), resulta o registo mostrado na Fig.13.

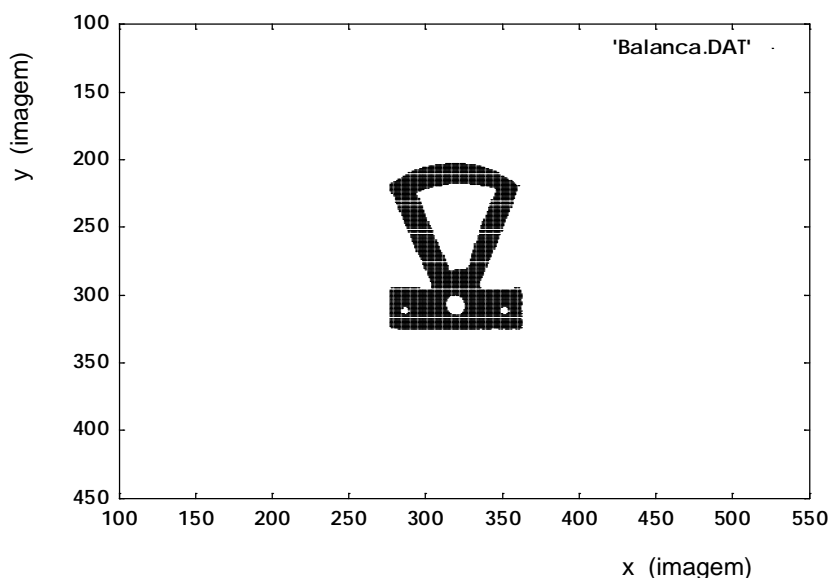


Fig. 12 - Objecto sujeito à análise geométrica.

O "blob" nº 1 corresponde ao fundo da imagem tendo uma área associada de 298.157 pixeis (de um máximo de $640 \times 512 = 327.680$ pixeis). O "filho" único daquele "blob" é o nº 2 (*child=2*) com uma área de 5.663 pixeis e um perímetro de 566 pixeis e que constitui o único objecto na imagem. Os seus 4 filhos constituem o 4 buracos que o

objecto possui e que são acedidos no vector de "blobs" através do campo ".child = 5", sendo este o 1º filho da lista e obtendo-se os seguintes através do campo ".sibling" (de que resultam os "blobs" 8, 6 e 3).

number = 1	color = 0	child = 2	nholes = 1
level = 0	parent = -1	sibling = -1	
xmin = 0	xmax = 639	ymin = 0	ymax = 479
area = 298157	perim = 958		
<X> = 317.9	<Y> = 240.4		
$\langle(X-\langle X \rangle)^2\rangle = 34858.91$ $\langle(Y-\langle Y \rangle)^2\rangle = 19605.31$ $\langle XY \rangle - \langle X \rangle \langle Y \rangle = 13.68$			
number = 2	color = 1	child = 5	nholes = 4
level = 1	parent = 1	sibling = -1	
xmin = 304	xmax = 413	ymin = 166	ymax = 282
area = 5663	perim = 566		
<X> = 355.5	<Y> = 224.0		
$\langle(X-\langle X \rangle)^2\rangle = 1086.50$ $\langle(Y-\langle Y \rangle)^2\rangle = 801.77$ $\langle XY \rangle - \langle X \rangle \langle Y \rangle = 92.73$			
number = 5	color = 0	child = -1	nholes = 0
level = 2	parent = 2	sibling = 8	
xmin = 347	xmax = 400	ymin = 204	ymax = 265
area = 2054	perim = 236		
<X> = 378.2	<Y> = 230.9		
$\langle(X-\langle X \rangle)^2\rangle = 190.88$ $\langle(Y-\langle Y \rangle)^2\rangle = 178.71$ $\langle XY \rangle - \langle X \rangle \langle Y \rangle = 19.41$			
number = 8	color = 0	child = -1	nholes = 0
level = 2	parent = 2	sibling = 6	
xmin = 314	xmax = 319	ymin = 252	ymax = 257
area = 27	perim = 24		
<X> = 316.3	<Y> = 254.6		
$\langle(X-\langle X \rangle)^2\rangle = 2.06$ $\langle(Y-\langle Y \rangle)^2\rangle = 2.25$ $\langle XY \rangle - \langle X \rangle \langle Y \rangle = -0.13$			
number = 6	color = 0	child = -1	nholes = 0
level = 2	parent = 2	sibling = 3	
xmin = 320	xmax = 332	ymin = 211	ymax = 225
area = 154	perim = 56		
<X> = 325.8	<Y> = 217.7		
$\langle(X-\langle X \rangle)^2\rangle = 10.24$ $\langle(Y-\langle Y \rangle)^2\rangle = 14.71$ $\langle XY \rangle - \langle X \rangle \langle Y \rangle = -0.28$			
number = 3	color = 0	child = -1	nholes = 0
level = 2	parent = 2	sibling = -1	
xmin = 327	xmax = 331	ymin = 175	ymax = 181
area = 26	perim = 24		
<X> = 329.1	<Y> = 178.4		
$\langle(X-\langle X \rangle)^2\rangle = 1.53$ $\langle(Y-\langle Y \rangle)^2\rangle = 2.78$ $\langle XY \rangle - \langle X \rangle \langle Y \rangle = -0.11$			

Fig. 13 - 1.ºs resultados da análise geométrica registados em disco.

Para além dos centros de massa, obtêm-se também já os valores dos momentos de 2ª ordem, normalizados e calculados relativamente aos respectivos centros de massa. Em função deles o programa calcula quer o ângulo de orientação, α , quer o momento de inércia geométrico relativo ao seu centro, I_{cm} . Este último, aliado aos parâmetros $area$ e p_sq_a vai constituir um parâmetro caracterizador da forma da figura, potencialmente utilizável portanto na sua identificação.

Um outro registo é ainda efectuado contendo toda esta nova informação (Fig.14):

<pre> There are 1 objects ! --- Object nr. 1 --- Features: Peri = 566 Area = 5663 Peri/sqr(Area) = 7.52130 Icm/area = 0.33344 Position and orientation: <X> = 355.5, <Y> = 224.0 Angle = 16.539° Nr of holes: 4 Hole nr. 1 Peri = 236 Area = 2054 Peri/sqr(Area) = 7.52130 <X> = 378.2, <Y> = 230.9 </pre>	<pre> Icm/area = 0.06526 Hole nr. 2 Peri = 24 Area = 27 Peri/sqr(Area) = 7.52130 <X> = 316.3, <Y> = 254.6 Icm/area = 0.00076 Hole nr. 3 Peri = 56 Area = 154 Peri/sqr(Area) = 7.52130 <X> = 325.8, <Y> = 217.7 Icm/area = 0.00441 Hole nr. 4 Peri = 24 Area = 26 Peri/sqr(Area) = 7.52130 <X> = 329.1, <Y> = 178.4 Icm/area = 0.00076 </pre>
---	---

Fig. 14 - Registo de informação seleccionada onde se evidenciam as relações de parentesco entre "blobs".

Outra função auxiliar desenha no monitor e sobreposto às figuras, os respectivos centros de massa assim como linhas que definem a sua orientação. Isto serve essencialmente para confirmar a boa execução do algoritmo de detecção e medida das figuras contidas na imagem, assim como a correcta eliminação de falsos objectos. A Fig.15 mostra a execução desta função para um conjunto de objectos. Nela pode constatar-se que o algoritmo está preparado para considerar e desenhar linhas verticais.

A marcação dos centros de massa é imediata: basta consultar os campos ".x" e ".y" dos "blobs" válidos e marcar alguns pixeis em torno de (x,y) com a cor inversa à de cada um deles. O traçado do segmento de recta que define a orientação da figura utiliza para além daqueles campos, o campo ".angle". Em função deles assim como das coordenadas que delimitam a zona de interesse (neste caso (0,0) e (639,511) são os pontos que a definem) são encontrados os seus dois pontos extremos quaisquer que sejam os eixos coordenados que sejam intersectados por ele.

De seguida é definido um valor de separação de cada ponto que o compõe, em pixeis, e calculado o incremento a dar a x para o obter na direcção pretendida (deste modo todas as linhas possuirão o mesmo espaçamento entre pontos); todos os pontos

são então marcados invertendo a cor que cada um possuía anteriormente desde o ponto inicial até ao ponto final.

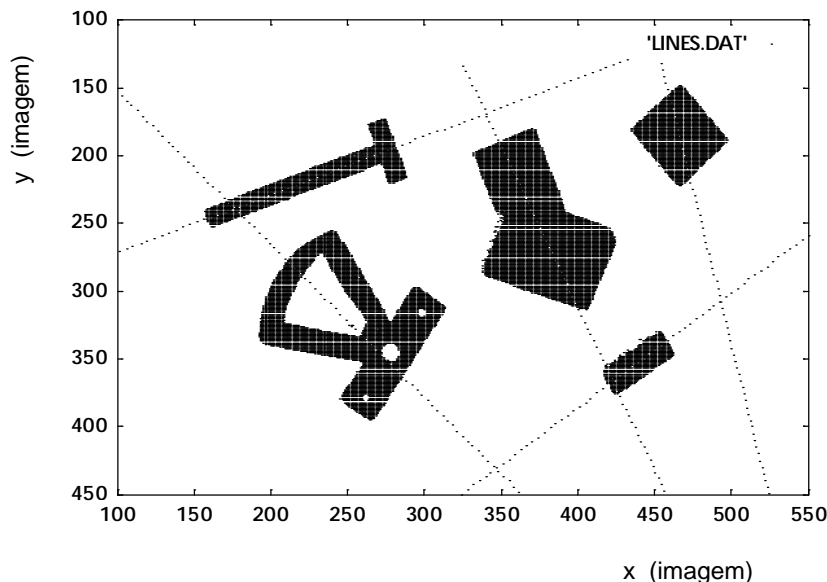


Fig. 15 - Desenho no monitor dos centros de massa e de linhas que definem a orientação dos vários objectos.

Se a linha estiver próxima da vertical, então as coordenadas x e y são trocadas para evitar valores de declive muito elevados e os cálculos referentes àqueles valores para cada pixel do segmento são calculados usando precisamente o mesmo código. Apenas quando da marcação em monitor, as coordenadas x e y são de novo trocadas.

Como se poderá constatar por observação da Fig.15, para objectos não alongados a linha de orientação obtida poderá não ser a mais correcta (objecto na parte superior direita). Esse erro pode ser posteriormente corrigido após comparação com a direcção do maior segmento da fronteira, caso ele exista.

No Anexo C.1 é feita uma descrição das funções que compõem o programa de análise de características geométricas a 2D.

No Anexo C.2 é apresentada uma demonstração do Teorema de Steiner aplicado a massas pontuais discretas, cujo resultado é utilizado no cálculo de um dos parâmetros característicos: o momento de inércia geométrico relativo ao centro geométrico, I_{cm_a} .

No Anexo C.3 é apresentada a demonstração do resultado,

$$XY = \langle xy \rangle = \frac{\sum ij_o}{area} = \frac{\sum ij}{area} - \langle x \rangle \langle y \rangle$$

que corresponde à soma normalizada dos produtos $i \times j$ (com estas coordenadas referidas ao centro geométrico) referentes a todos os pixels pertencentes a uma figura na imagem, que é utilizada no cálculo do ângulo de orientação α da referida figura.

3.10 Conclusão

Foi apresentado aqui um processo de análise das características geométricas de objectos presentes numa imagem adquirida pela câmara.

Algumas dessas características, de acordo com a formulação teórica apresentada, podem ser utilizados para determinar a sua localização e orientação, dados essenciais para a manipulação robótica. Os resultados apresentados, nomeadamente a representação gráfica dos centros geométricos e das linhas de orientação, permitiram concluir que essa determinação é eficiente.

Para além disso, verificou-se que essas características podem definir inequivocamente os objectos, pelo que são adequadas para determinar posteriormente a respectiva identidade, processo que será desenvolvido no capítulo seguinte.

4.1 Introdução

Uma vez na posse das características geométricas de todos os objectos considerados válidos (para além da sua posição e orientação), resta agora proceder à sua identificação. Esta é obtida por comparação com todos os modelos contidos numa base de dados. A construção desta base de dados é prévia à execução deste algoritmo e consiste no registo de todos os parâmetros geométricos de todos os objectos de interesse, e em todos os seus estados estáveis. Entende-se por estado estável aquele em que o objecto pode ser encontrado em equilíbrio estável e sem contacto físico com outros. O resultado de uma identificação consiste na atribuição de um nome, na indicação do estado estável em que o objecto se encontra e de um coeficiente de comparação ou "matching" para os dois melhores modelos encontrados. A escolha dos dois modelos é necessária pois podem existir objectos com a mesma vista de topo mas com alturas diferentes, de que resultariam coeficientes de "matching" muito próximos. O levantamento desta ambiguidade seria depois efectuada após o pedido da medida da 3ª dimensão ao Agente Laser.

4.2 A Estrutura do Ficheiro de Modelos

O registo da informação contida nas estruturas para o ficheiro de modelos, é feito sequencialmente, em modo binário e apenas para a informação existente, ou seja, os registos dos vários objectos possuem comprimentos diferentes, por forma a poupar tanto o espaço em disco como em memória RAM. A poupança desta deve-se ao facto do ficheiro de modelos ser integralmente lido para memória para maior rapidez de acesso no processo de consulta dos modelos (Fig.1).

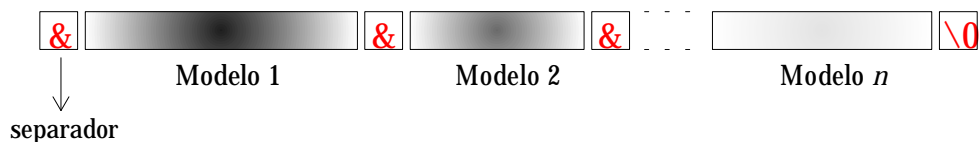


Fig. 1 - Registo sequencial e em modo binário da informação dos modelos. Os registos possuem comprimento variável consoante a informação associada a cada modelo.

A inclusão de um separador ('&') deveu-se à necessidade de confirmar o correcto posicionamento dos apontadores que permitiam a leitura dos vários parâmetros geométricos, quando da depuração do programa de leitura do ficheiro de modelos. Essa verificação manteve-se sempre mesmo durante a execução do programa.

A estrutura de cada registo é a seguinte (Fig.2): logo de início apresenta-se o registo da identificação do modelo através da estrutura referente à identidade que contém o seu nome e o número de estados estáveis. Seguem-se n estruturas referentes aos parâmetros geométricos de cada estado estável, sendo n o seu número total. Esta estrutura contém, como pode ser visto na figura, a altura máxima do modelo, a informação geométrica completa e finalmente o número de buracos associados.

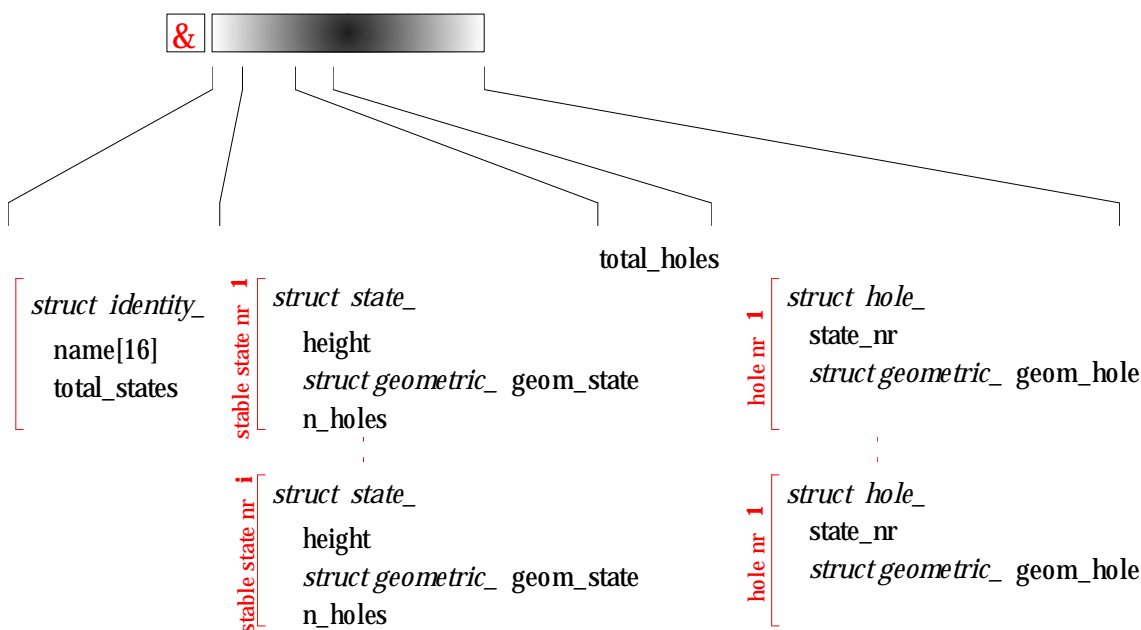


Fig. 2 - Estrutura de cada registo no ficheiro de modelos.

Segue-se o registo do número total l de buracos para todos os estados estáveis e as l estruturas que lhes correspondem. Cada uma destas estruturas contém o número do estado estável a que pertencem e a estrutura de parâmetros geométricos.

Sempre que um registo é acedido, toda a informação dos estados estáveis é lida para um vector e a dos buracos para outro. Estes valores servirão para efectuar a comparação com os parâmetros medidos de cada um dos objectos da cena.

Durante o processo de comparação é mantida uma estrutura que contém os dois melhores modelos até então encontrados. A informação nela contida para cada um deles é o nome, o estado estável detectado, o número do respectivo modelo no ficheiro, o coeficiente de comparação ou "matching" e a maior altura associada.

4.3 A Consulta e Gestão do Ficheiro de Modelos

A gestão do ficheiro de modelos é efectuada com comandos do tipo "r-model *xx yy*". O primeiro argumento (*xx*) é um conjunto de 2 caracteres que define o tipo de operação a ser efectuada, enquanto que o segundo (*yy*) se refere ao número do modelo a ser considerado caso seja necessário. É a seguinte a lista de operações efectuadas por este programa:

lm	listagem do nome de todos os modelos com o respectivo nº de estados estáveis
gm <i>i</i>	visualização da informação relativa ao modelo nº <i>i</i>
em <i>i</i>	edição da informação relativa ao modelo nº <i>i</i>
rm <i>i</i>	remoção do modelo nº <i>i</i>

4.3.1 A Consulta Genérica da Base de Dados ("lm")

Nesta opção o programa procura sucessivamente os vários caracteres '&' (não sequencialmente, mas antes "saltando" as várias estruturas que se encontram registadas consoante o seu comprimento) e recolhe os nomes e os números dos estados estáveis associados, após o que faz o "display" de toda esta informação:

r-model lm

listing all model names and s.s. ...

model nr - 1- :	açucareiro	nr of ss = 1
model nr - 2- :	caixa	nr of ss = 1
model nr - 3- :	pimenteiro	nr of ss = 1
model nr - 4- :	block_1	nr of ss = 3
model nr - 5- :	block_2	nr of ss = 3
model nr - 6- :	block_3	nr of ss = 3
model nr - 7- :	sal	nr of ss = 2
model nr - 8- :	pimenta	nr of ss = 2

4.3.2 A Consulta do Conteúdo de um dado Modelo ("gm i")

Nesta opção o programa procura o separador '&' a partir do qual se encontra registado o modelo nº *i* (do modo já indicado atrás), lê para variáveis previamente definidas e com as estruturas adequadas a informação a ele associada e por fim faz o seu "display", tal como apresento a seguir:

r-model gm 4

```
reading model nr -4- ...
displaying model info...
```

```
name -- block_1
nr. of states -- 3
total nr of holes -- 0
```

```
state nr. 1:
  height = 26
  perim = 407
  area = 5830
  perim/sqrt(area) = 5.327
  Icm/area = 1282.6
  nr. of holes -- 0
```

```
state nr. 2:
  height = 56
  perim = 274
  area = 2839
  perim/sqrt(area) = 5.142
  Icm/area = 1002.2
  nr. of holes -- 0
```

```
state nr. 3:
  height = 122
  perim = 170
  area = 1458
  perim/sqrt(area) = 4.452
  Icm/area = 288.7
  nr. of holes -- 0
```

Se o modelo contiver buracos a informação aparece com a seguinte disposição:

```
reading model nr -14- ...
displaying model info...
```

```
name -- exp-4
nr. of states -- 2
total nr of holes -- 3
```

```
state nr. 1:
  height = 1
  perim = 2
  area = 3
  perim/sqrt(area) = 4.000
  Icm/area = 5.000
  nr. of holes -- 1
  (state nr -- 1)
    perim = 1
    area = 2
    perim/sqrt(area) = 3.000
```

Icm/area = 4.000

```
state nr. 2:
  height = 1
  perim = 2
  area = 3
  perim/sqrt(area) = 4.000
  Icm/area = 5.000
  nr. of holes -- 2
  (state nr -- 2)
    perim = 1
    area = 2
    perim/sqrt(area) = 3.000
    Icm/area = 4.000
  (state nr -- 2)
    perim = 5
    area = 6
    perim/sqrt(area) = 7.000
    Icm/area = 8.000
```

Este último modelo apresenta já a informação geométrica relativa aos buracos pertencentes aos dois estados estáveis. Este objecto foi acrescentado em modo manual ao ficheiro de modelos como experiência: ele não tem existência real e os seus valores não têm também qualquer significado.

4.3.3 A Eliminação de Modelos ("rm i")

Se se eliminar o modelo nº 2, 4 vezes consecutivas e se consultar de novo a base de dados, obtém-se o seguinte:

4× r-model *rm 2*

```
removing model nr -2- ...
Removal of model -2- successful !
```

r-model lm

```
listing all model names and s.s. ...

model nr - 1- :   açucareiro      nr of ss = 1
model nr - 2- :   block_3        nr of ss = 3
model nr - 3- :     sal           nr of ss = 2
model nr - 4- :   pimenta        nr of ss = 2
```

Para cada eliminação é efectuado um deslocamento em memória da informação à direita do modelo eliminado para o início da referente a este. O registo em RAM é então regravado em disco tendo em conta o seu novo comprimento.

4.3.4 A Edição de um Modelo ("em i")

Consideremos o novo modelo nº 3, e suponhamos que por erro de programação os momentos de inércia geométricos (relativamente aos centros de massa) normalizados, haviam sido divididos duas vezes pela área. Sendo assim, há necessidade de editar o modelo para efectuar as correcções, o que pode ser feito com o comando de linha "r-model em 3":

r-model em 3

```
editing model nr -3- ...

name -- sal;      new name -- sal

----- State nr -1- -----

height -- 80;      new value = 80
perim  -- 158;     new value = 158
area   -- 989;     new value = 989
p_sq_a -- 5.024;   new value = 5.024
lcm_a  -- 0.166;   new value = 164,2
nr of holes = 2 !
```

```

--- Hole nr -1- --- (belongs to state nr 1)

perim = 12;          new value = 12
area = 5;           new value = 5
p_sq_a = 5.024;    new value = 5.024
Icm_a = 0.001;     new value = 0.005

--- Hole nr -2- --- (belongs to state nr 1)

perim = 14;          new value = 14
area = 8;           new value = 8
p_sq_a = 5.024;    new value = 5.024
Icm_a = 0.001;     new value = 0.008

----- State nr -2- -----

height -- 42;       new value = 42
perim -- 256;       new value = 256
area -- 2508;       new value = 2508
p_sq_a -- 5.112;    new value = 5.112
Icm_a -- 0.215;     new value = 539.2
nr of holes = 0 !

Editing of model nr -3- successful !

```

Após a edição dos valores, o ficheiro de modelos é regravado em disco, passando o modelo nº 3 a exibir os seus novos valores de *Icm_a*.

4.4 A Construção do Ficheiro de Modelos

Para o acréscimo de um modelo ao ficheiro já existente é solicitada ao utilizador a entrada do nome, número de estados estáveis e para cada um destes, os valores das várias características geométricas. São também pedidos os valores referentes aos vários buracos, se existirem. Todos estes valores são então acrescentados ao ficheiro de modelos já existente. Se ele não existir, é criado com estes mesmos valores.

Alternativamente pode-se construir um novo modelo automaticamente: nesta opção, é perguntado ao utilizador o número de estados estáveis e para cada um, este é solicitado a colocar o objecto nas coordenadas do robot $(X;Y)=(100;0)$. Logo de seguida é executado o programa de análise de figuras e medição das características geométricas. Uma vez analisados os n estados estáveis, a informação recolhida é mostrada em ecrã para verificação por parte do utilizador e caso seja aprovada ela é acrescentada ao ficheiro de modelos existente. Mais uma vez, no caso de este não existir, é criado com toda aquela informação.

4.5 A Fiabilidade dos Parâmetros Identificadores

O sucesso do processo de identificação depende obviamente da fiabilidade dos vários parâmetros identificadores utilizados. Sendo assim, houve que medir experimentalmente a tolerância associada a cada um deles: foi escolhido um objecto poliédrico e colocado em vários pontos da cena em diferentes orientações; para cada posição e após a execução do programa de medida das características geométricas, estas foram registadas. De posse de todos os valores correspondentes aos quatro parâmetros foram então calculados os respectivos desvios padrão:

$$\sigma_x = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n - 1}}$$

onde x é um dos 4 parâmetros.

Os resultados foram os seguintes:

Tolerância dos parâmetros identificadores			
Perim	Area	P_sq_a	Icm
13,5 %	3,0 %	13,5 %	8,0 %

Como já havia sido previsto quando da descrição do processo de medida do perímetro, este apresenta uma variação superior aos restantes parâmetros, não considerando p_sq_a que é dependente dele. Na Fig.3 mostra-se o efeito de rotação na medida deste parâmetro supondo que o pixel tem uma forma quadrada.

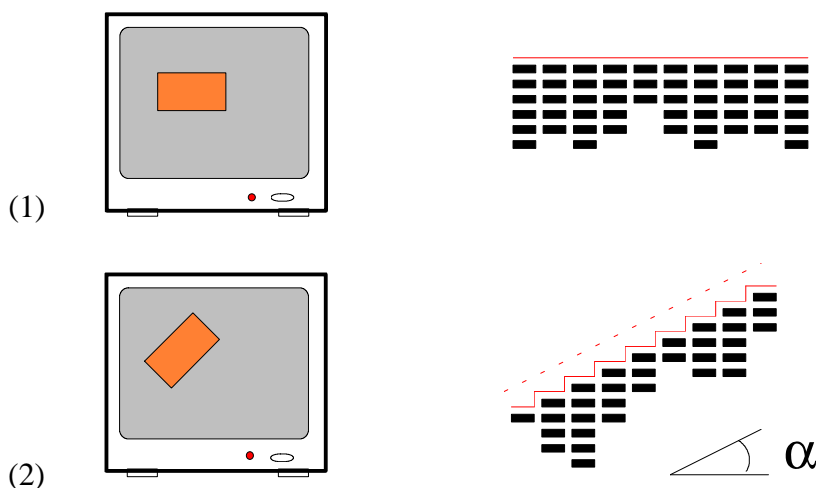


Fig. 3 - Efeito da rotação do objecto na medida do perímetro.
 (1) rectângulo na horizontal.
 (2) rectângulo fazendo um ângulo de α com a horizontal.

Quando a linha de fronteira se encontra inclinada como em (2), cada pixel a ela pertencente contribui com os seus dois catetos para a medida do perímetro quando deveria contribuir apenas com a hipotenusa. Como durante a execução do algoritmo de medida tal não é possível já que as contribuições horizontais e verticais são consideradas separadamente, o valor do perímetro vem sempre incorrecto sendo o desvio tanto maior quanto mais próximo for o ângulo de inclinação de 45° . Ele vai assim variar entre um valor mínimo (quando os segmentos de fronteira se encontram alinhados com os eixos coordenados) e um valor máximo (quando os mesmos formam um ângulo de 45° com a horizontal como já foi dito). Esses valores limites (considerando o pixel quadrado) são então os seguintes:

se $perim$ for o valor mínimo, ou seja quando os segmentos se encontram alinhados com os eixos coordenados, então o valor obtido para uma rotação de α , $perim'$ será dado por,

$$perim' = perim \times f(\alpha)$$

$$\text{em que } f(\alpha) = (\cos \alpha + \sin \alpha) = \sqrt{2} \cdot \sin(\alpha + \pi/4)$$

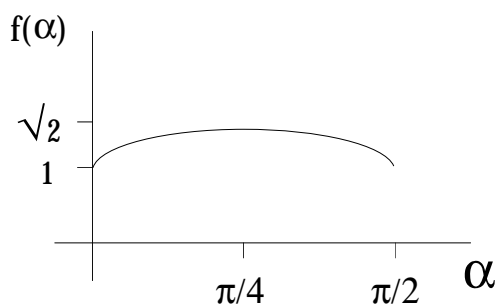


Fig. 4 - Variação do factor que afecta o perímetro em função do ângulo de rotação α .

no caso particular de $\alpha=45^\circ$ temos que, $f(\alpha) = \sqrt{2} \approx 1,41 \rightarrow 14,1\%$

este valor de tolerância confirma o obtido experimentalmente.

Sendo assim, ele deverá ter menos peso quando do cálculo do coeficiente de comparação ou "matching".

A área, por outro lado, possui uma variação muito pequena pelo que constitui um excelente parâmetro para identificação, devendo este possuir maior peso no cálculo daquele coeficiente.

A justificação para aquelas diferenças em fiabilidade não se devem apenas ao processo de cálculo dos parâmetros (que no caso do perímetro considera todos os comprimentos da linha de fronteira). Elas devem-se também à própria forma do pixel que contrariamente ao desejado não é quadrado mas sim rectangular.

Vejamus então primeiramente qual o efeito desta forma no cálculo do perímetro de um rectângulo em duas orientações perpendiculares (Fig.5).

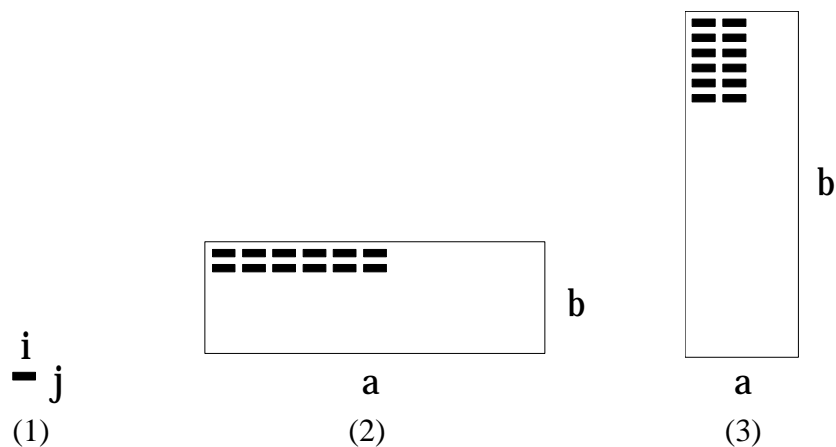


Fig. 5 - (1) Formato e medida do pixel na imagem
 (2) Medida do perímetro com o rectângulo na horizontal.
 (3) Medida do perímetro com o rectângulo na vertical.

Na situação (2) o perímetro, tendo em conta as medidas i e j do pixel, é igual a:

$$perim_1 = n^\circ \text{ de pixels de fronteira} = 2.(a/i) + 2.(b/j)$$

Enquanto que na (3) ele é dado por:

$$perim_2 = n^\circ \text{ de pixels de fronteira} = 2.(a/j) + 2.(b/i) \neq perim_1$$

que é um resultado diferente do anterior (mais exactamente inferior).

Para os mesmos dois casos da Fig.5, vejamos agora qual o efeito da forma do pixel no cálculo da área:

para o 1º caso temos: $area_1 = (a/j) \times (b/i) = (a.b)/(i.j)$

enquanto que no 2º caso temos: $area_2 = (b/j) \times (a/i) = (a.b)/(i.j) = area_1$

Ou seja, a forma do pixel não influi na medida da área do rectângulo! Daí a pequena dispersão apresentada pelos seus valores (3,0%).

4.6 A Estrutura das Funções de Identificação e a sua Execução

O programa de identificação é composto por várias funções que se encontram ilustradas na Fig.6. Cada uma executa uma tarefa simples que o seu próprio nome traduz. Na mesma figura são apresentados as várias ligações de chamada entre funções assim como as variáveis passadas.

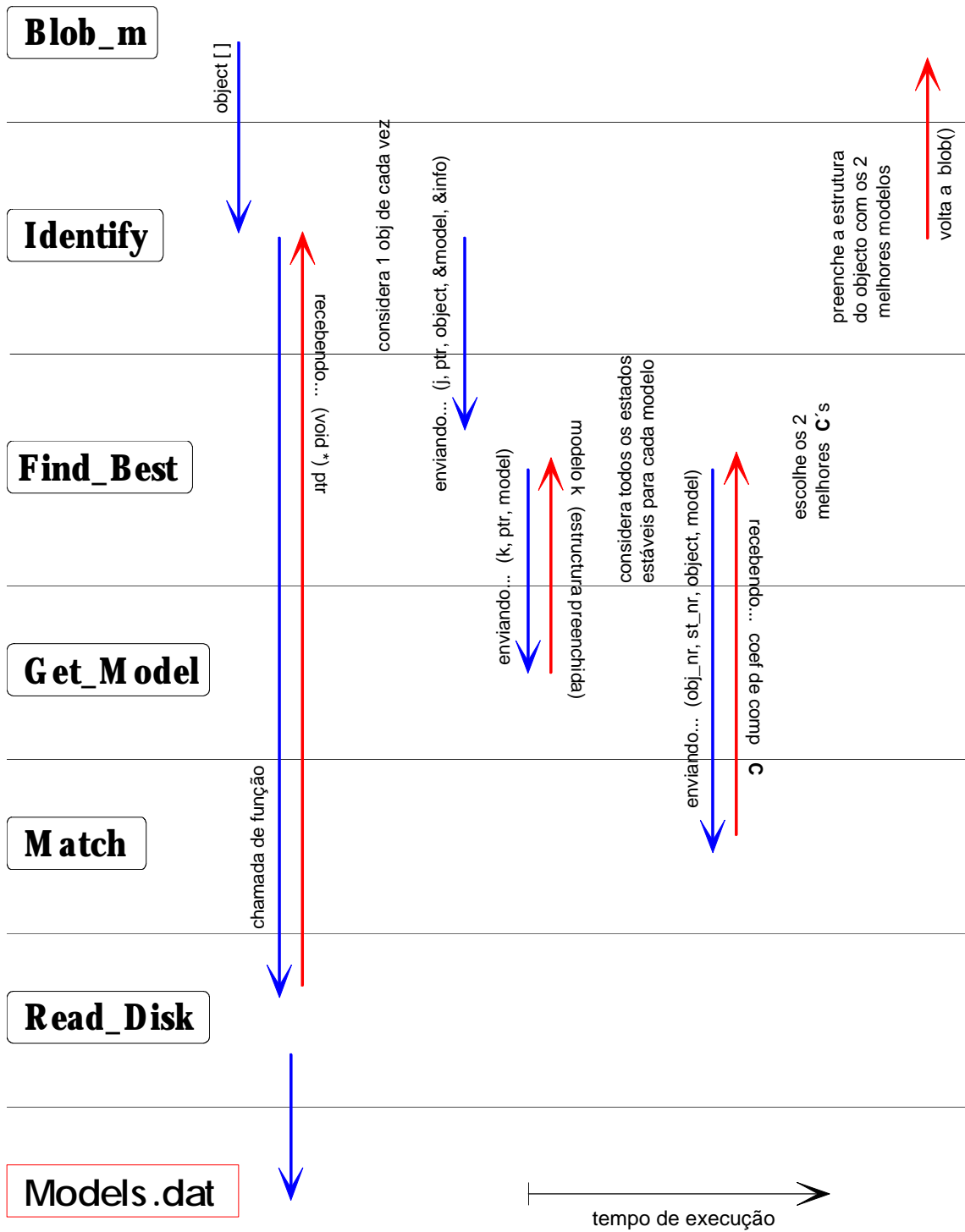


Fig. 6 - Algoritmo de identificação.

De início dispõe-se das características geométricas do objecto na variável *object[]* que é cedida pela função de chamada "Blob_m.c". Estas características irão ser comparadas com as registadas no ficheiro de modelos.

A primeira acção do programa é então a chamada à função "*Read_disk()*" que efectua a leitura integral do ficheiro de modelos "*models.dat*" cuja constituição foi já descrita acima. Para tal é reservada memória RAM suficiente para albergar todo este ficheiro. A leitura para RAM da totalidade dos modelos vem assim substituir o registo em disco. Esta opção deve-se ao menor tempo de acesso aos vários modelos (se estes estiverem em RAM) comparativamente à leitura individual dos mesmos a partir do disco. A estrutura do ficheiro, contudo, mantém-se tal como descrito atrás pelo que o modo de acesso aos vários valores que irão preencher os campos da variável *model* não sofre qualquer alteração.

Para cada elemento do vector *object[]* é então ordenada a procura dos dois melhores modelos que são detectados na função "*Find-best()*" a partir dos vários coeficientes de comparação calculados na função "*Match()*" e retornados àquela. Para esta verificação são considerados todos os modelos existentes em RAM, que são lidos um a um na função "*Get_model()*". Nela, as características registadas de todos os estados estáveis são lidas para a variável *model*. Esta estrutura contém o nome do modelo, o número de estados estáveis, toda a informação referente aos estados estáveis, o total de buracos e finalmente a informação a eles associada. Cada elemento *state[]* contém, para além de toda a informação geométrica, a altura e o número de buracos que possui. Cada buraco contém na sua estrutura, para além das características geométricas, a referência ao estado estável a que pertence.

Neste algoritmo não é efectuada qualquer procura indexada. Esta só seria válida se pudéssemos garantir que uma dada característica era sempre mais fiável que as outras, pelo que a procura se reduziria a um pequeno grupo com o valor daquela característica próximo do valor medido para o objecto. Mas tal não é possível: a ocorrência de "ruído" na imagem pode eventualmente afectar essa característica, mais do que o usual, pelo que o modelo a que o objecto corresponderia iria apresentar um valor fora da gama utilizada para essa procura indexada. Sendo assim, optou-se pela comparação com todos os modelos existentes em RAM. O único critério de rejeição imediata utilizada foi o número de buracos que o modelo contém, com uma área acima de um certo valor mínimo (este valor mínimo deve-se à possibilidade de buracos pequenos serem eliminados durante o processo de binarização pelo que não podem ser utilizados naquele critério).

De referir ainda que cada modelo pode possuir mais do que um estado estável (situação de repouso no plano XOY sem qualquer contacto físico com outro objecto) pelo que todos eles são considerados.

O resultado (os dois melhores modelos) é registado numa variável própria *info*. Nela, encontram-se registados os nomes dos dois melhores modelos encontrados, os respectivos números dos estados estáveis, os números de ordem em que se encontravam registados no ficheiro, os coeficientes de comparação encontrados e ainda a altura máxima dos objectos correspondentes para posterior verificação. Aliás, é nesta variável que vão sendo registados os melhores modelos até então encontrados, ao longo de todo o processo de procura.

4.7 O Cálculo do Coeficiente de Comparação

Primeiramente o ficheiro de modelos é acedido e a informação nele registada para um deles é armazenada numa variável apropriada. Esta variável vai conter toda a informação do modelo, ou seja, de todos os seus estados estáveis incluindo as características geométricas de todos os buracos. Embora estes últimos não contribuam para o cálculo do coeficiente de "matching" no actual algoritmo (excepto para a confirmação do seu número o que constitui um critério imediato de rejeição) a sua informação foi mantida tanto no ficheiro de modelos como na estrutura que recebe o modelo, prevendo que num desenvolvimento posterior ela possa vir a ser utilizada.

De posse então das características do objecto a ser identificado e do modelo para apenas um dos estados estáveis, resta compará-las duas a duas. Para cada par pode obter-se um desvio relativo dado por:

$$\delta_i = \frac{param_obj_i - param_model_i}{param_model_i}$$

Em função do resultado pode obter-se o respectivo coeficiente de comparação que virá entre 0 e 1 e dado por:

$$c_i = \frac{1}{1 + \delta_i \times f[i]}$$

onde $f[i]$ é o factor de fiabilidade associado à característica a ser considerada. Este factor é tanto maior quanto menor for o desvio padrão medido nas experiências de fiabilidade e o seu valor é tal que para um desvio relativo entre o valor do objecto e do modelo exactamente igual ao desvio padrão relativo, dá um coeficiente de comparação de **0,8**:

$$\delta_i = \sigma_i \Rightarrow c_i = 0,8 \rightarrow 80\%$$

onde δ_i é o desvio relativo medido e σ_i é a razão entre o desvio padrão e o valor da característica, obtidos nas medições de fiabilidade dos parâmetros.

Este valor de 0,8 é puramente convencional. Ele define a fronteira entre o certo e o duvidoso, se bem que, contemplando a possibilidade de o objecto se encontrar numa posição desfavorável para a medida dos parâmetros (por exemplo num dos casos de maior erro considerados na medida da tolerância) então dever-se-á convencionar um outro valor para c_i como limite de identificação. A fracção $3/2$ de σ_i parece ser adequada o que utilizando a fórmula acima dá um coeficiente aproximadamente igual a **0,73**:

$$c_i = \frac{1}{1 + \delta_i \times f[i]} = \frac{1}{1 + \left(\frac{3}{2}\sigma_i\right) \times f[i]} = \frac{1}{1 + \frac{3}{2} \cdot (\sigma_i \times f[i])} = \frac{1}{1 + \frac{3}{2} \cdot (0,25)} \approx 0,73$$

Toma-se então este novo valor como fronteira para o inaceitável. Ou seja, se um dos coeficientes obtidos for inferior àquele valor, então o objecto é considerado como não identificado.

O uso do factor de fiabilidade em cada uma das comparações apenas serve para verificar se o desvio da característica medida em relação à registada no ficheiro de modelos é superior ou inferior ao desvio padrão apresentado por essa característica (se c_i for inferior a 0,8 tal como convencionado, então ele é maior que o desvio padrão). A sua maior utilidade será apresentada de seguida e consiste na atribuição de pesos para os vários coeficientes de comparação. Quanto maior for o seu valor, maior peso terá o respectivo valor de c_i no cálculo do coeficiente de comparação final! Ou seja:

$$C = 100 \times \frac{\sum_i c_i \times f[i]}{\sum_i f[i]}$$

em que o denominador é apenas o factor normalizador da expressão. C dá então o coeficiente final de comparação, já em percentagem, entre o objecto e modelo num dos seus estados estáveis.

O cálculo de cada um daqueles factores $f[i]$ é então efectuado da seguinte maneira:

$$0,8 = \frac{1}{1 + \sigma_i \times f[i]} \Leftrightarrow f[i] = \frac{0,25}{\sigma_i}$$

O que dá os seguintes valores finais, conhecidos os vários σ_i :

$$\begin{cases} \sigma(\text{perim}) = 0,135 \\ \sigma(\text{area}) = 0,030 \\ \sigma(\text{Icm}) = 0,080 \\ \sigma(\text{p_sq_q}) = 0,135 \end{cases} \Rightarrow \begin{cases} f(\text{perim}) = 0,25 / \sigma(\text{perim}) \\ f(\text{area}) = 0,25 / \sigma(\text{area}) \\ f(\text{Icm}) = 0,25 / \sigma(\text{Icm}) \\ f(\text{p_sq_a}) = 0,25 / \sigma(\text{p_sq_a}) \end{cases} \Rightarrow \begin{cases} f(\text{perim}) = 1,85 \\ f(\text{area}) = 8,33 \\ f(\text{Icm}) = 3,12 \\ f(\text{p_sq_a}) = 1,85 \end{cases}$$

Como se pode ver, os parâmetros com menor dispersão de valores são aqueles que possuem maior factor $f[i]$ e portanto também maior peso no cálculo do C final:

$$C = \frac{(1,85 \cdot \delta_{\text{perim}} + 8,33 \cdot \delta_{\text{area}} + 3,12 \cdot \delta_{\text{Icm}} + 1,85 \cdot \delta_{\text{p_sq_a}})}{(1,85 + 8,33 + 3,12 + 1,85)} =$$

$$= \frac{(1,85 \cdot \delta_{\text{perim}} + 8,33 \cdot \delta_{\text{area}} + 3,12 \cdot \delta_{\text{Icm}} + 1,85 \cdot \delta_{\text{p_sq_a}})}{15,15}$$

A razão da escolha daquela expressão para o cálculo dos coeficientes de comparação individuais deve-se ao seu comportamento adequado com a variação de δ_i : ela é bastante sensível a pequenas variações de δ_i relativamente a σ_i , principalmente quando $\delta_i > \sigma_i$. Como se viu, basta o desvio relativo na medida do parâmetro ser 1,5× maior que o desvio padrão relativo, que o coeficiente de comparação resultante baixa de 0,80 para 0,73! Por outro lado, para um $\delta_i = (2/3) \sigma_i$, obtém-se um coeficiente de 0,86 (Fig.7).

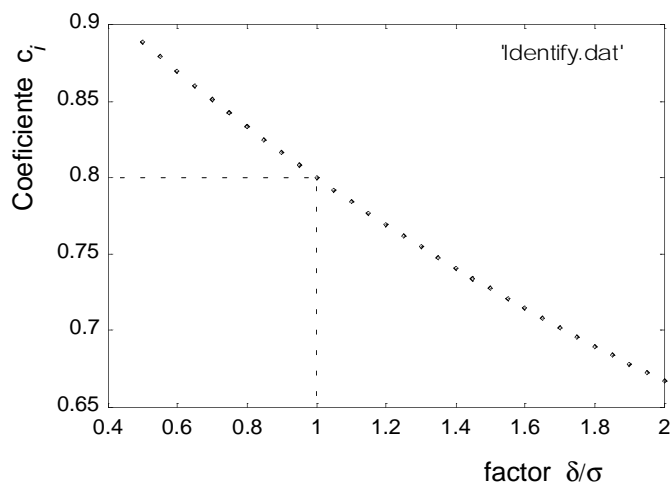


Fig. 7 - Variação do coeficiente de comparação c_i com a razão entre δ_i e σ_i .

4.8 Experiências de Identificação

Ao ficheiro de modelos já existente (e apresentado atrás) foram acrescentados os seguintes objectos:

- balança
- pêndulo
- nova_caixa
- placa
- fita_adesiva

e que se encontram representados na Fig.8.

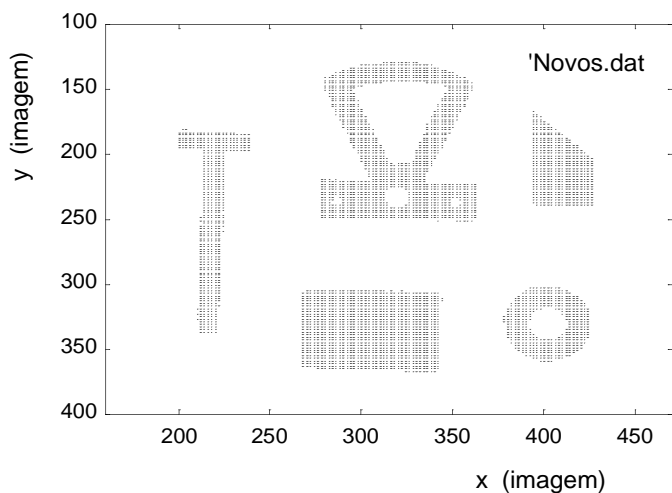


Fig. 8 - Objectos acrescentados ao ficheiro de modelos e usados nas experiências de identificação.

Todos aqueles modelos foram acrescentados ao ficheiro pelo processo automático de aquisição de valores. O total de modelos agora existente é o seguinte:

listing all model names and s.s. ...		
model nr - 1- :	açucareiro	nr of ss = 1
model nr - 2- :	caixa	nr of ss = 1
model nr - 3- :	pimenteiro	nr of ss = 1
model nr - 4- :	block_1	nr of ss = 3
model nr - 5- :	block_2	nr of ss = 3
model nr - 6- :	block_3	nr of ss = 3
model nr - 7- :	sal	nr of ss = 2
model nr - 8- :	pimenta	nr of ss = 2
model nr - 9- :	novo-1	nr of ss = 2
model nr - 10- :	novo-2	nr of ss = 2
model nr - 11- :	novo-3	nr of ss = 2
model nr - 12- :	novo-4	nr of ss = 2
model nr - 13- :	nova_caixa	nr of ss = 3
model nr - 14- :	balança	nr of ss = 1
model nr - 15- :	pêndulo	nr of ss = 1
model nr - 16- :	placa	nr of ss = 1
model nr - 17- :	fita_adesiva	nr of ss = 1

Foram então efectuadas várias experiências de identificação cujos resultados se apresentam a seguir.

4.8.1 Identificação da "balança" numa posição diferente à usada para registo de dados no Ficheiro de Modelos

Foi colocado o objecto numa nova posição, rodado de cerca de 120° no sentido negativo, tal como mostra a Fig.9, e executado o algoritmo de análise de figuras e de identificação.

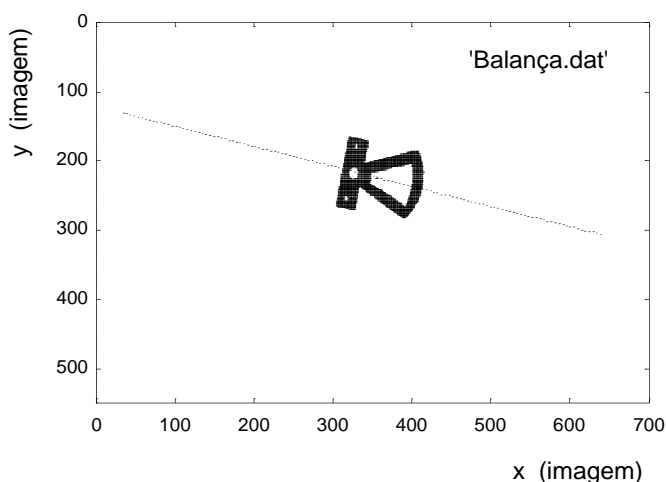


Fig. 9 - Objecto de teste nº 1: "balança".

De notar na Fig.9 que a linha de orientação desenhada é a correcta o que confirma a boa execução do programa de análise de figuras.

Como resultado da execução deste algoritmo obtiveram-se os seguintes valores:

<pre> There are 1 objects ! --- Object nr. 1 --- Features: Peri = 566 Area = 5663 Peri/sqr(Area) = 7.52130 Icm/area = 0.33344 Position and orientation: <X> = 355.5, <Y> = 224.0 Angle = 16.539° Nr of holes: 4 Hole nr. 1 Peri = 236 Area = 2054 Peri/sqr(Area) = 7.52130 <X> = 378.2, <Y> = 230.9 Icm/area = 0.06526 Hole nr. 2 Peri = 24 Area = 27 Peri/sqr(Area) = 7.52130 <X> = 316.3, <Y> = 254.6 </pre>	<pre> Icm/area = 0.00076 Hole nr. 3 Peri = 56 Area = 154 Peri/sqr(Area) = 7.52130 <X> = 325.8, <Y> = 217.7 Icm/area = 0.00441 Hole nr. 4 Peri = 24 Area = 26 Peri/sqr(Area) = 7.52130 <X> = 329.1, <Y> = 178.4 Icm/area = 0.00076 </pre>
<pre> There are 1 objects ! Position and orientation: --- Object nr. 1 --- Features: <X> = 135.6, <Y> = 35.7 Angle = 106.539° </pre>	

O 2º conjunto de valores refere-se já ao sistema de referência do robot, após conversão de coordenadas.

Como resultado da execução do algoritmo de identificação obtiveram-se finalmente os seguintes resultados:

```

identify> recognizing only 1 object...

identify> object nr -1-: -balança- (91%)

identify> The object nr -1- is a - balança-
> (state nr -1-, model nr -14-, (height=0), coef = 91%)
-or- > The object nr -1- is a - açucareiro-
> (state nr -1-, model nr -1-, (height=74), coef = 66%)

```

```
blob> nr of objects = 1; identified = 1
blob_m > 'results.Vis' stored to disk and sent to Turing !
blob_m > --- Done ---
```

O objecto, apesar da sua nova posição, foi identificado como sendo a "balança" com um coeficiente de 91%! O 2º modelo possível foi o açucareiro mas com um coeficiente de 66%, ou seja, abaixo dos 72% que constituem o limiar para a identificação.

4.8.2 Identificação do Objecto "nova_caixa"

Este novo objecto foi colocado numa posição qualquer dentro dos limites de visão da câmara, como mostra a Fig.10. Após execução do programa de análise de figuras geométricas, obteve-se não só uma lista das características medidas como também o desenho do centro da figura e uma linha indicadora da sua orientação.

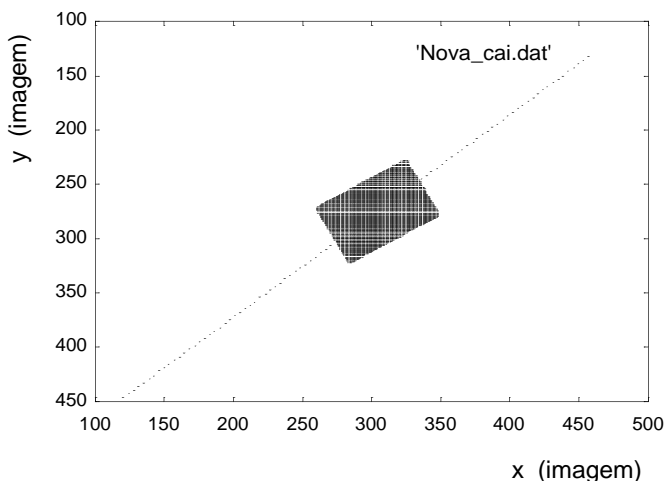


Fig. 10 - Objecto de teste nº 2: "nova_caixa".

Como se pode observar, também neste caso a linha de orientação desenhada confirma a boa execução do algoritmo.

A análise de figuras deu o seguinte resultado:

```
There are 1 objects !

--- Object nr. 1 ---

Features:
Peri = 370
Area = 4577
Peri/sqr(Area) = 5.46904
```

```
Icm/area = 0.17746

Position and orientation:
<X> = 304.2, <Y> = 274.9
Angle = -42.800°

Nr of holes: 0
```

```

There are 1 objects !

Position and orientation:

--- Object nr. 1 ---

Features:
<X> = 83.6, <Y> = -26.9
Angle = 47.200°

```

O 2º conjunto de valores refere-se, mais uma vez, ao sistema de referência do robot.

Como resultado da execução do algoritmo de identificação obtiveram-se os seguintes resultados:

```

identify> recognizing only 1 object...
identify> object nr -1-: -nova_caixa- (80%)
identify> nr of identified objects = 1

identify> The object nr -1- is a - nova_caixa-
> (state nr -1-, model nr -13-, (height=0), coef = 80%)
-or- > The object nr -1- is a - balança-
> (state nr -1-, model nr -14-, (height=0), coef = 44%)

blob> nr of objects = 1; identified = 1
blob_m > 'results.Vis' stored to disk and sent to Turing !
blob_m > --- Done ---

```

Neste caso e apesar da orientação do objecto também ser diferente da utilizada para o registo de dados no ficheiro de modelos, o algoritmo conseguiu identificá-lo como sendo a "nova_caixa" (no estado estável nº 1) com um coeficiente de comparação de 80%! O modelo mais próximo é a "balança" com 44%: valor muito abaixo do limiar de identificação que é de 72%!

De notar que para estas identificações não foi utilizado o número de buracos como critério de rejeição para que se pudesse obter um coeficiente de comparação para o segundo melhor modelo, diferente de zero. Deste modo pôde-se verificar a grande diferença entre os dois coeficientes de comparação para os dois melhores modelos.

4.8.3. Identificação do Objecto "fita_adesiva"

O objecto agora utilizado encontra-se representado na Fig.11. A linha de orientação mostrada, embora seja o resultado dos cálculos efectuados pelo algoritmo, não tem significado uma vez que o objecto possui simetria de revolução.

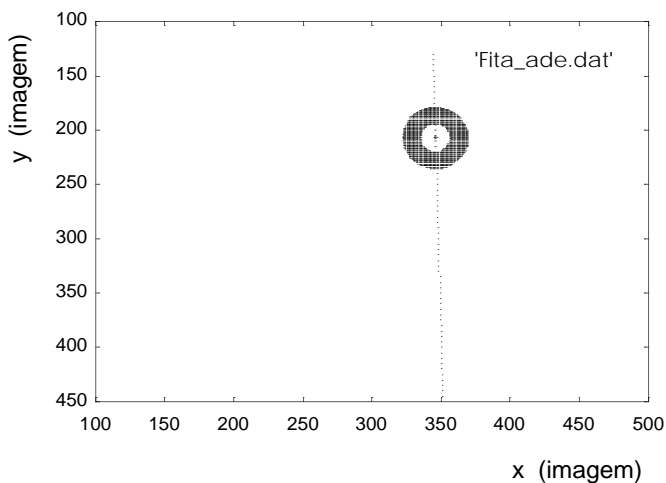


Fig. 11 - Objecto de teste nº 3: "fita_adesiva".

As características geométricas medidas são as seguintes:

```

There are 1 objects !

--- Object nr. 1 ---

Features:
Peri = 210
Area = 1742
Peri/sqr(Area) = 5.03147
Icm/area = 0.23892

Position and orientation:
<X> = 345.8, <Y> = 207.2
Angle = 88.380°

Nr of holes: 1

Hole nr. 1
Peri = 96
    
```

```

Area = 417
Peri/sqr(Area) = 5.03147
<X> = 345.5, <Y> = 206.8
Icm/area = 0.03886
    
```

```

There are 1 objects !

Position and orientation:

--- Object nr. 1 ---

Features:
<X> = 152.8, <Y> = 23.9
Angle = 178.380°
    
```

O 2º conjunto de valores refere-se, mais uma vez, ao sistema de referência do robot.

Como resultado da execução do algoritmo de identificação obtiveram-se os seguintes resultados:

```

identify> recognizing only 1 object...
identify> object nr -1-: -fita_adesiva- (91%)
identify> nr of identified objects = 1

identify> The object nr -1- is a - fita_adesiva-
      > (state nr -1-, model nr -17-, (height=0), coef = 91%)
-or- > The object nr -1- is a - placa-
      > (state nr -1-, model nr -16-, (height=0), coef = 76%)

blob> nr of objects = 1; identified = 1
blob_m > 'results.Vis' stored to disk and sent to Turing !
blob_m > --- Done ---

```

A obtenção de um coeficiente superior ao limiar de identificação (72%) para o 2º melhor modelo deve-se a valores muito próximos entre parâmetros equivalentes, de que resultaria uma identificação errada caso o modelo "fita_adesiva" não constasse do ficheiro de modelos. Contudo, ele existe nesse ficheiro; e o valor de coeficiente correspondente é tal (91%) que não deixa qualquer margem para dúvidas quanto à identidade do objecto! De qualquer modo, a utilização do 2º critério de rejeição (número de buracos) nunca permitiria a aceitação daquele 2º modelo, por muito próximos que fossem os parâmetros equivalentes de identificação!

O ficheiro de resultados 'results.Vis' que é enviado para a zona de acesso comum dos agentes contém a identidade do objecto, o número do estado estável em que foi detectado, o coeficiente de comparação, a localização do seu centro de massa no referencial do robot e finalmente a sua orientação em radianos:

results.Vis

```
position(fita_adesiva, 1, 91, 166, 73, 0, 3.081405).
```

Este formato corresponde àquele que é exigido e reconhecido pelo agente planeador de alto nível (HLP).

4.9 Conclusão

Foram efectuadas experiências variadas de identificação com este algoritmo que se mostraram bastante satisfatórias. Apenas as situações em que a 3ª dimensão se mostrasse essencial é que o algoritmo falharia ou não seria eficaz totalmente, como por exemplo para objectos de diferentes alturas mas com iguais vistas de topo. Neste caso o algoritmo apresentaria ambos os modelos como possíveis e com coeficientes de comparação muito próximos. Seria portanto necessário recorrer ao agente Laser para se

obter a altura do objecto, após o que esses dois modelos seriam diferenciados, desde que o valor dessa altura estivesse especificado no ficheiro (o que não seria o caso se os modelos tivessem sido inseridos por análise automática e não tivessem sido posteriormente editados para inclusão daquela medida!).

Outra situação, esta mais drástica, é aquela em que dois ou mais objectos se encontram encostados, ou pior ainda, sobrepostos. Nesta situação o conjunto de objectos em contacto físico são vistos como um só, sendo as características geométricas medidas para toda a figura binarizada.

Esta abordagem é suposta resolver a maioria dos casos já que não são consideradas situações de grandes amontoados de objectos como no caso do "bin picking" que consiste na recolha de peças de um reservatório onde elas se encontram armazenadas. Para situações eventuais de sobreposição será utilizado um outro algoritmo a 3D, que embora imponha algumas restrições no tipo de objectos a detectar, se tenha mostrado bastante eficaz. Este constitui contudo um recurso no caso de ocorrerem casos de incapacidade de identificação a 2D, pois o seu tempo de execução é maior e contempla menos tipos de formato de objectos.

Capítulo 5 **A TRANSFORMADA DE HOUGH MODIFICADA COM PÓS-PROCESSAMENTO**

5.1 Introdução

O algoritmo correspondente à transformada de Hough modificada com pós-processamento apresentado neste capítulo, aliado ao apresentado no capítulo seguinte, constitui a 2ª opção de funcionamento do Agente Visão. Esta opção constitui uma alternativa à análise a 2D das características geométricas de figuras, e nela são analisados os segmentos de fronteira dos objectos presentes na imagem com o objectivo de os poder identificar mesmo quando se encontrem parcialmente ocultos.

A transformada de Hough foi introduzida por Paul Hough [Hough62] com o objectivo de detectar curvas em fotografias tiradas em câmaras de bolhas de hidrogénio. Com Rosenfeld [Rosenfeld69] esta técnica foi genericamente adoptada para detectar diversos tipos de padrões em imagens, especialmente quando a presença de ruído ou a falha de informação torna difícil a aplicação de outras técnicas.

Basicamente a transformada de Hough (T. de H.) converte o problema genérico de detecção de formas no espaço imagem, numa detecção de um máximo local no espaço dos parâmetros. No caso presente trata-se da detecção de segmentos que pertencem às linhas de fronteira de objectos presentes na imagem.

5.2 A Descrição Matemática da Transformada de Hough

A ideia básica da T. de H. é substituir o teste de colinearidade para cada par de pontos no plano da imagem (pertencentes a um segmento de fronteira) pela avaliação do ponto de intersecção de rectas no espaço dos parâmetros.

Se o declive m e a ordenada na origem b forem escolhidos como parâmetros, então cada linha contendo o ponto imagem (x,y) será descrita pela equação $y = mx + b$, que no espaço dos parâmetros vai ser representada pelo ponto (m,b) (Fig.1):

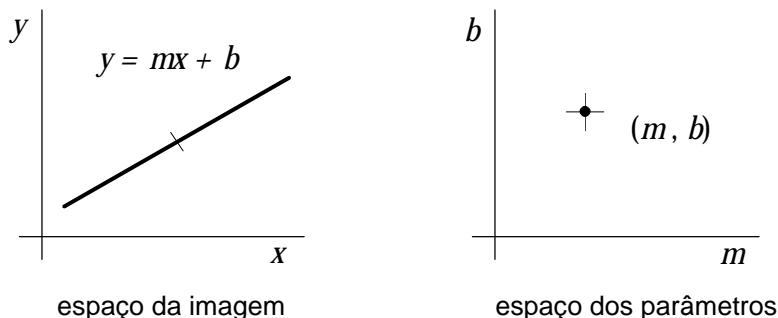


Fig. 1 - A representação de uma recta no espaço dos parâmetros.

Por outro lado, qualquer ponto no espaço imagem vai ser representado no espaço dos parâmetros por uma linha recta, uma vez que m e b serão livres de variar contanto que (x,y) se mantenha constante (Fig.2):

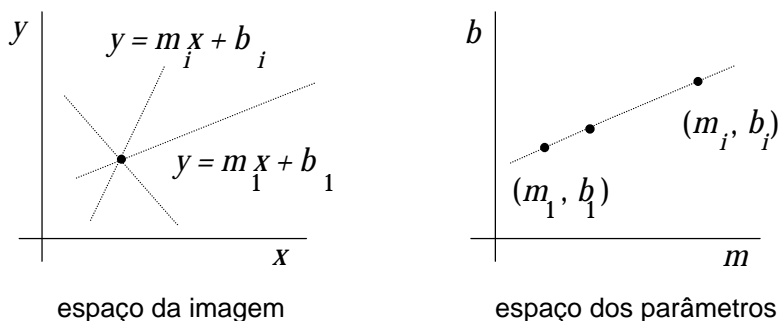


Fig. 2 - A representação de um ponto no espaço dos parâmetros.

O valor correcto dos parâmetros (m,b) para um conjunto de pontos colineares será então o par de valores para os quais todas as linhas no espaço dos parâmetros (cada uma representando todas as possíveis rectas que passam por cada ponto da imagem) se intersectam naquele espaço.

Na prática é impossível considerar todas as linhas que passam por um dado ponto (o seu número é infinito) pelo que o espaço dos parâmetros tem que ser discretizado em intervalos de largura dm e db . Aliás esta discretização é essencial para que possa haver tratamento computacional do problema. Para cada valor de b , m é calculado usando a equação geral da recta (considerando um ponto de cada vez), daí resultando um ponto (m,b) que é representado no espaço dos parâmetros. Para outros valores de b , outros valores de m serão obtidos que irão desenhar uma recta naquele espaço. Para outros pontos (x,y) , outras rectas serão obtidas que se irão cruzar num único ponto: aquele que fornecerá os parâmetros da recta que contém o segmento de fronteira no espaço imagem.

Como é que na prática se simula a intersecção das rectas? Uma vez que o espaço se encontra agora discretizado, este vai poder ser representado por uma matriz de valores com dimensão $a \times c$, onde $a = \Delta m / dm$ e $c = \Delta b / db$ (sendo Δm e Δb a gama de valores cobertos respectivamente por m e b). Esta matriz é também chamada de tabela de acumulação, pois para cada par (m,b) obtido por cálculo, a sua representação no espaço dos parâmetros vai corresponder na prática ao incremento de uma célula naquela tabela. Dado então um certo ponto (x,y) consideram-se todos os valores de b possíveis (correspondentes a todos os intervalos discretos daquela grandeza, ou o que é

equivalente, a todas as divisões na tabela de acumulação) e mantendo fixos (x,y) calculam-se os respectivos valores de m e incrementam-se todas as células (m,b) obtidas. Desta forma obtêm-se todos os incrementos correspondentes à recta no espaço dos parâmetros. Repetindo o processo para todos os pontos do segmento obtêm-se todos os incrementos associados a todas as "rectas". A intersecção dessas "rectas" vai ser dado pela célula mais incrementada!

Uma outra forma de interpretar este resultado é a seguinte: para cada ponto do segmento são experimentadas todas as rectas possíveis permitidas pela discretização do espaço; mas para todas elas há um (m,b) que se repete sempre: ele é, precisamente, o da recta experimentada que corresponde ao segmento de recta real e por isso vai dar origem a um incremento sistemático da célula respectiva pelo que ela exibirá um máximo; a selecção deste máximo vai originar a escolha correcta da recta que corresponde ao segmento de teste (Fig.3).

<i>b</i>	2																		
	2	2																	
	2	2	2																
	2	3	3	3															
	2	2	4	6	5														
		1	1	2	5	10	4	2	1	1									
							6	5	3	2									
								3	4	3									
										2	2								
																			2
																			<i>m</i>

Fig. 3 - O "ponto" de intersecção das várias "rectas" no espaço dos parâmetros vai corresponder ao par (m,b) do segmento na imagem [Illingworth88].

A Fig.4 ilustra ainda uma possível tabela de acumulação em forma de histograma bidimensional:

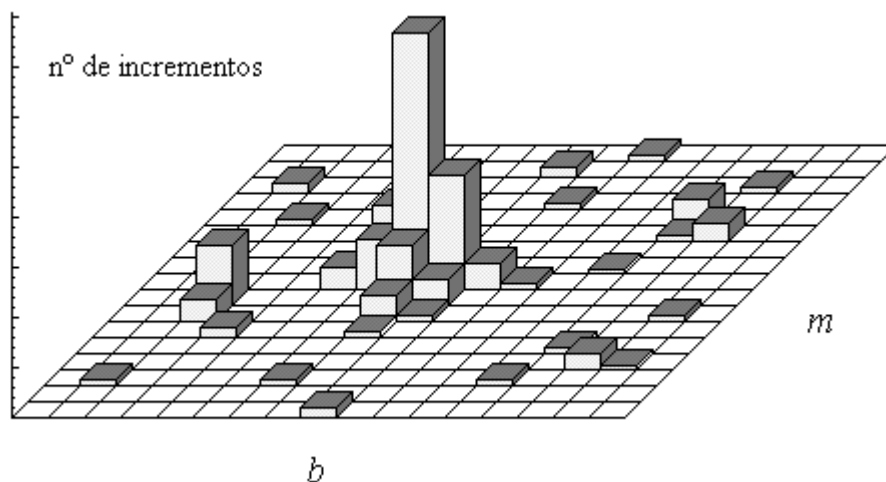


Fig. 4 - Ilustração de uma possível tabela de acumulação na forma de histograma bidimensional.

5.3 Os Primeiros Melhoramentos

5.3.1 O Registo Actualizado das Coordenadas dos Pontos Extremos de cada Segmento

Devido à necessidade de discretização, os valores encontrados para m e b não vão corresponder exactamente aos reais. E quanto maior for essa discretização (ou seja, quanto menor for o número de células da tabela de acumulação) maior vai ser a sua diferença. Os valores correctos podem, contudo, ser recuperados, se informação extra for registada naquela tabela, para além do número de incrementos. As coordenadas reais dos pontos extremos de cada segmento, parecem ser um dado adequado à futura avaliação exacta de (m,b) . Assim a tabela de acumulação serviria apenas para detectar a existência de um segmento devido ao seu número elevado de incrementos. No entanto a avaliação de (m,b) já não seria dado pelos valores associados à célula, mas sim obtida pela substituição das coordenadas referidas na equação geral da recta: $y = mx + b$.

5.3.2 A Escolha de Novos Parâmetros

A escolha dos parâmetros (m,b) parece não ser a mais aconselhável uma vez que ambos os seus valores não são limitados: para rectas verticais eles seriam infinitos. Assim, nem todas as rectas poderiam ser contempladas pelo algoritmo se algum limite tivesse que ser imposto, o que não parece razoável. Sendo assim, em [Duda77] sugere-se a utilização de dois outros parâmetros: ρ e θ que traduzem a recta em coordenadas cilíndricas. A equação geral de uma recta com estes parâmetros é a seguinte: $\rho = x \cos\theta + y \sin \theta$. O modo de os obter é ilustrado na Fig.5:

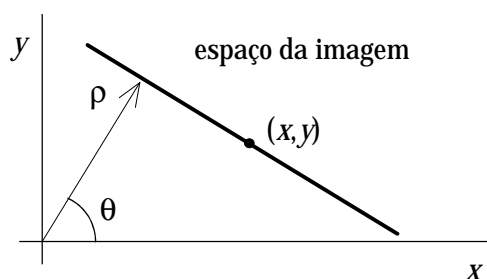


Fig. 5 - Obtenção dos novos parâmetros (ρ,θ) de uma recta.

Estes parâmetros mantêm as propriedades de transformação que já foram referidas atrás, tendo a vantagem dos seus valores serem limitados, o que permite representar totalmente o seu espaço: admitindo que a imagem corresponde ao 1º quadrante do espaço imagem, então ρ variará entre 0 e o valor da diagonal e θ variará entre $-\pi/2$ e π . A única diferença para o caso anterior é que cada ponto imagem é agora representado no espaço dos parâmetros por uma sinusóide em vez de por uma recta. Os valores aproximados (ρ,θ) são agora obtidos na intersecção daquelas sinusóides.

A Fig.6 ilustra um resultado obtido por [Duda77] no qual se podem detectar alguns máximos correspondentes a segmentos reais da imagem.

		θ								
		2	.
		1	2	.
		4	4	.
		.	.	2	.	.	.	6	5	.
		.	.	.	2	.	.	8	4	2
		6	6	2
		4	3	3
		.	2	.	1	.	1	4	2	3
		.	.	1	4	.	12	4	3	5
		.	.	3	.	2	14	2	3	4
		.	.	1	.	.	11	1	2	4
		5	2	.	2	4
		1	.	3	9
	4	1	.	.	11	9	1	8	12	
	4	3	.	3	10	12	3	10	15	
	9	5	.	4	5	4	5	11	12	
	6	6	.	4	10	.	11	9	14	
	4	9	.	4	20	2	11	10	8	
	5	6	.	2	10	3	11	13	8	
	8	4	4	4	.	2	13	10	10	
	4	7	14	3	.	.	1	6	8	
	4	18	2	5	.	1	12	10	8	
	9	17	21	15	.	25	18	7	8	
	8	20	21	13	.	22	11	11	7	
	12	17	22	17	.	9	10	9	10	
	38	14	17	17	38	8	7	9	6	
	37	16	22	21	42	10	5	9	9	
	35	11	21	23	23	8	11	9	10	
	13	18	18	23	20	14	13	9	9	
	7	16	12	30	20	20	7	9	6	
	7	18	12	32	19	27	8	7	8	
	8	12	11	20	17	52	11	6	7	
	7	17	12	23	8	11	15	11	10	
	9	14	12	16	7	7	14	6	7	
	9	12	12	16	6	9	16	12	7	
	8	13	13	11	7	10	16	14	10	
ρ	10	9	15	11	7	10	16	13	6	
	12	11	13	14	40	10	16	13	13	
	10	10	16	14	8	9	14	2	22	
	10	8	22	12	41	6	7	12	21	
	11	12	15	11	23	6	11	14	14	
	13	1	1	8	18	7	1	16	15	
	10	14	17	11	7	8	9	10	12	

Fig. 6 - Exemplo de uma tabela de acumulação no caso da abordagem clássica. (tabela retirada de [Duda77])

A escolha de um limiar no número de incrementos permitirá seleccionar os máximos.

No exemplo apresentado, embora se consigam discernir alguns máximos, a sua correcta selecção implica o conhecimento *a priori* do seu número, como o admitem os próprios autores de [Duda77]. De facto a sua selecção apresenta muitas dificuldades dado o grande número de máximos que ocorrem. Numa situação em que o número real de segmentos não seja conhecido, como é o caso da situação mais geral, estes resultados não são satisfatórios pelo que deverão ser introduzidos melhoramentos no algoritmo.

5.3.3 Os Operadores de Sobel

Outro grande melhoramento está, contudo, ainda para ser introduzido e encontra-se referido em [Ramos89]. Trata-se da utilização dos operadores de Sobel, cuja introdução vai originar uma diminuição considerável no tempo de execução computacional para além de originar máximos mais distintos.

A grande vantagem no uso destes operadores, que consistem em duas máscaras com dimensão 3×3, está no facto da sua aplicação fornecer de imediato um valor aproximado dos parâmetros pretendidos (ρ, θ), substituindo as $c = \Delta b / db$ operações por uma única. O resultado irá incrementar somente 1 célula. Daqui resulta, como já foi referido, um aumento substancial na velocidade de execução assim como uma tabela de acumulação mais "limpa".

O conteúdo das máscaras é o apresentado na Fig.7:

-1		1
-2		2
-1		1

h

-1	-2	-1
1	2	1

v

Fig. 7 - Os operadores de Sobel utilizados na nova abordagem.

A aplicação destas máscaras num ponto de fronteira de um objecto, fornece as componentes segundo x e y do vector gradiente naquele ponto. Alguns cálculos simples fornecem de imediato os parâmetros ρ e θ pretendidos.

Os cálculos são os seguintes:

$$m = \text{sqrt}(h^2 + v^2)$$

Se,

$$\begin{cases} X_1 = x + y(v/h) & \text{excepto para linhas horizontais} \\ Y_1 = y + x(h/v) & \text{excepto para linhas verticais} \end{cases}$$

então,
$$\begin{cases} \theta = \text{atan}(v / h) & \text{excepto para linhas horizontais} \\ \rho = X_1(h / m) & \text{para linhas quase verticais} \\ \rho = Y_1(v / m) & \text{para linhas quase horizontais} \end{cases}$$

A tabela de acumulação apresentada na Fig.9, obtida por aplicação daqueles operadores a uma imagem real esquematizada na Fig.8, ilustra bem o que atrás foi dito (ela apresenta incomparavelmente menos células incrementadas que a da Fig.6).

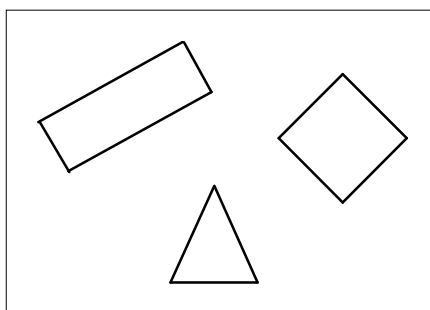


Fig. 8 - Esquematização da imagem sujeita à aplicação da Transformada de Hough com os operadores de Sobel.

5.4 Proposta de Novos Melhoramentos

No sentido de melhorar aqueles resultados, nomeadamente de reduzir o número de células incrementadas, introduzi em [Lopes93a] uma nova máscara de 5x5. Esta máscara, dado que actua em mais pontos da imagem que a anterior, permite avaliar mais correctamente o valor aproximado do par (ρ, θ) . No caso da Fig.10, por exemplo, os operadores de Sobel iriam detectar um segmento horizontal o que não corresponde à realidade. A nova máscara, por seu lado, já detecta uma inclinação não nula.

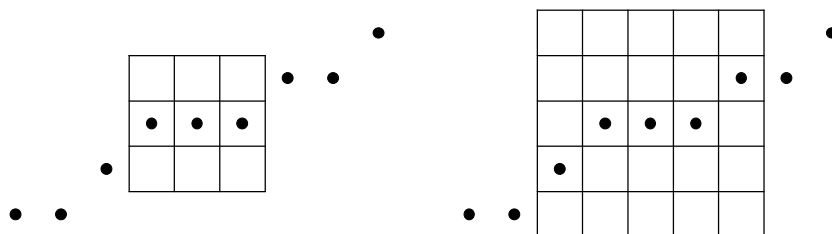


Fig. 10 - O uso das novas máscaras de 5x5 permitem obter valores de (ρ, θ) mais correctos.

		θ							
	0	0	0	0	0	0	0	1	0
	0	0	0	10	0	0	0	0	0
	0	0	0	4	0	0	0	0	0
	0	0	0	0	0	0	0	0	10
	3	0	10	0	0	0	0	1	0
	1	0	0	0	0	0	0	0	8
	0	0	11	0	0	0	0	0	3
	0	0	0	0	0	0	0	0	0
	1	0	0	1	0	0	0	0	0
	4	0	0	0	0	0	0	0	0
	0	0	26	4	0	0	0	0	0
	0	0	0	10	4	0	0	0	0
	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	22	0	0	0
	0	0	1	0	0	39	0	0	0
	0	34	26	0	2	0	1	0	0
	0	0	0	0	1	24	0	0	0
	0	0	0	0	0	38	0	0	0
	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
	0	31	0	0	0	0	0	0	0
	0	0	0	7	0	0	0	0	0
	0	0	0	17	0	0	0	0	0
	0	0	0	9	0	0	0	0	0
	0	0	1	10	0	0	0	0	0
	0	0	0	5	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	0	14	0	0
	0	0	0	0	10	1	0	0	0
	0	0	0	0	5	0	0	0	0
	0	0	0	36	0	0	0	0	0
	0	0	0	9	1	0	0	0	0
	0	0	2	17	3	0	0	0	0
	0	0	0	17	11	1	0	0	0
	0	0	0	0	0	0	0	0	0
	0	0	0	0	0	1	0	0	0
	0	0	0	0	0	0	0	0	0
	0	0	0	37	1	0	0	0	0
	0	0	0	0	0	0	0	0	0

Fig. 9 - Tabela de acumulação obtida por aplicação dos operadores de Sobel à imagem esquematizada anteriormente.

A discretização do espaço dos parâmetros aliado à incorrecta avaliação de (ρ, θ) provocam um efeito altamente indesejado: o espalhamento da informação de um segmento ao longo de ρ ! Como se pode observar na Fig.11, a inclinação mal avaliada no 1º caso, dá origem, à medida que os pontos do segmento real vão sendo analisados, à modificação progressiva no valor de ρ , mantendo θ constante. No 2º caso da mesma figura, embora o referido espalhamento se mantenha, ele é menor se os valores de (ρ, θ) calculados pelo uso das novas máscaras for mais próximo dos correctos. Este espalhamento, embora atenuado, vai continuar a existir com especial notoriedade para segmentos longos. Ele vai aliás constituir, como se verá adiante, um dos maiores problemas na correcta detecção e reconstrução dos segmentos e que será alvo de especial atenção.

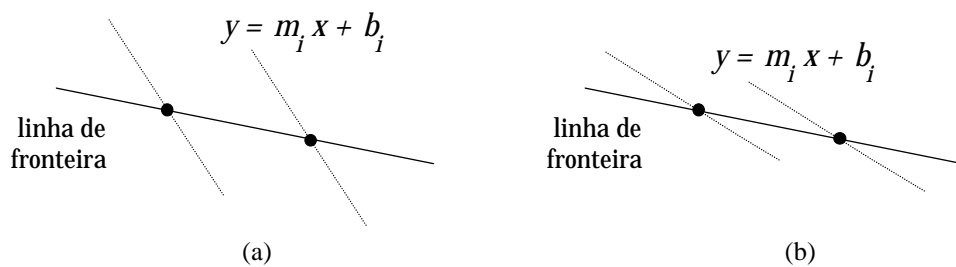


Fig. 11 - Justificação do efeito de espalhamento dos incrementos ao longo de ρ .
 (a) avaliação incorrecta de $(\rho, \theta) \Rightarrow$ maior espalhamento.
 (b) avaliação mais correcta de $(\rho, \theta) \Rightarrow$ menor espalhamento.

Faltou ainda escolher o conteúdo das duas máscaras de 5×5 , tarefa que se mostrou algo problemática. No sentido de encontrar os mais adequados, foram realizadas inúmeras experiências com conteúdos distintos. Após a análise do espalhamento na tabela de acumulação provocado por todas as máscaras experimentadas (para um mesmo conjunto de objectos de teste), decidi adoptar aquelas que apresento na Fig.12.

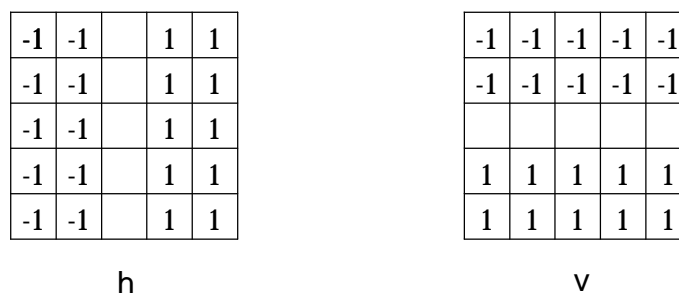


Fig. 12 - Máscaras de 5×5 adoptadas para substituir os operadores de Sobel.

Devo ainda referir que foram também experimentadas máscaras com maior dimensão até 11×11 e com diversos conteúdos. Contudo, e embora o tempo de execução não aumentasse muito com o aumento na dimensão das máscaras, não constatei melhoria nos seus resultados. Sendo assim, adoptei aquelas duas máscaras em definitivo.

Outro dos melhoramentos introduzidos foi o de detectar os pontos de fronteira com duas novas máscaras de 2×2 . Os operadores de Sobel na abordagem anterior são utilizados simultaneamente para detectar os pontos de fronteira (dado que fornecem as componentes segundo x e y do vector gradiente), e para calcular a inclinação da fronteira nesses pontos. Ora, a tarefa de encontrar aqueles pontos pode ser executada por máscaras mais simples, rápidas e igualmente eficazes. Para além disso as recém adoptadas máscaras de cálculo com dimensão 5×5 não são de modo nenhum apropriadas para o efeito. Uma vez que as imagens submetidas a análise são previamente binarizadas, as transições são de muito fácil detecção pelo que decidi adoptar as novas máscaras apresentadas na Fig.13:

$$\begin{array}{cc}
 \begin{array}{|c|c|}
 \hline
 -1 & 1 \\
 \hline
 -1 & 1 \\
 \hline
 \end{array} &
 \begin{array}{|c|c|}
 \hline
 1 & 1 \\
 \hline
 -1 & -1 \\
 \hline
 \end{array} \\
 h & v
 \end{array}$$

Fig. 13 - Novas máscaras para detecção de fronteira em imagens binarizadas.

O uso destas dimensões, aliás, permite esqueletizar a fronteira, ou seja, apenas um ponto é considerado ao longo do gradiente; ou considerando que o varrimento na imagem é feito da esquerda para a direita, os pontos imediatamente à direita do ponto de fronteira considerado, nunca são alvo de cálculo. Desta forma o número de incrementos vai traduzir sempre o comprimento de um segmento ignorando a sua largura. O sucesso desta detecção pode ser confirmado sempre, à medida que o varrimento na imagem vai sendo feito, através do registo na imagem de pixeis com o nível 255 para pontos detectados como transição e pixeis com nível 0 para os restantes.

5.5 Dificuldades Remanescentes na Selecção dos Máximos

Estas alterações, embora reduzam o espalhamento de incrementos na tabela de acumulação pela medição mais correcta dos parâmetros (ρ, θ) , não evitam, contudo, a ocorrência de erros na detecção correcta dos segmentos. Apresento de seguida os vários problemas encontrados e as situações mais desfavoráveis que lhes dão origem:

- Em geral existe dificuldade na selecção do limiar correcto de selecção, a menos que se defina um comprimento mínimo para os segmentos o que não me parece muito eficiente; o resultado seria a falha na detecção de segmentos pequenos.
- Segmentos colineares vão incrementar a mesma célula dando origem à selecção de apenas um deles (ilustrada na secção seguinte).

- A ocorrência de ruído pode também originar incrementos na célula de um determinado segmento, alterando-lhe os respectivos valores das coordenadas dos extremos (ilustrada na secção seguinte).
- Segmentos diferentes mas pertencentes a objectos muito pequenos podem incrementar a mesma célula, dando origem a um número errado de segmentos detectados. O exemplo na Fig.13 ilustra esta ocorrência.
- segmentos muito próximos podem também incrementar a mesma célula, mesmo que possuam comprimentos apreciáveis.
- observa-se o espalhamento segundo ρ dos incrementos provocados por um mesmo segmento, caso ele possua um comprimento médio ou grande, formando assim um aglomerado de células pertencentes ao mesmo segmento (Figs. 16 e 17).
- a própria informação sobre o segmento, nomeadamente o seu início e fim, encontra-se distribuída por esse aglomerado, pelo que a selecção de apenas 1 máximo por aglomerado não constitui solução (Fig.14).

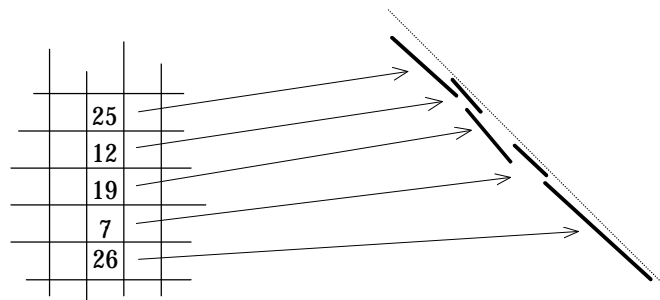


Fig. 14 - Espalhamento dos incrementos ao longo de ρ .

- se os segmentos estiverem suficientemente próximos, pode ocorrer a sobreposição parcial daqueles aglomerados, originando erro na sua correcta detecção. O exemplo concreto da Fig.21 ilustra essa ocorrência.

5.6 Apresentação de Resultados Experimentais

Apresento de seguida algumas tabelas de acumulação obtidas experimentalmente com objectos reais (não foi efectuado qualquer tipo de simulação), após execução do algoritmo de Hough modificado. Nestas tabelas poder-se-ão observar todos os efeitos descritos acima, e em especial o espalhamento segundo ρ .

Estas tabelas são acompanhadas com um desenho do objecto utilizado e ainda com os segmentos detectados.

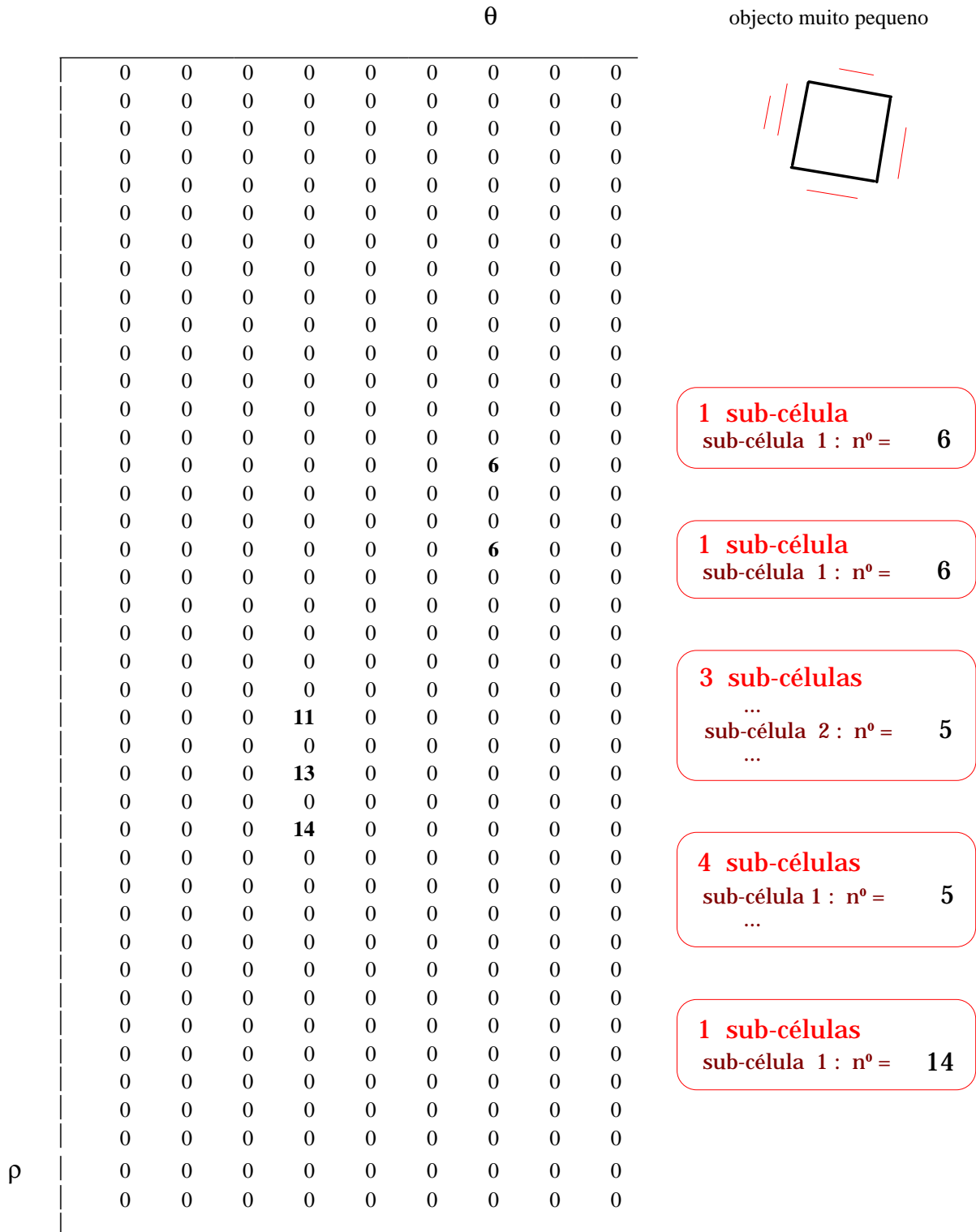


Fig. 15 - Neste caso apenas existem 3 células incrementadas: uma delas recebe informação de dois segmentos. O desenho apresentado é uma representação de um objecto real sujeito ao algoritmo desenvolvido; não foram efectuadas simulações. Os traços que o acompanham correspondem aos segmentos detectados e extraídos dos máximos encontrados; eles pretendem mostrar o seu real comprimento em comparação com o segmento que lhes deu origem; a sua posição na imagem seria coincidente com cada um deles.

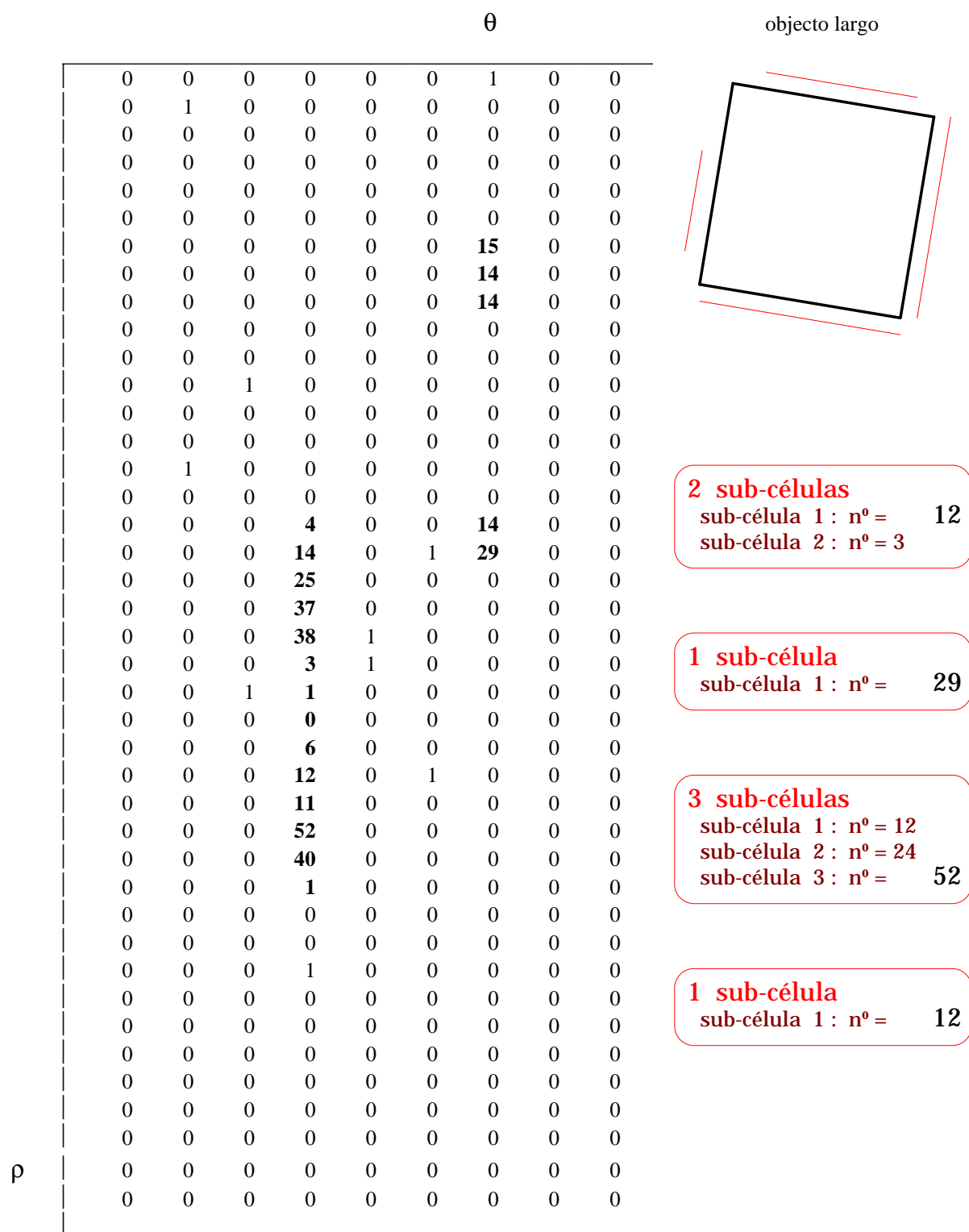


Fig. 16 - Neste caso, o espalhamento em ρ provocou a sobreposição dos dois aglomerados correspondentes a dois lados paralelos. De notar que não é possível, portanto, escolher um limiar de selecção de máximos ou células: cada segmento vai incrementar mais do que 1 célula; por outro lado a sobreposição de aglomerados não permite que se escolha apenas um máximo por aglomerado, pois perder-se-ia a informação de um dos segmentos.

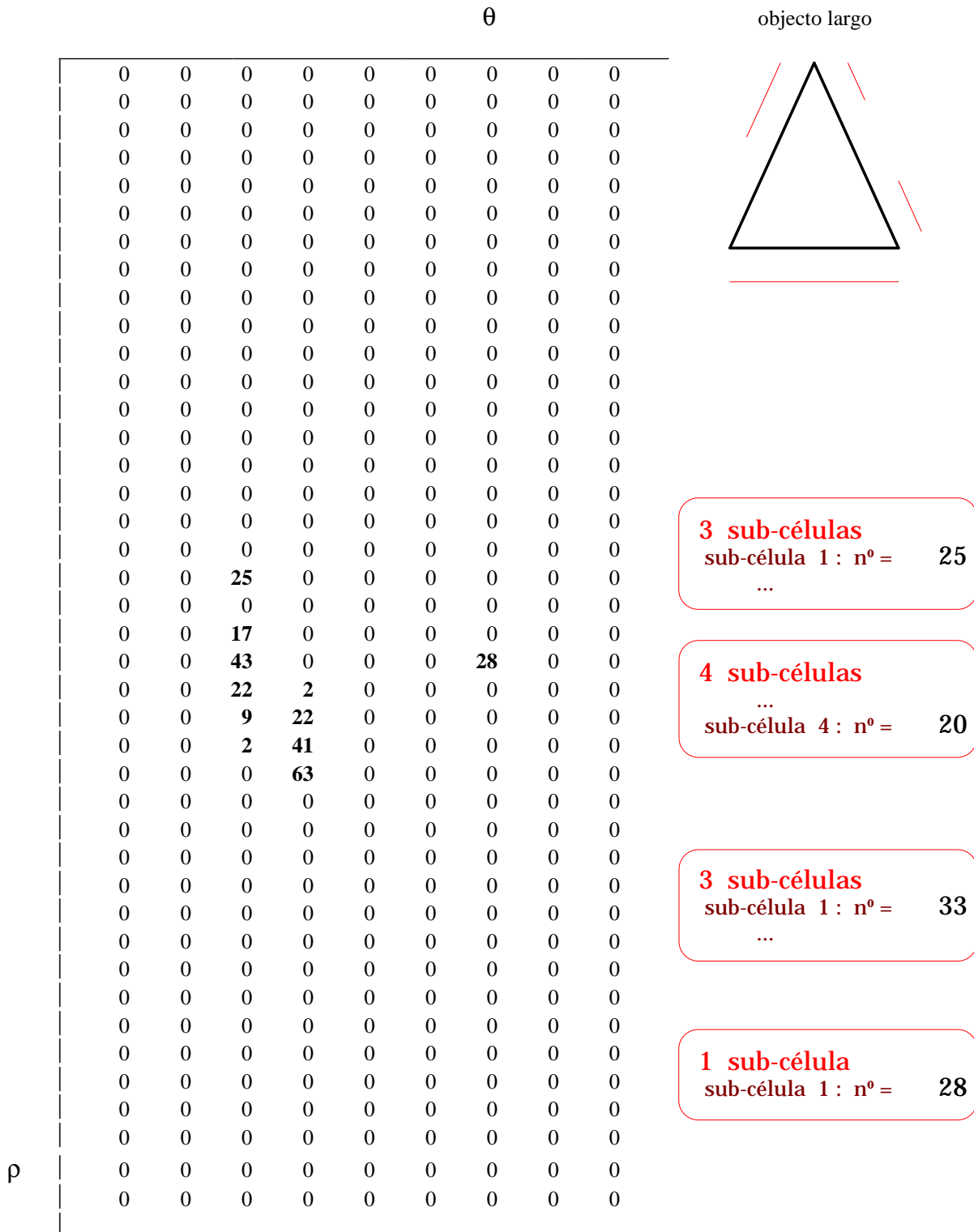


Fig. 17 - Escolher 1 célula por aglomerado é errado: neste caso existem 4 aglomerados para 3 segmentos. Por outro lado, ignorar as outras células do aglomerado é também errado: parte da informação do segmento pode ficar perdida.

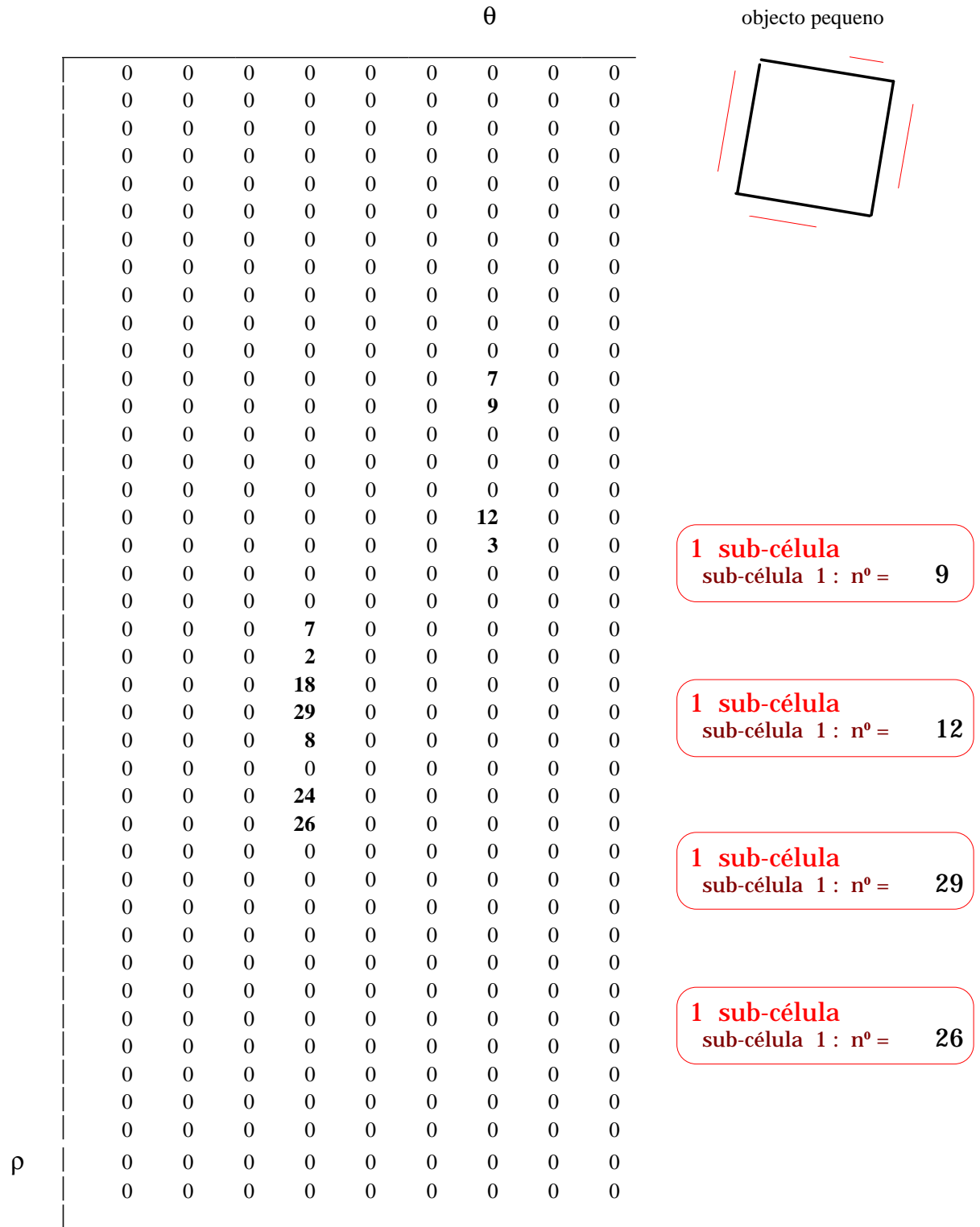


Fig. 18 - As células não devem ser seleccionados por escolha de um limiar de incrementos:
 (a) as células, mesmo que correctamente seleccionadas, podem conter apenas parte da informação do segmento respectivo
 (b) podem ser seleccionadas células que não correspondam a segmentos ou sejam redundantes
 (c) neste caso, dois segmentos poderiam ficar perdidos uma vez que as células respectivas possuem poucos incrementos

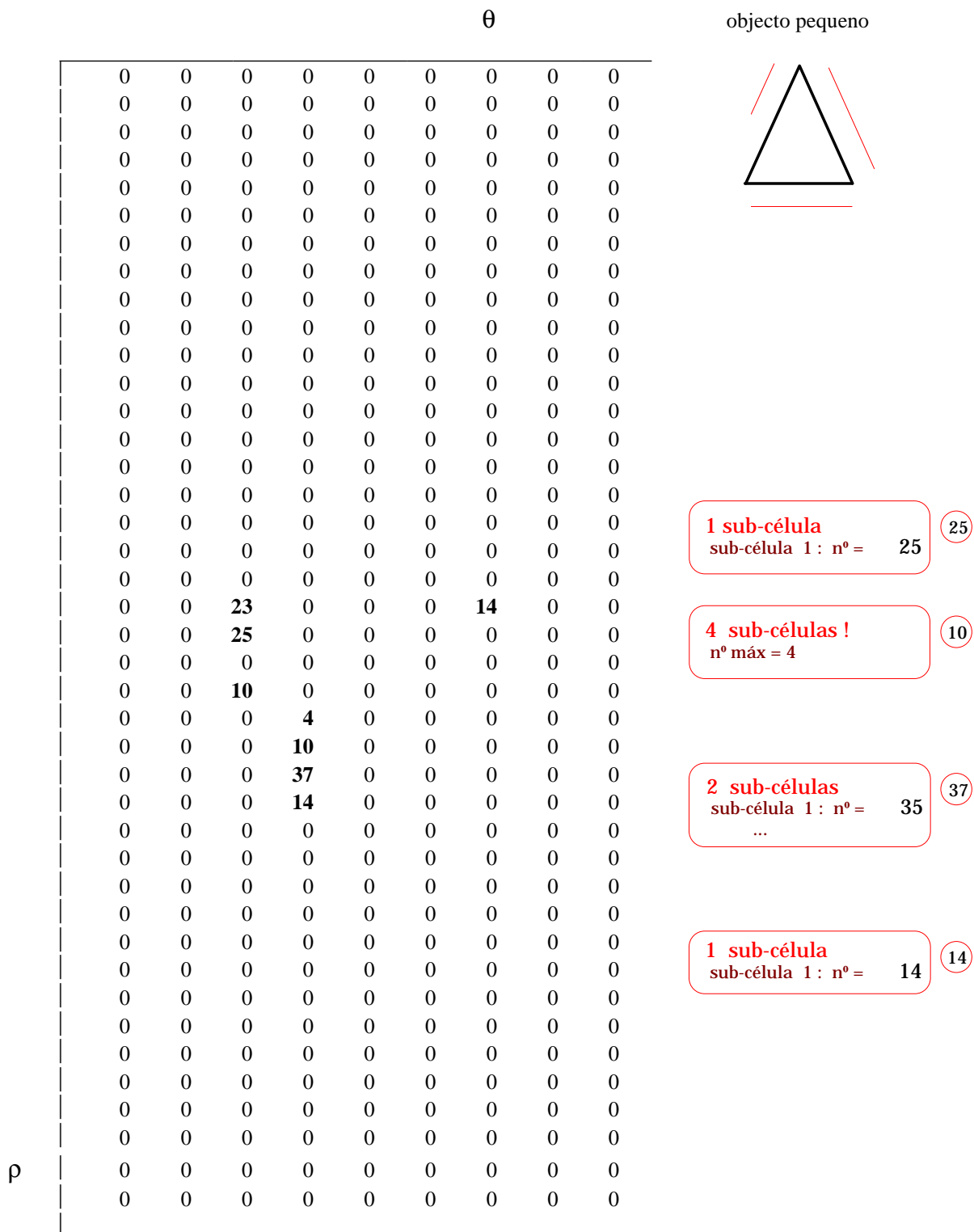


Fig. 19 - Se for seleccionada 1 célula por aglomerado, são encontrados 4 segmentos.
 Se as restantes células do aglomerado forem rejeitadas, parte dos segmentos fica perdido.

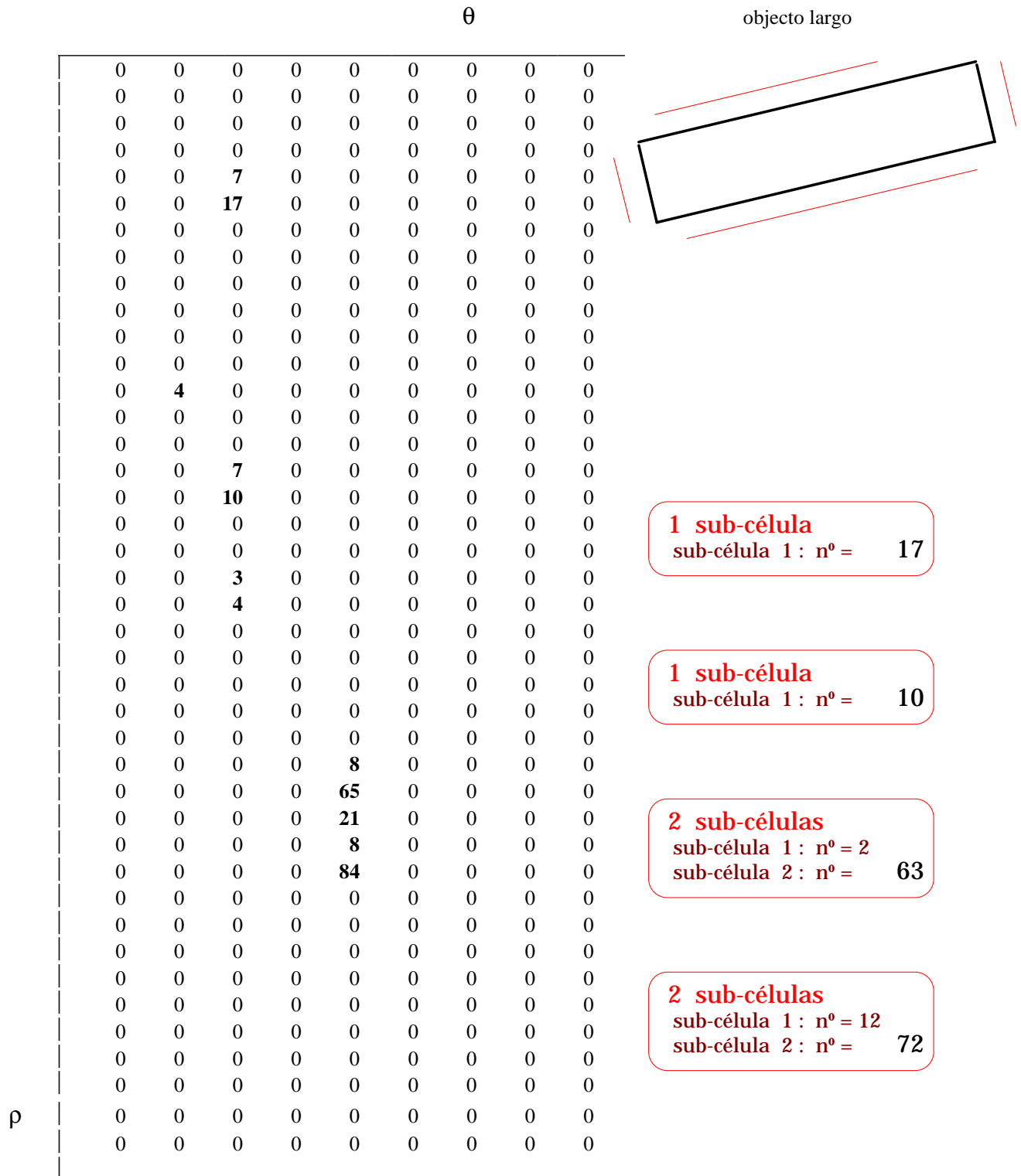


Fig. 20 - Dois aglomerados encontram-se parcialmente sobrepostos. Se fosse escolhido apenas um máximo por aglomerado, então ambos os segmentos seriam interpretados como sendo apenas um.

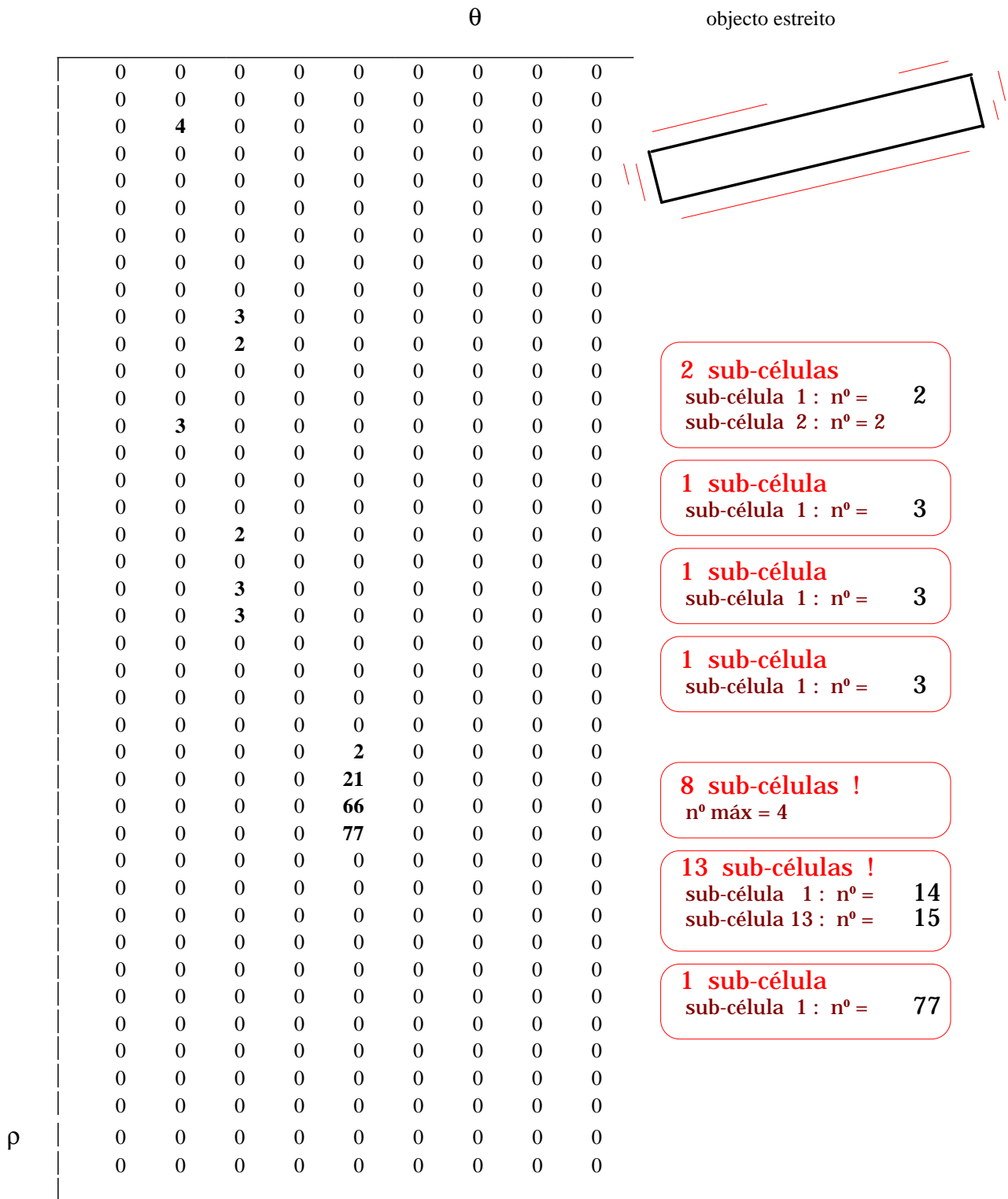


Fig. 21 - Dois aglomerados estão parcialmente sobrepostos e ambos os segmentos longos incrementam a mesma célula

5.7 A Solução Encontrada (modificação da abordagem de Hough)

O registo em cada célula da tabela de acumulação das coordenadas dos pontos extremos do segmento respectivo, e o uso de novas máscaras, não são ainda suficientes para se obter bons resultados. Cada célula vai estar associada a uma recta com parâmetros (ρ, θ) . Ora, pode existir na imagem mais do que um segmento pertencente a essa recta, pelo que ambos vão incrementar a mesma célula (Fig.22). A selecção dessa célula dará origem a apenas um segmento correspondente a um deles ou a um outro que ligue ambos. Em qualquer dos casos referidos o resultado é erróneo pelo que requer atenção. A própria existência de ruído na imagem pode induzir a falsa detecção de um outro segmento que lhe poderá alterar o verdadeiro comprimento (Fig.23).

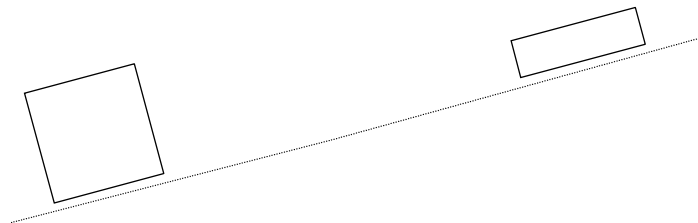


Fig. 22 - Presença de um 2º segmento com o mesmo (ρ, θ) .

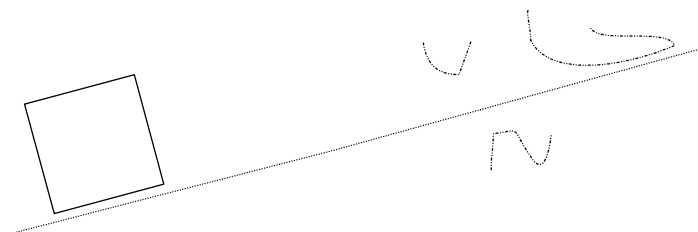


Fig. 23 - Presença de ruído que poderá originar incrementos na mesma célula e alterar os valores das coordenadas dos pontos extremos do segmento real.

A solução proposta [Lopes92, Lopes93a, Lopes93b] consiste na construção de listas ligadas de sub-células, todas associadas a uma mesma célula de parâmetros (ρ, θ) e potencialmente associadas a diferentes segmentos colineares. Assim, no caso da Fig.22, quando o 2º segmento, colinear com o 1º, fosse alvo de análise, seria verificada a distância do ponto considerado ao último registado na célula (através das coordenadas dos pontos extremos). Se essa distância fosse superior a um valor pré-definido, seria acrescentada uma outra sub-célula ligada à anterior com 1 incremento. Todos os pontos seguintes irão então incrementar esta última sub-célula; a anterior não seria mais alvo de novos incrementos. Em geral, de facto, o processo sequencial de varrimento implica que não se volte a atingir o segmento anterior a ela associado pelo que considerarei dispensável a procura da sub-célula à qual um dado incremento deverá pertencer (esse processo iria alongar excessivamente o tempo de execução, especialmente se houvesse muitas sub-células associadas a uma dada célula da tabela de acumulação). A única situação em que tal poderia acontecer seria para segmentos quase colineares e quase horizontais (Fig.24): nesta situação deverá criar-se uma 3ª sub-célula que posteriormente poderá fundir com a 1ª.

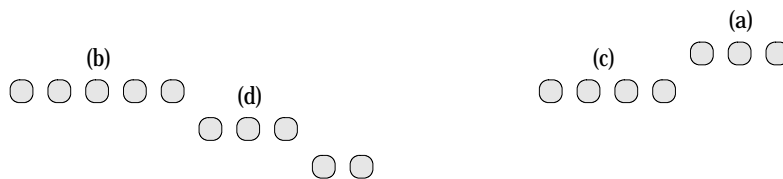


Fig. 24 - Numa situação de quase colinearidade e também quase horizontalidade, todos os pixeis vão contribuir para a mesma célula, mas criando sucessivamente várias sub-células: os 1.ºs pixeis *a* criam a 1ª; os pixeis *b* vão ter que criar outra sub-célula devido à distância *a* que se encontram dos anteriores; os pixeis *c* deveriam incrementar a mesma sub-célula dos *a*, mas para evitar um processo de procura, criam uma outra sub-célula o mesmo acontecendo com os pixeis *d*. Posteriormente estas sub-células passarão por um processo de fusão.

A Fig.25 ilustra a referida lista ligada com origem em células da tabela de acumulação, assim como as estruturas de valores que são mantidas em memória:

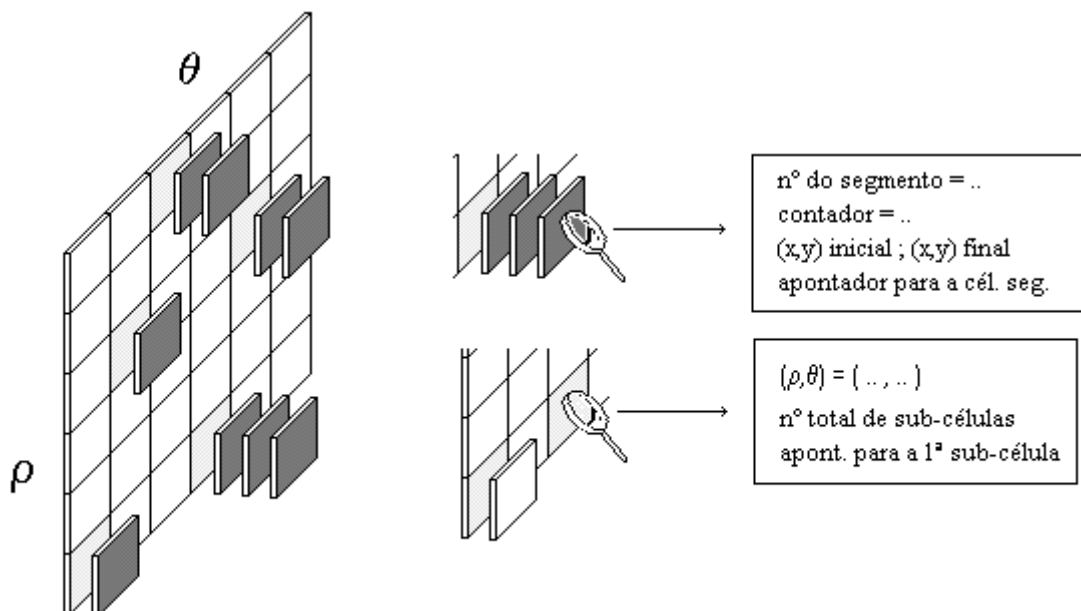


Fig. 25 - Representação da tabela de acumulação de onde emergem as sub-células respectivas. São ainda apresentadas as estruturas de valores que são mantidas em memória.

A adopção desta abordagem dispensa ainda o cuidado na escolha do limiar de selecção de máximos referida em 3.5.1, podendo adoptar-se um valor baixo por forma a contemplar pequenos segmentos que doutra forma seriam eliminados. Uma outra vantagem está na desnecessidade de efectuar ajustes cuidados na discretização do espaço dos parâmetros. Em princípio essa discretização deverá permitir discernir segmentos próximos, através do incremento de células diferentes. Idealmente o tamanho das células deveria ser o mais pequeno possível para permitir aquela distinção. Contudo, essa escolha deve considerar o aumento drástico no tempo de execução do algoritmo pelo que deverá ser estabelecido um compromisso entre discretização e um tempo aceitável de execução.

Esta nova abordagem dispensa, contudo, a escolha de células muito pequenas, uma vez que no caso de segmentos próximos não conseguirem incrementar células diferentes para serem distinguidos, pelo menos elas darão origem a diferentes sub-células que serão tratadas separadamente em fase posterior.

O fenómeno de espalhamento segundo ρ já referido atrás ficará também resolvido, assim como o facto da informação de cada segmento se encontrar distribuída por várias células, uma vez que o baixo limiar de selecção validará todas as sub-células (pertencentes ou não à mesma célula) com informação relevante.

A secção seguinte proporá um meio de tratar estas sub-células no sentido de reconstruir correctamente cada segmento e distinguir segmentos que incrementem a mesma célula.

5.8 Proposta de um Algoritmo de Pós-Processamento

Uma vez na posse de uma única lista ligada contendo todas as sub-células pré-seleccionadas, resta agora reconstruir cada segmento real através da fusão de pequenos segmentos que estão associados às referidas sub-células.

O procedimento de verificação é sequencial: começando com o primeiro elemento da lista, aplicam-se os critérios de fusão (a apresentar mais adiante) para juntá-lo com qualquer outro à sua direita. Sempre que se verifiquem as condições impostas por aqueles critérios, a informação de ambos é armazenada em apenas um deles sendo o outro imediatamente libertado de memória (Fig.26).

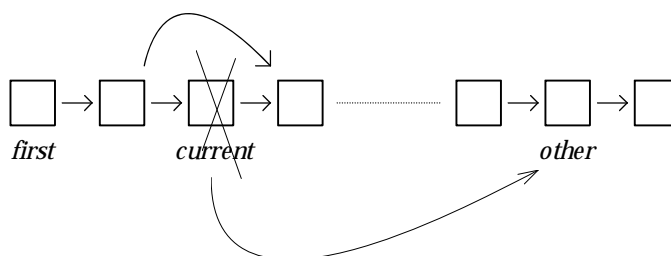


Fig. 26 - Fusão de sub-células na lista ligada.

A informação actualizada é o número de incrementos e as coordenadas dos pontos extremos do novo segmento. A tentativa de fusão prossegue então para a direita (para os restantes elementos da lista) até que o fim seja atingido. Considera-se depois o elemento seguinte, neste caso o segundo, e testam-se os critérios de fusão para todos os restantes elementos à sua direita (os elementos à esquerda já não são considerados); e assim sucessivamente até que o elemento a considerar seja o penúltimo que apenas tenta a fusão com o último. De referir ainda que numa fusão, o elemento a ser actualizado é o que se encontra mais à direita. Se fosse o da esquerda, este não seria mais objecto de novas tentativas de fusão o que poderia falhar possibilidades reais de fusão. É o caso da Fig.27 em que os segmentos parciais a , b e c , registados na lista nesta ordem, só serão todos fundidos se o elemento a receber a informação actualizada fôr o da direita.

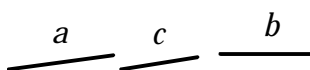


Fig. 27 - Partes de um segmento a serem fundidos.

Se assim não fosse, a fundiria com c (mas não com b devido à distância a que se encontra), recebendo a toda a informação; mas quando b fosse considerado não teria nenhum elemento à sua direita na lista com aquela informação com o qual pudesse fundir, uma vez que o elemento que continha c havia já sido removido!

Na solução adoptada, c receberá toda a informação actualizada, sendo a removido. Quando b é considerado, encontra c à sua direita na lista com o qual pode fundir.

O processo de fusão está sujeito a certas condições, que são consideradas pelo programa escrito na seguinte sequência:

- Colinearidade aproximada:

Os ângulos reais dos dois segmentos, que entretanto foram calculados, devem ser próximos; o valor limite de rejeição adoptado foi o da resolução de θ na tabela de acumulação: 30° . Se, esta condição for verificada, então o programa passa para o teste seguinte.

- O centro de um dos segmentos deve pertencer à linha definida pelo segundo (ou pelo menos estar próximo):

Os ângulos formados por dois segmentos a fundir podem ser aproximados, mas a sua distância física ser excessiva. A Fig.28 ilustra o processo de medir essa distância. Se os segmentos forem quase horizontais é testada a distância do centro de um deles à linha definida pelo outro, medida na vertical. Para segmentos quase verticais essa distância é medida na horizontal. Para uma distância maior que um dado valor, a fusão é rejeitada.

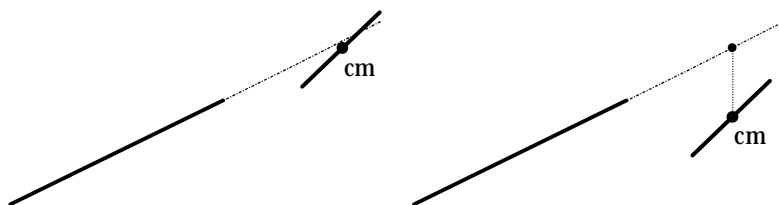


Fig. 28 - 2º teste de colinearidade dos segmentos: o centro de um deles (o mais pequeno) deve estar na vizinhança próxima da linha definida pelo outro.

- A separação entre os dois segmentos não deve exceder um determinado valor pré-definido:

Se apesar dos segmentos serem colineares e terem portanto passado os testes anteriores, eles se encontrarem demasiado afastados, então eles não pertencem ao mesmo segmento real e não devem portanto ser fundidos (Fig.29).

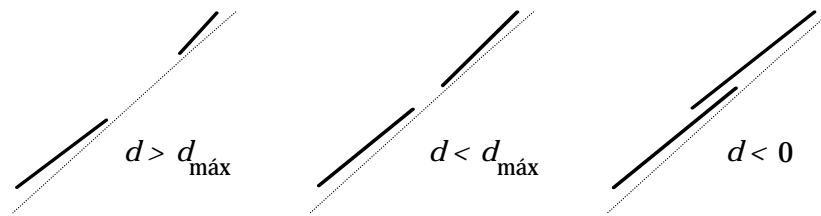


Fig. 29 - Verificação da distância entre dois segmentos colineares.

A distância é medida em unidades de pixel da imagem e utiliza as coordenadas dos pontos extremos de cada um dos segmentos. Para que o teste não utilize o ponto extremo errado, ele é efectuado para ambos, contemplando a possibilidade do ponto considerado inicial do segundo segmento se encontrar mais distante que o ponto considerado final.

5.9 O Software Desenvolvido

No sentido de implementar o que atrás foi descrito, foram escritos vários programas, que serão especificados e descritos no Anexo D.

Convém primeiramente, contudo, especificar o tipo de tabela de acumulação utilizada. Para a sua construção dividi o espaço de ρ em 60 intervalos enquanto que o de θ ficou dividido apenas em 9 intervalos. As gamas de valores dos dois espaços foram os seguintes:

- Valores de ρ :

Atendendo a que o ficheiro de imagem contém 640×512 pixeis, o mais distanciado do ponto $(0,0)$ situado no canto superior esquerdo, encontra-se à distância de 819,6 pixeis, pelo que convencionei para máximo o valor 800: $\rho \in [0, 800]$; a resolução é portanto de 13,3 pixeis por intervalo.

- Valores de θ :

Tendo em conta os valores possíveis de θ através do resultado da aplicação das duas máscaras h e v , decidi atribuir àquela grandeza os seguintes valores limites: $\theta \in [-\pi/2, \pi]$; tendo em conta a divisão desta gama de valores em 9 intervalos, obtém-se uma resolução de $4,71 \text{ rad} = 30^\circ$.

No Anexo D.1 é apresentado o esquema de chamadas para as várias funções envolvidas neste procedimento.

No Anexo D.2 é feita uma descrição breve de cada uma daquelas funções.

5.10 Resultados Experimentais

Para testar este conjunto de algoritmos, foram utilizados objectos poliédricos com formas e dimensões variadas. A escolha daquele tipo de objectos deveu-se à abundância de segmentos de fronteira visíveis cuja detecção é o alvo desses algoritmos. Apresento de seguida os resultados obtidos para dois conjuntos de objectos.

5.10.1 Imagem nº 1

Nesta configuração, pretendo efectuar 3 testes distintos: (1) detecção de segmentos pequenos, (2) discriminação de segmentos colineares pertencentes a objectos diferentes e (3) efeito de excessiva proximidade entre dois objectos.

As Figs. 32, 33, 34, 35 e 36 apresentam sucessivamente a imagem binarizada, o resultado da detecção de pontos de fronteira, os segmentos seleccionados da tabela de acumulação, o resultado da sua fusão e o resultado final após a eliminação de pequenos segmentos remanescentes.

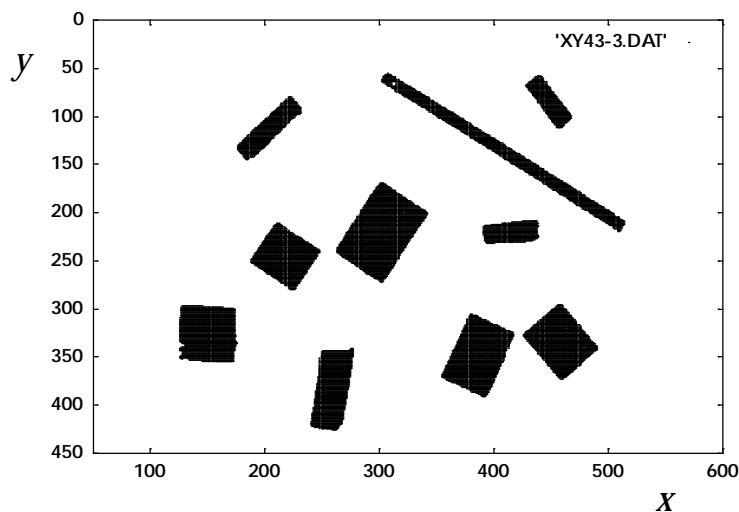


Fig. 30 - Imagem binarizada do conjunto de objectos.

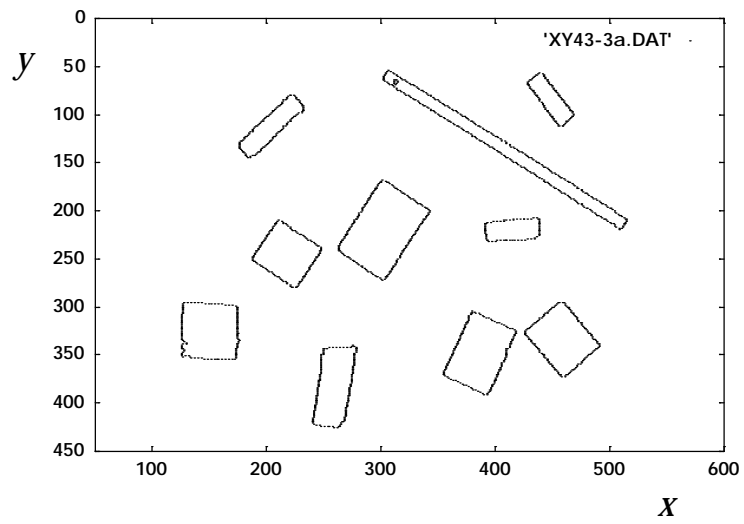


Fig. 31 - Pontos de fronteira sujeitos à Transformada de Hough.

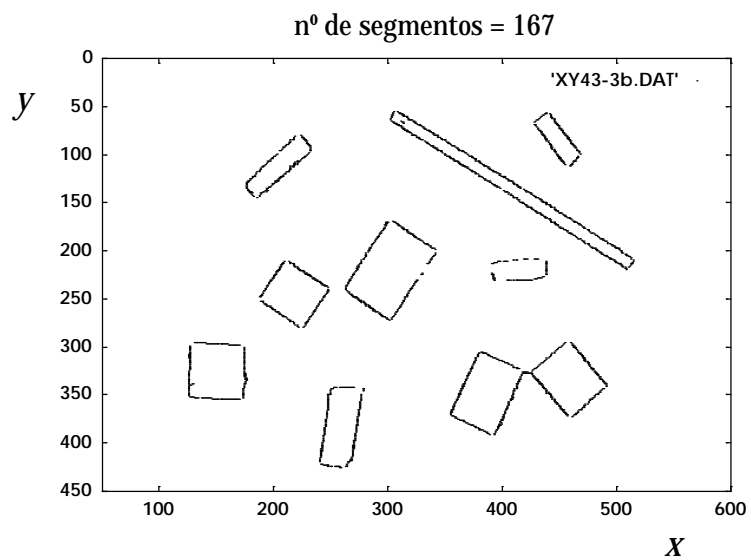


Fig. 32 - Segmentos seleccionados da tabela de acumulação.

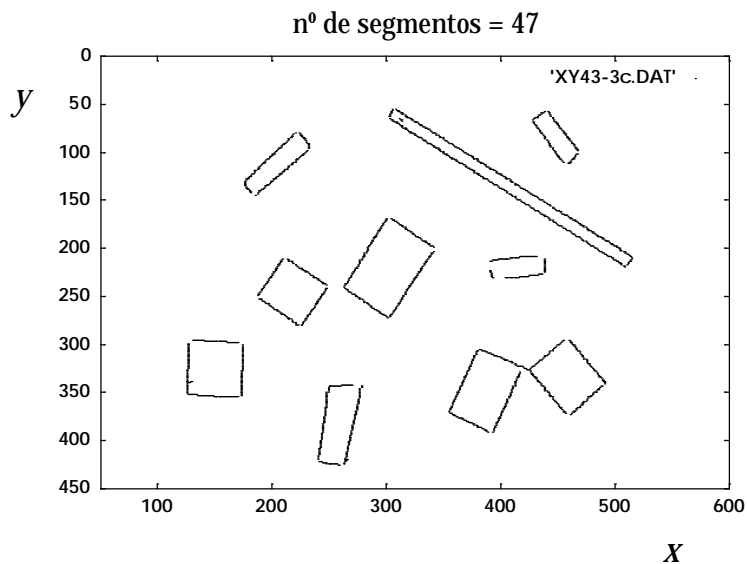


Fig. 33 - Segmentos resultantes do algoritmo de fusão.

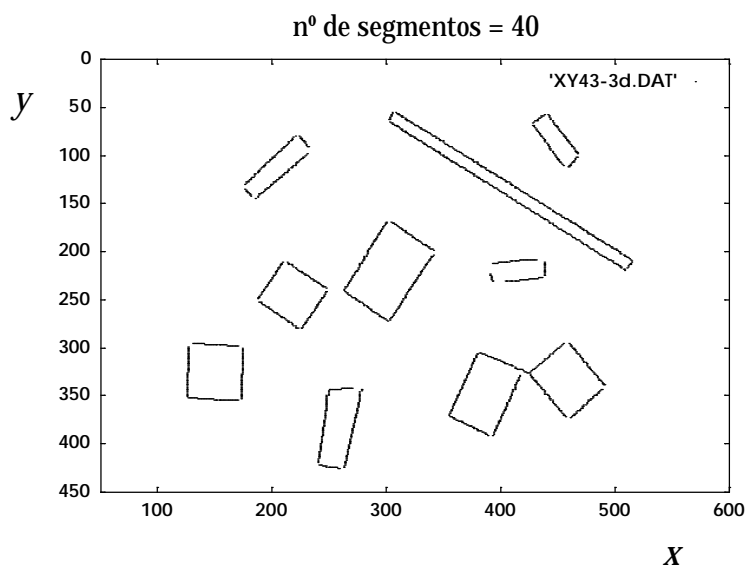


Fig. 34 - Imagem final dos segmentos reconstruídos após eliminação dos segmentos pequenos.

(1) Como se pode observar, o objecto alongado no canto superior direito exhibe os seus dois segmentos menores o que comprova a capacidade deste algoritmo de detectar segmentos pequenos.

(2) Dois dos objectos centrais apresentam arestas colineares, que na abordagem de [Duda77] originaria a sua fusão automática. Nesta abordagem eles são correctamente discriminados.

(3) No caso dos dois objectos do canto inferior direito, a sua proximidade é tal que os segmentos são interligados. A solução para ultrapassar esta inconveniência é diminuir o valor definido para distância máxima a que duas porções de segmento se podem encontrar para se considerarem como pertencentes ao mesmo segmento. A desvantagem neste procedimento está na possibilidade de várias partes de um mesmo segmento ligeiramente afastados não conseguirem fundir.

5.10.2 Imagem nº 2

Neste caso, dada a grande proximidade dos objectos, houve necessidade de utilizar as máscaras de menor dimensão: 3×3 .

Mais uma vez, as Figs. 37, 38, 39, 40 e 41 apresentam sucessivamente a imagem binarizada, o resultado da detecção de pontos de fronteira, os segmentos seleccionados da tabela de acumulação, o resultado da sua fusão e o resultado final após a eliminação de pequenos segmentos remanescentes.

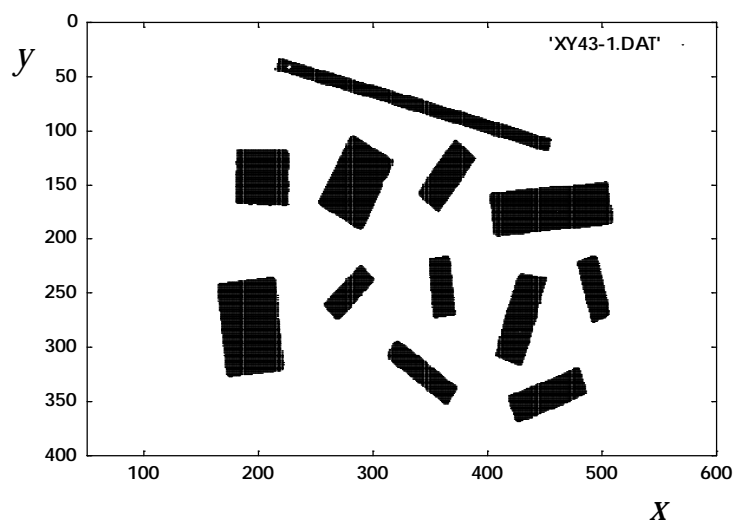


Fig. 35 - Imagem binarizada do conjunto de objectos.

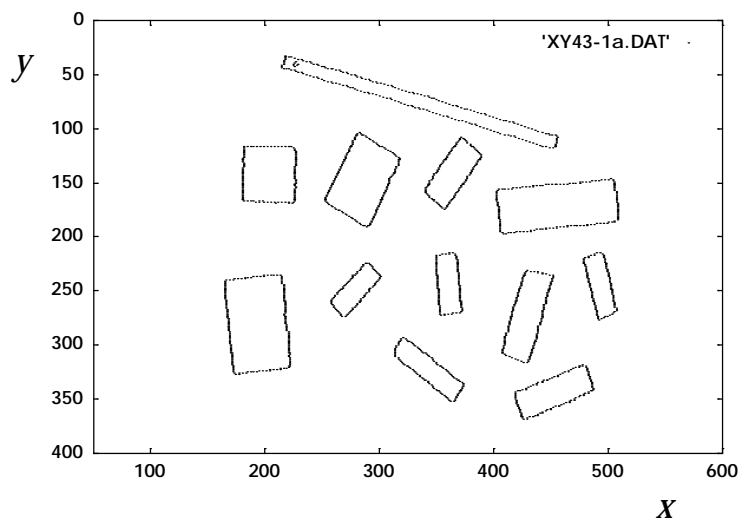


Fig. 36 - Pontos de fronteira sujeitos à Transformada de Hough.

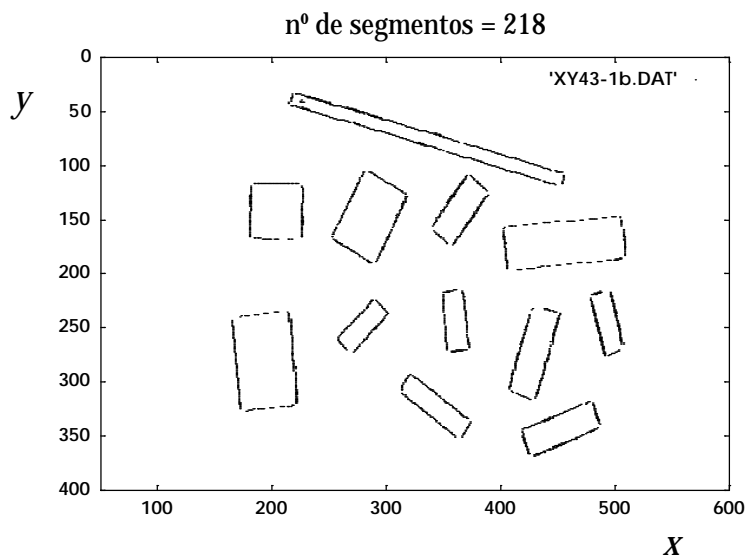


Fig. 37 - Segmentos seleccionados da tabela de acumulação.

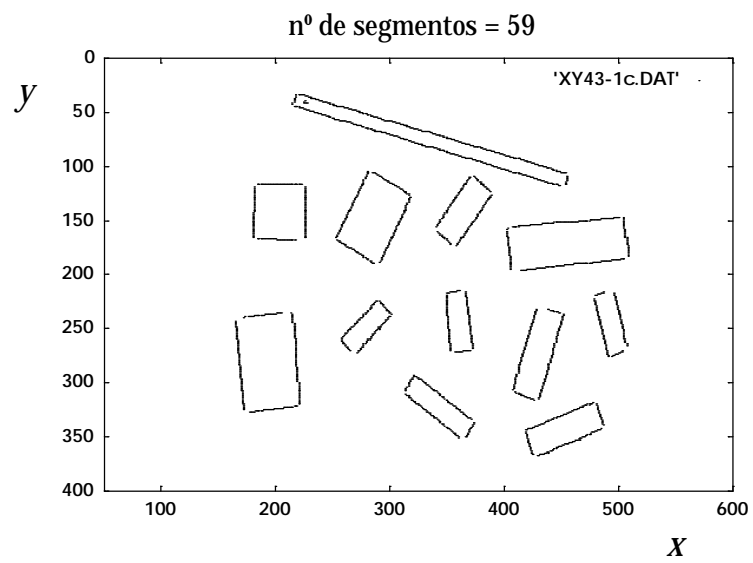


Fig. 38 - Segmentos resultantes do algoritmo de fusão.

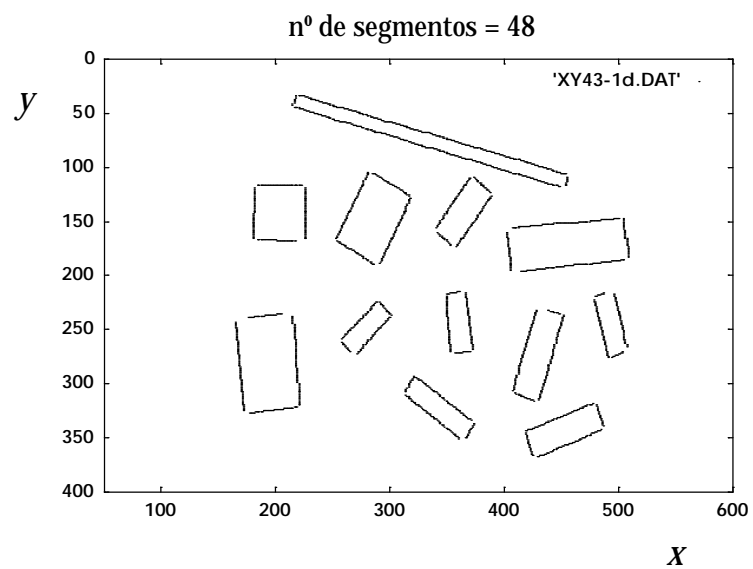


Fig. 39 - Imagem final dos segmentos reconstruídos após eliminação dos segmentos pequenos.

Segment - 1- : Segm(254,168) - (287,190) , angle= 29°, length= 37)
 Segment - 2- : Segm(438,232) - (449,235) , angle= 12°, length= 11)
 Segment - 3- : Segm(216, 44) - (449,118) , angle= 14°, length=240)
 Segment - 4- : Segm(409,309) - (426,316) , angle= 18°, length= 17)
 Segment - 5- : Segm(220, 33) - (454,106) , angle= 14°, length=241)
 Segment - 6- : Segm(258,262) - (265,270) , angle= 43°, length= 9)
 Segment - 7- : Segm(184,117) - (223,117) , angle= 0°, length= 39)
 Segment - 8- : Segm(341,160) - (354,172) , angle= 37°, length= 16)
 Segment - 9- : Segm(323,294) - (372,335) , angle= 34°, length= 59)
 Segment -10- : Segm(285,105) - (317,126) , angle= 28°, length= 36)
 Segment -11- : Segm(407,156) - (500,147) , angle= -4°, length= 93)
 Segment -12- : Segm(314,311) - (361,351) , angle= 35°, length= 57)
 Segment -13- : Segm(165,243) - (172,325) , angle= 84°, length= 68)
 Segment -14- : Segm(184,167) - (217,168) , angle= 1°, length= 33)
 Segment -15- : Segm(373,109) - (389,124) , angle= 37°, length= 20)
 Segment -16- : Segm(291,225) - (301,237) , angle= 44°, length= 14)
 Segment -17- : Segm(217, 34) - (215, 42) , angle=-73°, length= 6)
 Segment -18- : Segm(181,119) - (180,165) , angle=-88°, length= 38)
 Segment -19- : Segm(176,239) - (205,236) , angle= -4°, length= 29)
 Segment -20- : Segm(226,119) - (226,166) , angle= 90°, length= 39)
 Segment -21- : Segm(281,104) - (253,166) , angle=-61°, length= 58)
 Segment -22- : Segm(350,217) - (361,215) , angle= -8°, length= 11)
 Segment -23- : Segm(318,128) - (291,189) , angle=-61°, length= 57)
 Segment -24- : Segm(215,238) - (221,320) , angle= 84°, length= 68)
 Segment -25- : Segm(349,219) - (353,270) , angle= 84°, length= 42)
 Segment -26- : Segm(366,216) - (372,268) , angle= 82°, length= 43)
 Segment -27- : Segm(289,224) - (258,258) , angle=-42°, length= 41)
 Segment -28- : Segm(370,109) - (340,156) , angle=-52°, length= 49)
 Segment -29- : Segm(300,238) - (271,272) , angle=-44°, length= 40)
 Segment -30- : Segm(177,327) - (219,322) , angle= -5°, length= 42)
 Segment -31- : Segm(415,196) - (506,185) , angle= -5°, length= 91)
 Segment -32- : Segm(389,126) - (358,173) , angle=-51°, length= 49)
 Segment -33- : Segm(402,160) - (406,195) , angle= 82°, length= 29)
 Segment -34- : Segm(480,219) - (488,216) , angle=-17°, length= 8)
 Segment -35- : Segm(357,271) - (370,270) , angle= -3°, length= 13)
 Segment -36- : Segm(419,346) - (425,368) , angle= 71°, length= 19)
 Segment -37- : Segm(504,270) - (494,276) , angle=-26°, length= 11)
 Segment -38- : Segm(321,293) - (313,308) , angle=-57°, length= 14)
 Segment -39- : Segm(455,109) - (454,117) , angle=-81°, length= 6)
 Segment -40- : Segm(479,222) - (491,276) , angle= 75°, length= 46)
 Segment -41- : Segm(481,320) - (487,340) , angle= 70°, length= 17)
 Segment -42- : Segm(373,337) - (366,351) , angle=-58°, length= 13)
 Segment -43- : Segm(496,217) - (507,266) , angle= 74°, length= 42)
 Segment -44- : Segm(428,232) - (407,307) , angle=-71°, length= 65)
 Segment -45- : Segm(419,344) - (478,318) , angle=-20°, length= 62)
 Segment -46- : Segm(504,147) - (509,183) , angle= 80°, length= 30)
 Segment -47- : Segm(452,237) - (430,315) , angle=-71°, length= 68)
 Segment -48- : Segm(429,368) - (485,343) , angle=-20°, length= 59)

Fig. 40 - Registo textual da lista final dos segmentos.

5.11 Conclusão e Trabalho Futuro

Pela análise de todos os resultados obtidos, dos quais apenas alguns foram apresentados aqui, pode-se concluir que se conseguiu ultrapassar com grande sucesso as dificuldades e impossibilidades que as anteriores abordagens apresentavam. Com este novo algoritmo é já possível reconstruir correctamente segmentos de qualquer comprimento (desde, claro está, que não se trate de segmentos de comprimento comparável ao do pixel), assim como distinguir devidamente pares de segmentos colineares ou muito próximos. Sendo assim, ele pode ser utilizado como potencial identificador de objectos que na sua fronteira contenham segmentos.

Futuramente este algoritmo poderá ser estendido a outras formas geométricas com especial destaque para os arcos de circunferência e de elipse, desde que sejam escolhidos outros parâmetros que os definam devidamente. Embora vários outros desenvolvimentos e adaptações tenham já sido feitos com algum sucesso, a completa compreensão das propriedades e limitações ainda não foi atingida, como é bem observado em [Maitre86]. Algumas tentativas de as prever podem ser encontradas em [Sklansky78] [Shapiro79] [VanVeen81] [Brown83] e [Svalbe89]. Apesar disso, A Transformada de Hough continua a ser objecto de estudos [Yuen89] [Grimson90] [Corazza91], e resultados tais como os apresentados nesta Dissertação continuam a aparecer continuamente nas publicações científicas da especialidade.

6.1 Introdução

No capítulo anterior demonstrou-se que é possível detectar, medir e parametrizar com grande eficiência, todos os segmentos presentes numa imagem e pertencentes às fronteiras dos objectos. Esse resultado, contudo, ainda não permite a caracterização completa daqueles objectos. Para tal será ainda necessário agrupá-los e sequenciá-los. Uma vez executadas estas duas tarefas será possível então calcular os ângulos internos e estabelecer sequências de valores comprimento-ângulo que caracterizem univocamente cada um dos objectos presentes.

Aquelas sequências poderão então ser comparadas com as que se encontrem registadas numa base de dados para se obter as respectivas identidades.

Este desenvolvimento encontra-se documentado em [Lopes93a] e [Lopes93b].

6.2 O Agrupamento dos Segmentos

Dada a lista final com a informação de todos os segmentos presentes na imagem, o algoritmo inicia a sua acção considerando o 1º elemento daquela lista e inserindo-o numa outra com diferente estrutura. Nessa nova estrutura foi dispensada informação agora desnecessária nomeadamente o número de incrementos associado ao segmento, a ordem de registo na lista e a posição na tabela de acumulação (esta última já sem significado para a própria lista de segmentos uma vez que estes podem ter resultado da fusão de sub-células provenientes de mais do que uma célula da tabela de acumulação). Quanto ao comprimento l , ele será substituído por um valor mais próximo do real após medida da distância entre vértices. Finalmente, o valor do ângulo, embora essencial para estabelecer a relação angular entre segmentos, não será transferido para a nova estrutura. O novo valor *angle* conterà então o ângulo que o segmento faz com o seguinte.

Considerando um dos extremos do segmento acabado de inserir na nova lista, efectua-se uma procura sequencial na lista antiga de um outro segmento que com ele possa ligar. Essa ligação é estabelecida quando uma das extremidades do 2º segmento

está próximo do fim do 1º segmento, ou seja, a uma distância menor que um dado valor pré-definido ($1,5 \times max_dist$, com max_dist = distância máxima usada para efectuar a fusão dos segmentos seleccionados do capítulo anterior) (Fig.1). Essa verificação é efectuada para ambas as extremidades do 2º segmento. Sempre que uma ligação é estabelecida, o 2º segmento é inserido na nova lista e as novas coordenadas ($lastX, lastY$) do fim da cadeia são actualizadas (correspondem a (X_i, Y_i) ou a (X_f, Y_f) consoante o 2º segmento tiver sido invertido ou não) (Fig.2).

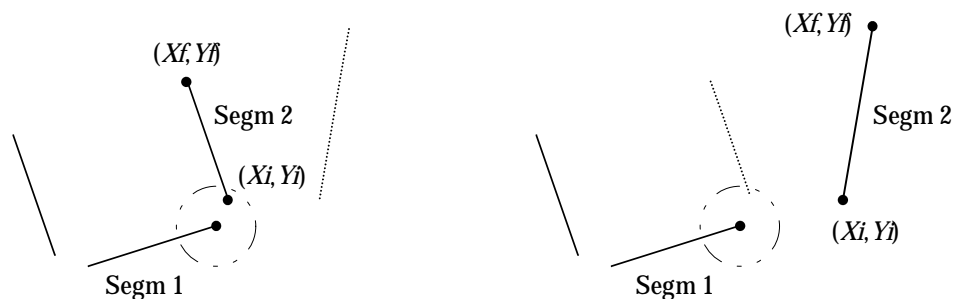


Fig. 1 - Verificação da proximidade do 2º segmento: sucesso e rejeição.

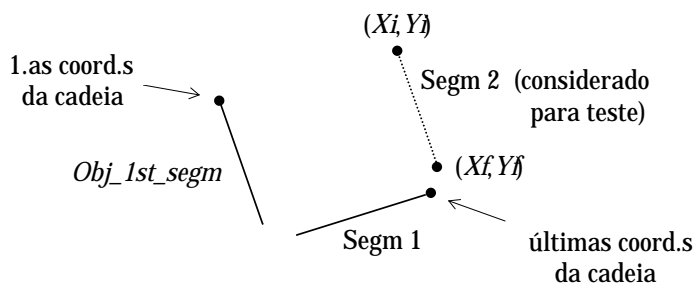


Fig. 2 - Actualização das coordenadas do fim da cadeia de segmentos com o objectivo de verificar novas proximidades.

Se o 2º segmento tiver sido invertido, as coordenadas iniciais e finais são trocadas quando da respectiva transferência para a nova lista, e o ângulo de inclinação corrigido. Dessa forma, a diferença entre os novos ângulos de inclinação dará sempre o valor do ângulo interno: $0 < angle < \pi$. Sempre que haja uma inserção na nova lista é feita a verificação de objecto fechado, ou seja, verificada a proximidade entre as coordenadas finais actualizadas da cadeia, ($lastX, lastY$), e as iniciais, ($firstX, firstY$). Se se verificar essa proximidade, então é colocada a indicação respectiva no campo *closed* do último segmento inserido, $closed=1$, e tomado outro segmento da lista antiga para efectuar novos encadeamentos (deverá ser o 1º segmento não ainda não transferido).

Se tal não acontecer, é reiniciada a procura de novo segmento próximo a partir do início da lista antiga. Os segmentos já inseridos já não são alvo de verificação e para que tal não aconteça é-lhes colocada uma marca: utiliza-se o campo *order* já sem utilidade e regista-se-lhe o valor 255. Este procedimento de procura repete-se sempre que um novo segmento é encontrado e inserido e termina quando o objecto fecha ou

quando é atingido o último segmento da lista. Nesta última circunstância o objecto não fecha pelo que é mantido o valor $closed=0$ em todos os elementos da cadeia e em particular no último considerado. É ainda mantido um contador actualizado de objectos encontrados, sejam eles fechados ou não, que é utilizado para distinguir os segmentos de cada objecto. Essa associação a cada objecto é conseguida pelo registo do nº daquele a que pertencem, na nova estrutura associada a cada segmento.

Obtém-se assim uma nova lista de segmentos encadeados com indicação do objecto a que pertencem e do estado fechado ou aberto de cada cadeia.

Para se poder ter uma visão global do modo de funcionamento deste algoritmo, é apresentada no Anexo E.1 uma descrição esquematizada.

6.3 A Extrapolação dos Segmentos

Os segmentos finais obtidos no capítulo anterior, como se pode observar nas respectivas figuras, embora correspondam aos da imagem original, não possuem o seu verdadeiro comprimento pelo que os vértices não ficam marcados. Tal deve-se à dimensão finita das máscaras aplicadas a cada ponto de fronteira: quando este se encontra próximo de um vértice, os pixels pertencentes ao outro segmento interferem na correcta avaliação das componentes do gradiente h e v . Como consequência, o algoritmo falha a célula correcta: são calculados valores errados para (ρ, θ) dando origem a um incremento numa célula errada da tabela de acumulação.

Há, portanto, que medir o valor correcto dos segmentos para que o processo de identificação seja eficiente.

Uma vez que os segmentos se encontram agora em sequência, torna-se fácil a determinação dos vértices associados a cada par (Fig.3).

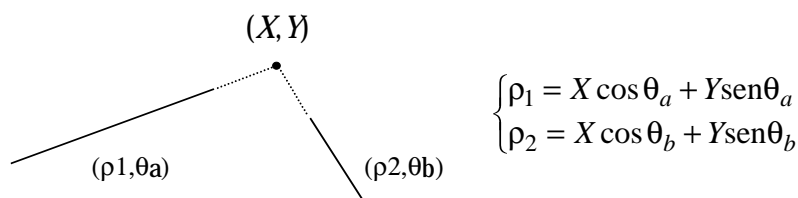


Fig. 3 - Determinação dos vértices.

Onde θ_a e θ_b dizem respeito a cada uma das rectas definidas pelos segmentos de acordo com a equação geral em coordenadas polares:

$$\rho = X \cos \theta + Y \sin \theta$$

onde θ é o ângulo que o vector posição do ponto da recta mais próximo da origem faz com o eixo dos xx e ρ o seu comprimento.

Conhecido o ângulo que a recta faz com o semi-eixo positivo dos xx , pode-se reescrever aquela equação. Como se verá adiante, existem 3 situações distintas de que resultam outras tantas equações. Considerando que o ângulo formado pelas rectas a intersectar não é pequeno (se se tratar de quadrados ou rectângulos ele é de 90°) vai existir um número de situações de intersecção igual ao de combinações de 3, 2 a 2:

$${}^3C_2 = \frac{3!}{2!} = \frac{6}{2} = 3$$

Como irei demonstrar, todas elas levam ao mesmo resultado.

Analisemos primeiro a alteração a introduzir às equações nas 3 situações, consoante a posição do segmento na imagem.

- 1ª situação ($0 < \theta < \pi/2$):

De acordo com a Fig.4a, a relação entre θ e θ_1 deverá ser: $\theta_1 = \theta + \pi/2$. Contudo, e atendendo a que para o cálculo de θ_1 é utilizada uma função que só dá valores no intervalo $[-\pi/2, \pi/2]$, a relação pode ser alterada para $\theta_1 = \theta - \pi/2$ ou ainda $\theta = \theta_1 + \pi/2$ (Fig.4b).

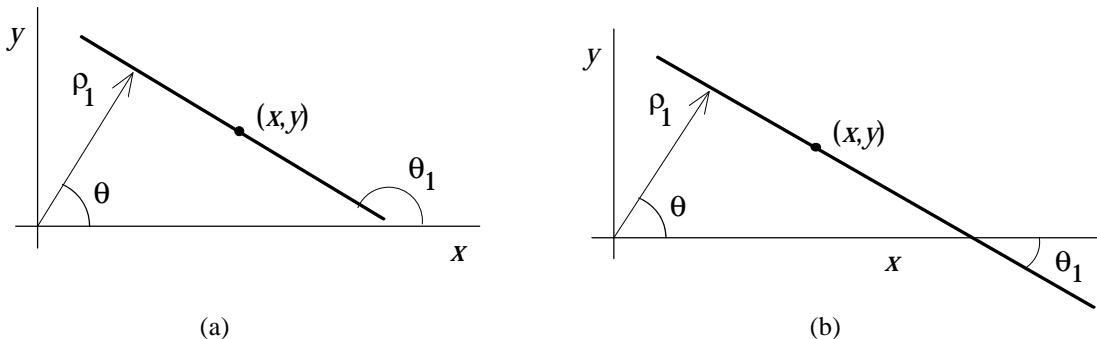


Fig. 4 - Determinação de θ em função de θ_1 .

Então, tendo em conta a relação acima, tem-se que,

$$\cos(\theta) = -\text{sen}(\theta_1) \quad \text{e} \quad \text{sen}(\theta) = \cos(\theta_1)$$

pelo que a equação geral da recta tomará a forma seguinte:

$$\rho_1 = -X \text{sen} \theta_1 + Y \cos \theta_1$$

- 2ª situação ($\pi/2 < \theta < \pi$):

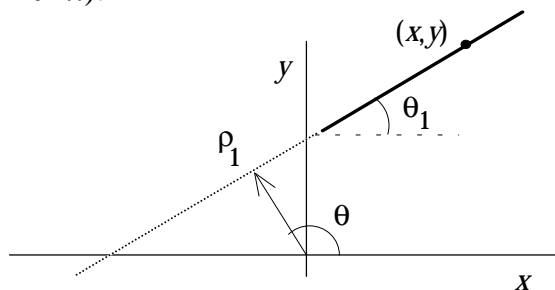


Fig. 5 - Determinação de θ em função de θ_1 .

De acordo com a Fig.5, a relação entre θ e θ_1 é imediata: $\theta = \theta_1 + \pi/2$, que é igual à obtida anteriormente, pelo que a forma da recta será também igual:

$$\rho_1 = -X \operatorname{sen} \theta_1 + Y \cos \theta_1$$

- 3ª situação ($3\pi/2 < \theta < 2\pi$):

Uma vez que não podem existir segmentos no 3º quadrante ($\pi < \theta < 3\pi/2$), resta esta 3ª situação (Fig.6).

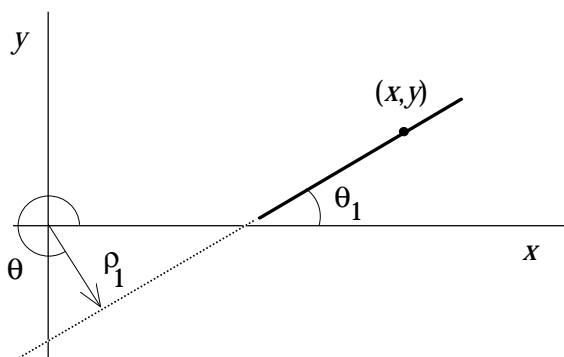


Fig. 6 - Determinação de θ em função de θ_1 .

A relação entre θ e θ_1 é a seguinte: $\theta = \theta_1 + 3\pi/2$. Sendo assim,

$$\cos(\theta) = \operatorname{sen}(\theta_1) \quad \text{e} \quad \operatorname{sen}(\theta) = -\cos(\theta_1)$$

O que substituindo na equação geral dá,

$$\rho_1 = X \operatorname{sen} \theta_1 - Y \cos \theta_1$$

Tendo em conta aquelas 3 equações consideram-se agora as 3 combinações possíveis.

- 1ª situação de intersecção: 1 e 2 ($0 < \theta_1 < \pi/2 \wedge \pi/2 < \theta_2 < \pi$)

As equações são as seguintes:

$$\begin{cases} \rho_1 = -X \operatorname{sen} \theta_1 + Y \cos \theta_1 \\ \rho_2 = -X \operatorname{sen} \theta_2 + Y \cos \theta_2 \end{cases}$$

Resolvendo aquelas equações para X e Y obtém-se,

$$\begin{cases} y = \frac{\rho_2 \operatorname{sen} \theta_1 - \rho_1 \operatorname{sen} \theta_2}{\operatorname{sen} \theta_1 \cos \theta_2 - \cos \theta_1 \operatorname{sen} \theta_2} = \frac{\rho_2 \operatorname{sen} \theta_1 - \rho_1 \operatorname{sen} \theta_2}{\operatorname{sen}(\theta_1 - \theta_2)} & (\operatorname{sen} \theta_1 > 0,4) \\ x = \frac{y \cos \theta_1 - \rho_1}{\operatorname{sen} \theta_1} \end{cases}$$

$$\vee \begin{cases} x = \frac{\rho_2 \cos \theta_1 - \rho_1 \cos \theta_2}{\operatorname{sen} \theta_1 \cos \theta_2 - \cos \theta_1 \operatorname{sen} \theta_2} = \frac{\rho_2 \cos \theta_1 - \rho_1 \cos \theta_2}{\operatorname{sen}(\theta_1 - \theta_2)} & (\operatorname{sen} \theta_1 < 0,4) \\ y = \frac{x \operatorname{sen} \theta_1 + \rho_1}{\cos \theta_1} \end{cases}$$

O 1º conjunto destina-se a segmentos próximos da vertical (o limiar de 0,4 é puramente convencional) enquanto que o 2º se destina aos restantes. O objectivo desta separação é evitar divisões por números próximos de 0.

- 2ª situação de intersecção: 1 e 3 ($0 < \theta_1 < \pi/2 \wedge 3\pi/2 < \theta_2 < 2\pi$):

As equações são as seguintes:

$$\begin{cases} \rho_1 = -X \operatorname{sen} \theta_1 + Y \cos \theta_1 \\ \rho_2 = X \operatorname{sen} \theta_2 - Y \cos \theta_2 \end{cases}$$

Resolvendo as equações em ordem a X e Y obtém-se o seguinte:

$$\begin{cases} y = \frac{\rho_2 \operatorname{sen} \theta_1 + \rho_1 \operatorname{sen} \theta_2}{-\operatorname{sen} \theta_1 \cos \theta_2 + \cos \theta_1 \operatorname{sen} \theta_2} = -\frac{\rho_2 \operatorname{sen} \theta_1 + \rho_1 \operatorname{sen} \theta_2}{\operatorname{sen}(\theta_1 - \theta_2)} & (\operatorname{sen} \theta_1 > 0,4) \\ x = \frac{y \cos \theta_1 - \rho_1}{\operatorname{sen} \theta_1} \end{cases}$$

$$\vee \begin{cases} x = \frac{\rho_2 \cos \theta_1 + \rho_1 \cos \theta_2}{-\operatorname{sen} \theta_1 \cos \theta_2 + \cos \theta_1 \operatorname{sen} \theta_2} = -\frac{\rho_2 \cos \theta_1 + \rho_1 \cos \theta_2}{\operatorname{sen}(\theta_1 - \theta_2)} & (\operatorname{sen} \theta_1 < 0,4) \\ y = \frac{x \operatorname{sen} \theta_1 + \rho_1}{\cos \theta_1} \end{cases}$$

- 3ª situação de intersecção: 2 e 3 ($\pi/2 < \theta_1 < \pi \wedge 3\pi/2 < \theta_2 < 2\pi$)

Uma vez que não podem existir segmentos no 3º quadrante ($\pi < \theta < 3\pi/2$), resta esta 3ª situação.

As equações são novamente as seguintes:

$$\begin{cases} \rho_1 = -X \operatorname{sen} \theta_1 + Y \operatorname{cos} \theta_1 \\ \rho_2 = X \operatorname{sen} \theta_2 - Y \operatorname{cos} \theta_2 \end{cases}$$

O resultado para (X, Y) deverá então ser também o mesmo. Esta situação deverá, contudo, ser rejeitada uma vez que vai corresponder a segmentos quase colineares (Figs. 6 e 7) restando apenas a 1ª e 2ª situações.

Aparentemente teremos que considerar essas 2 situações distintas para cada vértice a determinar. Contudo, se ρ_1 e ρ_2 forem substituídos pelas respectivas expressões, chega-se ao mesmo resultado! Note-se que o que difere as expressões da 2ª situação das da 1ª é o sinal que afecta ρ_2 . Ora ρ_2 é precisamente simétrico ao considerado anteriormente, pelo que as equações se reduzem todas a qualquer um dos conjuntos!

Então, já de posse das fórmulas que permitem o cálculo das coordenadas de um vértice, procede-se à sua determinação tomando sequencialmente na cadeia, pares de segmentos ligados. Para cada um deles são actualizadas as coordenadas de um dos pontos extremos: o final para o 1º segmento e o inicial para o 2º. Sempre que se atinge o fim de uma cadeia, é verificado o seu estado aberto ou fechado; no 2º caso, para além do 1º segmento do par (que corresponde ao último da cadeia) é também actualizado o ponto inicial do 2º (que corresponde ao 1º da cadeia). São também recalculados quer os ângulos de inclinação quer os comprimentos dos segmentos. O fluxograma apresentado no Anexo E.2 ilustra estes procedimentos.

6.4 Resultados Experimentais

6.4.1 1º Conjunto de Objectos

Uma das imagens sujeita ao teste de sucesso na extrapolação é aquela que apresento a seguir (Fig.7):

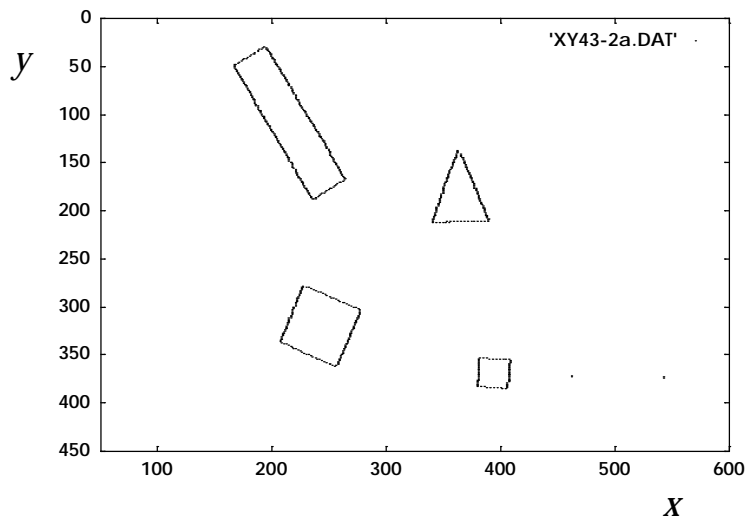


Fig. 7 - Pontos de fronteira pertencentes a 4 objectos de teste.

A esta imagem vai corresponder, após a análise da transformada de Hough modificada, a selecção de sub-células e a fusão dos segmentos respectivos, o seguinte resultado (Fig.8):

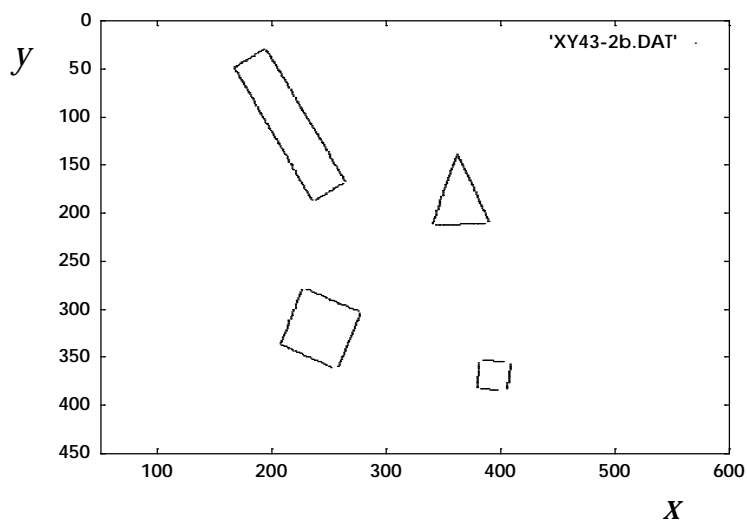


Fig. 8 - Segmentos após o algoritmo de fusão.

A descrição dos segmentos, que deve ser comparada com a obtida no final, é a seguinte:

Segments.txt	(segmentos finais após a fusão)
Segment - 1- :	Segm((167, 48) - (193, 29) , angle=-31, length= 30)
Segment - 2- :	Segm((230,279) - (275,302) , angle= 23, length= 48)
Segment - 3- :	Segm((167, 49) - (235,187) , angle= 59, length=133)


```

Segment - 4- : Segm( (208,337) - (253,361) , angle= 23, length= 49 )
Segment - 5- : Segm( (195, 30) - (264,167) , angle= 58, length=133 )
Segment - 6- : Segm( (238,186) - (262,169) , angle=-30, length= 27 )
Segment - 7- : Segm( (226,280) - (207,336) , angle=112, length= 50 )
Segment - 8- : Segm( (343,212) - (386,211) , angle= -1, length= 43 )
Segment - 9- : Segm( (385,353) - (402,354) , angle= 2, length= 17 )
Segment -10- : Segm( (362,139) - (390,209) , angle= 64, length= 64 )
Segment -11- : Segm( (383,383) - (397,384) , angle= 3, length= 14 )
Segment -12- : Segm( (361,141) - (340,211) , angle=109, length= 61 )
Segment -13- : Segm( (277,304) - (258,360) , angle=112, length= 50 )
Segment -14- : Segm( (381,356) - (380,382) , angle=92, length= 21 )
Segment -15- : Segm( (408,357) - (406,383) , angle=95, length= 21 )
    
```

Com esta lista pode-se numerar e atribuir orientação aos segmentos na imagem (Fig.9):

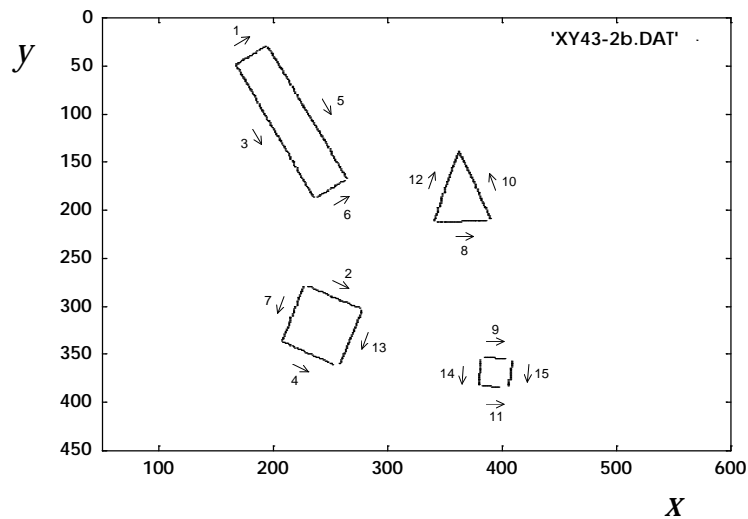


Fig. 9 - Segmentos após o algoritmo de fusão, numerados e orientados.

O seu agrupamento por objecto dá o seguinte resultado:

Chain.tx1 (antes de *Extra.c*)

```

----- group -1- -----
obj_nr = 1, i(167, 48) -- f(193, 29)
angle = -31, length = 30, closed=0
1
obj_nr = 1, i(195, 30) -- f(264,167)
angle = 58, length = 133, closed=0
5
obj_nr = 1, i(262,169) -- f(238,186)
angle = 150, length = 27, closed=0
~6
    
```

```

obj_nr = 1, i(235,187) -- f(167, 49)
angle = -121, length = 133, closed=1
----- Object -1- ----- ~3

----- group -2- -----
obj_nr = 2, i(230,279) -- f(275,302)
angle = 23, length = 48, closed=0
2
obj_nr = 2, i(277,304) -- f(258,360)
angle = 112, length = 50, closed=0
13
    
```

```

obj_nr = 2, i(253,361) -- f(208,337)
angle = -157, length = 49, closed=0
                                ~4
obj_nr = 2, i(207,336) -- f(226,280)
angle = -68, length = 50, closed=1
----- Object -2- ----- ~7

----- group -3- -----
obj_nr = 3, i(343,212) -- f(386,211)
angle = -1, length = 43, closed=0
                                8
obj_nr = 3, i(390,209) -- f(362,139)
angle = -116, length = 64, closed=0
                                ~10
obj_nr = 3, i(361,141) -- f(340,211)
angle = 109, length = 61, closed=1
    
```

```

----- Object -3- ----- 12

----- group -4- -----
obj_nr = 4, i(385,353) -- f(402,354)
angle = 2, length = 17, closed=0
                                9
obj_nr = 4, i(408,357) -- f(406,383)
angle = 95, length = 21, closed=0
                                15
obj_nr = 4, i(397,384) -- f(383,383)
angle = -177, length = 14, closed=0
                                ~11
obj_nr = 4, i(380,382) -- f(381,356)
angle = -88, length = 21, closed=1
----- Object -4- ----- ~14
    
```

Os algarismos em "bold" correspondem ao número de ordem dos segmentos na 1ª lista (quando precedidos de til, significa que foram invertidos antes de serem inseridos na cadeia). As 4 cadeias obtidas estão de acordo com a numeração e orientação mostradas na Fig.9:

1. **1 - 5 - ~6 - ~3**
2. **2 - 13 - ~4 - ~7**
3. **8 - ~10 - 12**
4. **9 - 15 - ~11 - ~14**

Seguindo a sequência dos segmentos apresentados com a respectiva nova orientação, consegue-se de facto contornar totalmente os objectos, obtendo-se assim uma codificação sequencial. Se forem calculados os valores correctos dos comprimentos e ângulos internos obtêm-se finalmente as sequências comprimento_ângulo que irão caracterizar os objectos. Cada ângulo interno é obtido pelo complementar para 180° da diferença entre o ângulo do segmento seguinte e o do que se está a considerar. Uma vez que a nova função que se utiliza ($atan2(dy,dx)$) só fornece valores entre $-\pi$ e $+\pi$, então o ângulo interno deverá ser dado pelo valor absoluto daquela diferença.

A lista apresentada a seguir corresponde ao mesmo agrupamento de segmentos após a aplicação do algoritmo de extrapolação (notar a ligeira alteração nas coordenadas dos pontos extremos que vão agora corresponder aos vértices encontrados assim como no comprimento dos segmentos).

Chain.tx2 (depois de *Extra.c*)

```

----- group -1- -----
obj_nr = 1, i(166, 48) -- f(194, 28)
angle = -30, length = 32, closed=0
                                1
obj_nr = 1, i(194, 28) -- f(264,167)
angle = 58, length = 135, closed=0
                                5
obj_nr = 1, i(264,167) -- f(235,187)
angle = 150, length = 33, closed=0
    
```

```

                                ~6
obj_nr = 1, i(235,187) -- f(166, 48)
angle = -120, length = 134, closed=1
----- Object -1- ----- ~3

----- group -2- -----
obj_nr = 2, i(226,277) -- f(277,303)
angle = 22, length = 55, closed=0
                                2
    
```

```

obj_nr = 2, i(277,303) -- f(256,363)
angle = 112, length = 54, closed=0
13
obj_nr = 2, i(256,363) -- f(206,336)
angle = -155, length = 54, closed=0
~4
obj_nr = 2, i(206,336) -- f(226,277)
angle = -67, length = 52, closed=1
----- Object -2- ----- ~7

----- group -3- -----
obj_nr = 3, i(339,212) -- f(390,210)
angle = -1, length = 51, closed=0
8
obj_nr = 3, i(390,210) -- f(361,138)
angle = -115, length = 66, closed=0
~10
obj_nr = 3, i(361,138) -- f(339,212)

```

```

angle = 109, length = 65, closed=1
----- Object -3- ----- 12

----- group -4- -----
obj_nr = 4, i(381,352) -- f(408,354)
angle = 3, length = 27, closed=0
9
obj_nr = 4, i(408,354) -- f(405,384)
angle = 96, length = 25, closed=0
15
obj_nr = 4, i(405,384) -- f(379,382)
angle = -176, length = 26, closed=0
~11
obj_nr = 4, i(379,382) -- f(381,352)
angle = -85, length = 25, closed=1
----- Object -4- ----- ~14

```

A imagem final é a apresentada na Fig.10,

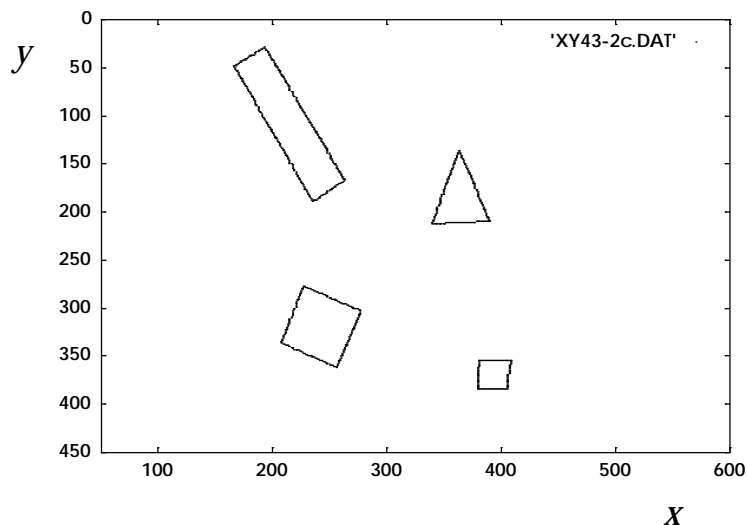


Fig. 10 - Segmentos após o algoritmo de extrapolação.

e a sequência associada a cada objecto é,

1. **32 – 92° – 135 – 88° – 33 – 90° – 134 – 90°**
2. **55 – 90° – 54 – 87° – 54 – 92° – 52 – 91°**
3. **51 – 64° – 66 – 44° – 65 – 70°**
4. **27 – 87° – 25 – 92° – 26 – 89° – 25 – 92°**

onde cada um dos ângulos internos é obtido por:

$$\theta_{\text{interno}} = |180^\circ - |\Delta\text{angle}||$$

com Δangle dado em graus.

De notar que o somatório dos ângulos internos é sempre 180° !

6.4.2 2ª Imagem: 1 Objecto

Em geral, quando o nível de ruído na imagem é desprezável e os segmentos são bem visíveis, o processo de encadeamento e extrapolação é completado com sucesso, obtendo-se cadeias fechadas que definem os objectos.

Se aquelas condições não se verificarem, um dos segmentos pode não figurar na lista final, obtendo-se assim cadeias abertas, tal como no exemplo que se apresenta a seguir (Fig.11).

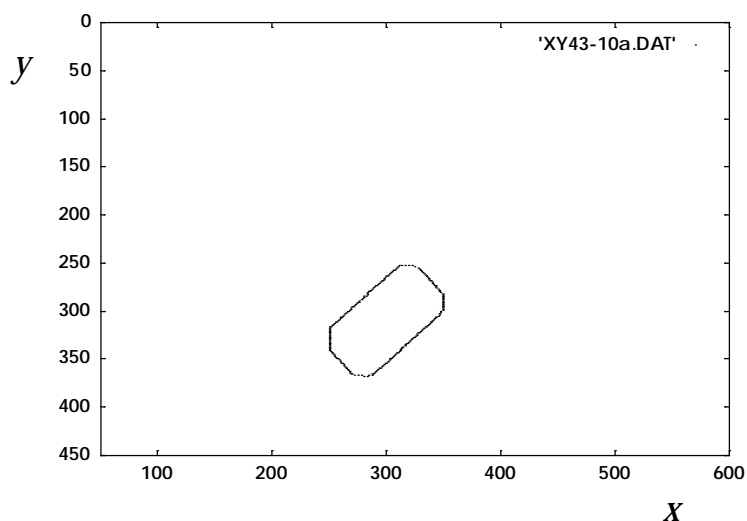


Fig. 11 - Pontos de fronteira pertencentes a 4 objectos de teste.

Trata-se de um objecto octaédrico com 4 segmentos de dimensão reduzida.

A lista de segmentos obtida após a fusão e eliminação de segmentos pequenos é a seguinte:

Segments.txt	(segmentos finais após a fusão)
Segment - 1- :	Segm(250,340) - (272,367) , angle= 45, length= 31)
Segment - 2- :	Segm(328,255) - (349,282) , angle= 46, length= 30)
Segment - 3- :	Segm(314,253) - (323,253) , angle= 0, length= 9)
Segment - 4- :	Segm(251,318) - (250,339) , angle= 93, length= 17)
Segment - 5- :	Segm(251,317) - (310,254) , angle=-41, length= 78)
Segment - 6- :	Segm(349,283) - (347,303) , angle= 96, length= 16)

Segment - 7- : Segm((288,367) - (346,304) , angle=-42, length= 78)

Estes segmentos encontram-se ilustrados na Fig.12 acompanhados do respectivo número com que aparecem na lista e da orientação.

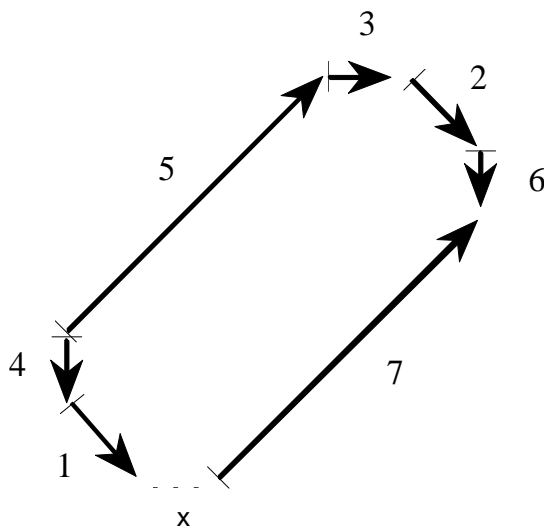


Fig. 12 - Resultado após a fusão e eliminação de segmentos pequenos (acompanhado de numeração e orientação).

Após a aplicação do algoritmo de encadeamento são obtidos os seguintes grupos (ou cadeias) de segmentos:

Chain.tx1 (antes de *Extra.c*)

```

----- group -1- -----
obj_nr = 1, i(250,340) -- f(272,367)
angle = 45, length = 31, closed=0
1

----- group -2- -----
obj_nr = 2, i(328,255) -- f(349,282)
angle = 46, length = 30, closed=0
2

obj_nr = 2, i(349,283) -- f(347,303)
angle = 96, length = 16, closed=0
6

obj_nr = 2, i(346,304) -- f(288,367)
angle = 138, length = 78, closed=0
    
```

```

~ 7
----- group -3- -----
obj_nr = 3, i(314,253) -- f(323,253)
angle = 0, length = 9, closed=0
3

----- group -4- -----
obj_nr = 4, i(251,318) -- f(250,339)
angle = 93, length = 17, closed=0
4

----- group -5- -----
obj_nr = 5, i(251,317) -- f(310,254)
angle = -41, length = 78, closed=0
5
    
```

Estes grupos traduzem as novas orientações dos segmentos tal como o ilustra a Fig.13.

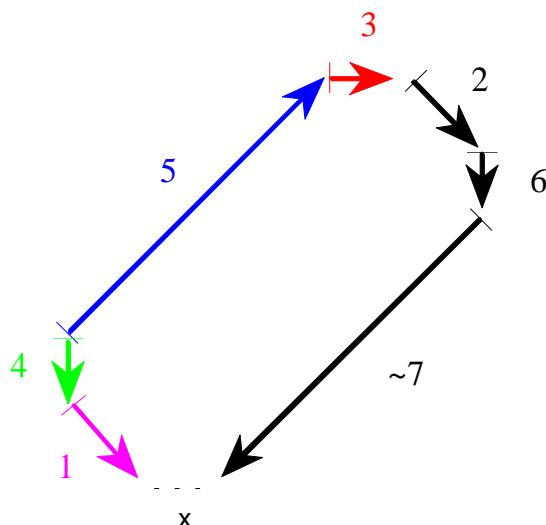


Fig. 13 - Encadeamento final dos segmentos (acompanhado de numeração e orientação).

Como se pode constatar, como resultado da ausência de um dos segmentos pequenos, apenas alguns deles formaram cadeias e o encadeamento total falhou. Isto implica a necessidade de se proceder a mais tentativas de ligação, agora entre grupos abertos de segmentos (ou cadeias). É precisamente esse o objectivo da secção seguinte, que utilizará este exemplo para testar a eficácia do algoritmo que irá ser apresentado.

6.5 A Ligação entre Grupos de Segmentos

Como se pôde verificar na lista apresentada atrás, a falha na detecção de um dos segmentos pequenos aliada às várias orientações dos segmentos, dá origem à criação para um mesmo objecto de mais do que um grupo (ou cadeia). Se o 1º segmento a ser considerado estivesse no meio do conjunto, então o encadeamento far-se-ia apenas num dos sentidos até se atingir uma das extremidades abertas, dando assim origem a pelo menos dois grupos. Apenas no caso desse 1º segmento se situar numa extremidade, seria possível obter o encadeamento correcto. Neste caso em particular, porém, e devido à sua orientação desfavorável, tal não será atingido, dando mais uma vez origem a mais do que um grupo. A solução neste caso poderia passar pela inversão prévia do segmento, mas no caso geral de se encontrar no meio da cadeia tal não daria o resultado pretendido. É claro que se poderia proceder ao encadeamento nos dois sentidos, mas pareceu mais simples ligar grupos *a posteriori*.

6.5.1 O Algoritmo Utilizado

Foi então desenvolvido um programa com o objectivo de testar a proximidade de grupos de segmentos previamente encadeados. Nele, são verificadas as posições de cada

grupo dentro da lista e esta copiada para um vector. Um outro vector de igual comprimento é também definido para receber o resultado das alterações efectuadas. O algoritmo consiste simplesmente na verificação de ligações entre grupos abertos ainda não inseridos no 2º vector. De cada vez que se verifica uma ligação, o conjunto é totalmente copiado para o 2º vector, sendo este marcado para não mais ser alvo de análise. Grupos fechados são de imediato copiados por não ser possível mais qualquer ligação. A verificação é feita entre cada um dos grupos abertos e qualquer um dos outros nas mesmas condições. Quando o 1º tiver percorrido todos os grupos possíveis, estará terminada a 1ª iteração. O programa terminará a sua execução na n^a iteração, logo que se verifique não ocorrerem mais ligações.

O vector 2 é sempre o vector destino e 1 o vector fonte. Sempre que se efectua uma iteração na qual grupos são ligados e copiados na nova ordem para 2, os respectivos apontadores são trocados para manter aquela definição (evitando assim copiar tudo de volta para 1). Quando não houver mais alterações, o conjunto de grupos final estará no vector 2.

É o seguinte o programa principal:

algoritmo de ligação de grupos

- conta o nº de segmentos na lista
- reserva memória para o vector fonte: *chain1[]*
- reserva memória para o vector destino: *chain2[]*
- copia o conteúdo da lista para *chain2[]*
- **enquanto houver ligações chama *one_iteration()***
- copia o conteúdo de *chain2[]* de volta para a lista ligada
- liberta a memória RAM utilizada

Fig. 14 - Descrição do programa principal de ligação entre grupos.

A função que efectua as ligações e que é chamada tantas vezes quantas as necessárias, é *one_iteration()* e a sua descrição é apresenta no Anexo E.3.

6.5.2 Ilustração das Várias Fases de Ligação

Vamos considerar o exemplo da Fig.11.

De início é tomado o 1º grupo que consiste apenas no segmento 1 e verificada uma possível ligação com qualquer um dos restantes. Este liga com o grupo 4 (segmento 4) desde que ambos invertam o seu sentido. São então ambos transferidos para o vector 2. De seguida é considerado o grupo 2 que contém 3 segmentos: o nº 2, 6 e 7 invertido. Este liga com o grupo 3 (segmento 3). Uma vez que o 2º grupo considerado deve ser

colocado no fim da cadeia correspondente ao 1º grupo, ambos são invertidos, passando o grupo comum a conter os segmentos nº 7, ~6, ~2 e ~3 (o til representa o segmento invertido) que é de imediato transferido também para o vector 2. Resta o grupo 5 que por não haver mais grupos a que ligar é também transferido para aquele vector.

Após esta 1ª iteração, obtém-se então o seguinte resultado:

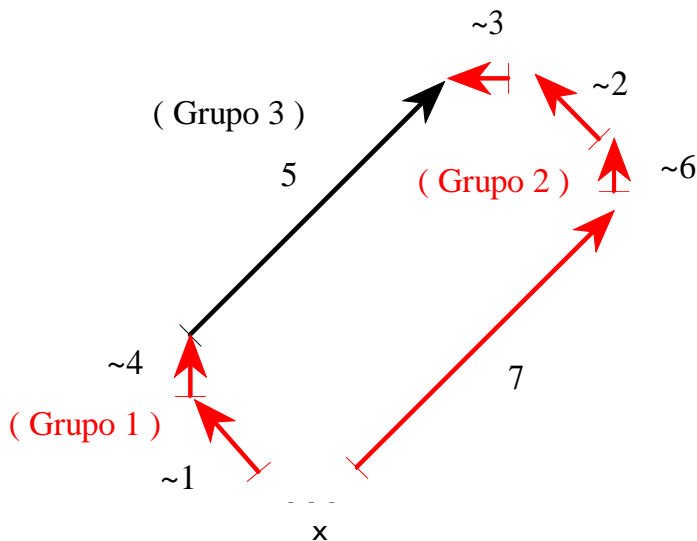


Fig. 15 - Resultado após a 1ª iteração.

```

----- group -1- -----
obj_nr = 1, i(272,367) -- f(250,340)
angle = -135, length = 31, closed=0
~1
obj_nr = 1, i(250,339) -- f(251,318)
angle = -87, length = 17, closed=0
~4
----- group -2- -----
obj_nr = 2, i(288,367) -- f(346,304)
angle = -42, length = 78, closed=0
7
obj_nr = 2, i(347,303) -- f(349,283)
    
```

```

angle = -84, length = 16, closed=0
~6
obj_nr = 2, i(349,282) -- f(328,255)
angle = -134, length = 30, closed=0
~2
obj_nr = 2, i(323,253) -- f(314,253)
angle = 180, length = 9, closed=0
~3
----- group -3- -----
obj_nr = 3, i(251,317) -- f(310,254)
angle = -41, length = 78, closed=0
5
    
```

Obtiveram-se 3 grupos que serão sujeitos a nova iteração. Inicia-se de novo o processo com o grupo 1 que liga ao 3 (segmento 5), sendo este invertido e todo o conjunto transferido para o vector 2. Resta todo o grupo 2 que por não poder efectuar mais nenhuma ligação é transferido também para o vector.

Após 2 iterações obtém-se então o seguinte resultado:

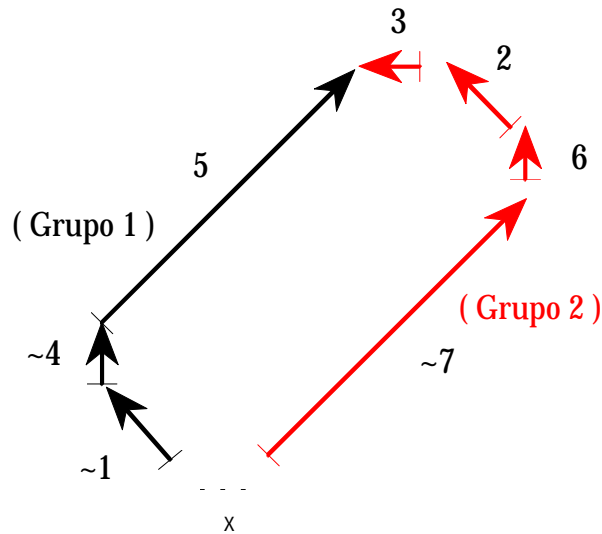


Fig. 16 - Resultado após a 2ª iteração.

```

----- group -1 -----
obj_nr = 1, i(272,367) -- f(250,340)
angle = -135, length = 31, closed=0
~1
obj_nr = 1, i(250,339) -- f(251,318)
angle = -87, length = 17, closed=0
~4
obj_nr = 1, i(251,317) -- f(310,254)
angle = -41, length = 78, closed=0
5
----- group -2 -----
obj_nr = 2, i(288,367) -- f(346,304)
    
```

```

angle = -42, length = 78, closed=0
7
obj_nr = 2, i(347,303) -- f(349,283)
angle = -84, length = 16, closed=0
~6
obj_nr = 2, i(349,282) -- f(328,255)
angle = -134, length = 30, closed=0
~2
obj_nr = 2, i(323,253) -- f(314,253)
angle = 180, length = 9, closed=0
~3
    
```

Finalmente, obtêm-se apenas 2 grupos que se ligam após a inversão do 2º que contém os segmentos nº 7, ~6, ~2 e ~3. Ambos são transferidos para o vector 2 na sequência correcta e mais nenhuma ligação ou transferência se efectua.

Então, e após 3 iterações, obtém-se o seguinte resultado final:

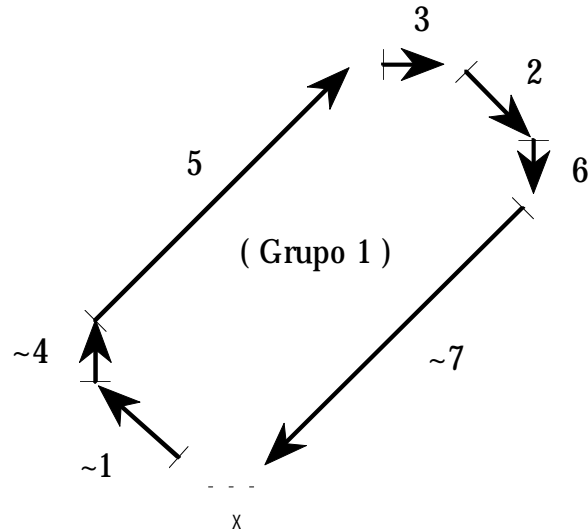


Fig. 17 - Resultado após a 3ª e última iteração.

<pre> ----- group -1- ----- obj_nr = 1, i(272,367) -- f(250,340) angle = -135, length = 31, closed=0 ~1 obj_nr = 1, i(250,339) -- f(251,318) angle = -87, length = 17, closed=0 ~4 obj_nr = 1, i(251,317) -- f(310,254) angle = -41, length = 78, closed=0 5 obj_nr = 1, i(314,253) -- f(323,253) </pre>	<pre> angle = 0, length = 9, closed=0 3 obj_nr = 1, i(328,255) -- f(349,282) angle = 46, length = 30, closed=0 2 obj_nr = 1, i(349,283) -- f(347,303) angle = 96, length = 16, closed=0 6 obj_nr = 1, i(346,304) -- f(288,367) angle = 138, length = 78, closed=0 ~7 </pre>
--	---

Agora, e após 3 iterações, os segmentos já se encontram totalmente encadeados formando um só objecto como era pretendido. Uma nova chamada ao procedimento de ligação não provoca mais qualquer alteração pelo que, detectado esse facto, a execução cessa e a sequência final é copiada de volta ao vector 1.

O novo encadeamento de segmentos seria então o seguinte:

1. ~1 - ~4 - 5 - 3 - 2 - 6 - 7

e o código sequencial comprimento-ângulo associado a este objecto seria:

1. 31 - 132° - 17 - 134° - 78 - 139° - 9 - 134°
 - 30 - 130° - 16 - 138° - 78 - ♦

Ou, usando uma forma mais legível:

l	θ
31	132°
17	134°
78	139°
9	134°
30	130°
16	138°
78	◆

O último ângulo não existe uma vez que se trata de um objecto aberto.

Uma vez obtido o encadeamento correcto de todos os segmentos pertencentes à fronteira do objecto, estes já se podem submeter ao algoritmo de determinação dos vértices que permite a correcção do comprimento dos segmentos e dos ângulos que estes formam entre si (Fig.18).

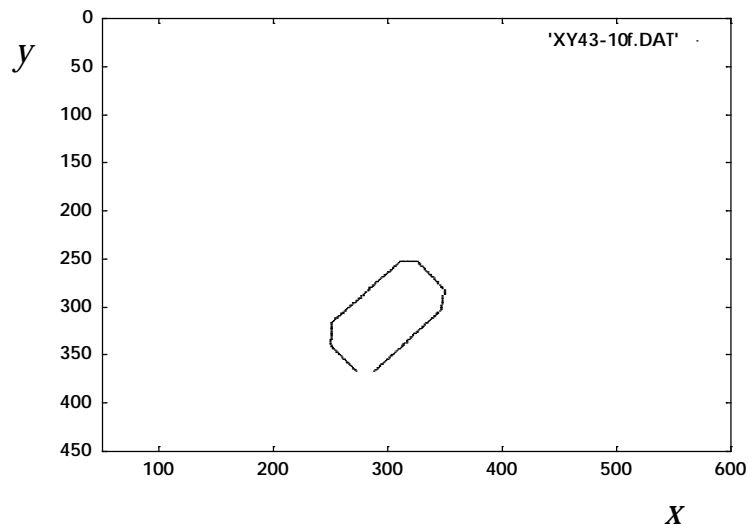


Fig. 18 - Apresentação final dos segmentos após a ligação ente grupos e a extrapolação (eles já se encontram todos ligados formando 1 só objecto).

A tabela seguinte apresenta já essas correcções:

```

----- group -1 -----
obj_nr = 1, i(272,367) -- f(249,339)
angle = -134, length = 32, closed=0
~1
obj_nr = 1, i(249,339) -- f(251,316)
angle = -84, length = 19, closed=0
~4
obj_nr = 1, i(251,316) -- f(310,253)
angle = -41, length = 78, closed=0
5
obj_nr = 1, i(310,253) -- f(326,253)

```

```

angle = 0, length = 16, closed=0
3
obj_nr = 1, i(326,253) -- f(349,282)
angle = 46, length = 33, closed=0
2
obj_nr = 1, i(349,282) -- f(347,302)
angle = 96, length = 16, closed=0
6
obj_nr = 1, i(347,302) -- f(288,367)
angle = 137, length = 80, closed=0
~7

```

Finalmente o respectivo código sequencial será o apresentado a seguir:

l	θ
32	130°
19	137°
78	139°
16	134°
33	130°
16	139°
80	♦

6.6 A Identificação de Objectos

Uma vez conseguida uma sequência de pares comprimento-ângulo, esta pode ser utilizada para caracterizar um dado objecto num determinado estado estável. Ou seja, ela poderá de futuro ser aplicada em processos de reconhecimento.

Uma vez que só tem interesse para aquele processo os valores do par, toda a informação contida na lista deixa de ter interesse pelo que pode ser libertada de memória. Basta manter um vector que contenha aqueles valores.

Como informação auxiliar poderão ser utilizados vectores que transportam a informação respectivamente sobre o estado fechado ou aberto de cada um dos objectos e sobre o número de pares segmento-ângulo que cada um possui.

Podemos ainda considerar algumas variantes no processo de parametrização das cadeias de segmentos. Até aqui considerámos apenas sequências de segmentos pertencentes à fronteira dos objectos, mas o processo pode ser estendido a segmentos internos quando aqueles apresentem buracos (Fig.19).

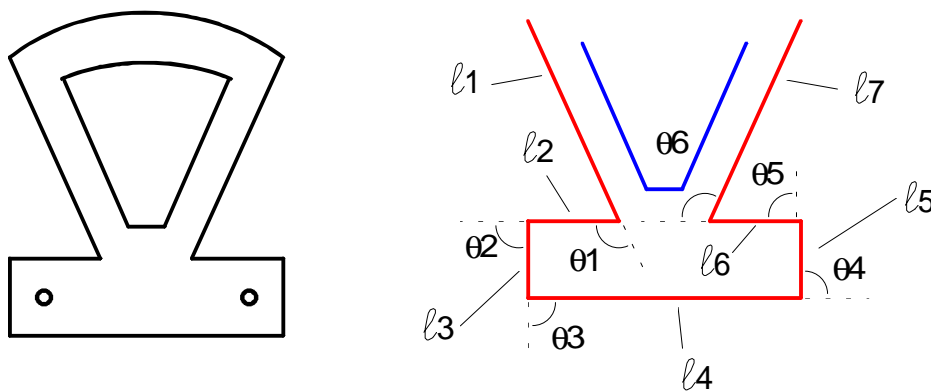


Fig. 19 - Objecto sujeito ao processo de identificação: neste caso os segmentos internos podem também ser utilizados para caracterizar o objecto. Mostra-se ainda um outro modo de medir os ângulos.

Por outro lado, e em alternativa aos ângulos internos, podem ser também utilizados os ângulos que formam os vectores correspondentes aos segmentos orientados, como se mostra na Fig.19. Nessa modalidade os ângulos associados aos comprimentos poderão ser positivos ou negativos consoante se rode num ou noutro sentido. A tabela de pares $l-\theta$ seria então a seguinte:

l	θ
l_1	$-\theta_1$
l_2	$+\theta_2$
l_3	$+\theta_3$
l_4	$+\theta_4$
l_5	$+\theta_5$
l_6	$-\theta_6$
l_7	♦

Finalmente, o processo de identificação deve obedecer ao esquema geral apresentado na Fig.20.

Contemplando a possibilidade exemplificada atrás de um objecto não estar completo, ou seja, faltar um ou mais segmentos, a diferença de "comprimento" entre a cadeia obtida da imagem e a do modelo não deve implicar a sua rejeição imediata. Pelo contrário, as duas cadeias deverão ser comparadas apenas na extensão da obtida a partir da imagem e num número de vezes igual à diferença entre os dois números de pares comprimento-ângulo (correspondendo ao "deslizamento" dessa cadeia até que os últimos comprimentos das duas cadeias estejam justapostos). Este procedimento deverá depois ser repetido para a cadeia invertida uma vez que o processo de encadeamento pode ter sido efectuado em qualquer um dos dois sentidos possíveis.

6.7 A Separação de Cadeias Pertencentes a Objectos Diferentes

Este processo de identificação pode ainda ser utilizado para reconhecer objectos parcialmente visíveis devido a sobreposição. Se a cadeia resultante da análise à fronteira visível do objecto parcialmente oculto for distinguível da restante, então ela poderá ser alvo de comparação com as que se encontram registadas na base de dados.

No caso esquematizado na Fig.21, dois objectos poliédricos encontram-se parcialmente sobrepostos.

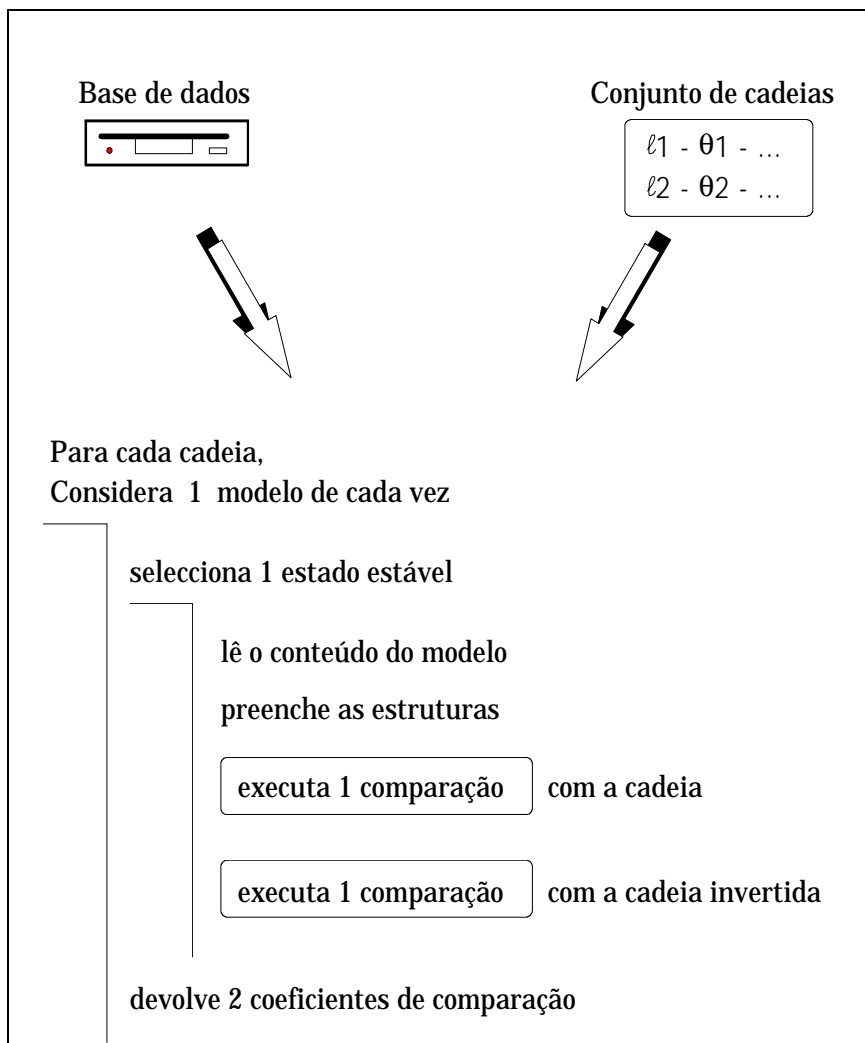


Fig. 20 - Esquema geral de comparação de cada cadeia obtida da imagem com qualquer modelo da base de dados.

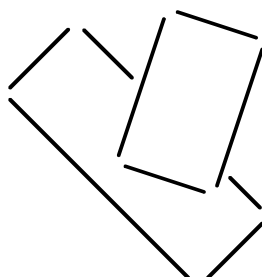


Fig. 21 - Análise de cadeias para um caso simples de sobreposição parcial.

Numa análise dos segmentos presentes na imagem (que implicaria a sua detecção a partir da imagem em níveis de cinzento e não binarizada) obter-se-iam duas cadeias distintas. Aquela associada ao objecto totalmente visível pode ser utilizada de imediato para uma tentativa de reconhecimento, enquanto que a restante deverá ser submetida ao processo com as devidas reservas dada a inexistência de informação completa.

O procedimento descrito acima implica que se possa conhecer de antemão qual a cadeia associada ao objecto visível e ao parcialmente oculto! Para que tal seja possível é enviado ao Agente Laser um pedido especial de aquisição de alturas cuja descrição detalhada pode ser encontrada no capítulo 7. Trata-se do envio do pedido codificado como "f" que faz executar um programa adequado daquele agente que consiste na aquisição de valores de altura associados a alguns pontos na imagem caracterizados pelas suas coordenadas (x,y) . Estes pontos pertencem às duas cadeias, e os valores de altura que a eles são então associados, permitem a sua distinção.

Para que o algoritmo de encadeamento possa fornecer as coordenadas dos pontos e ter conhecimento dessas várias alturas, envia conjuntamente com o pedido, um ficheiro que corresponde ao registo de um vector de estruturas; estas contêm campos para as coordenadas, quer de visão (x,y) já preenchidas, quer no sistema de referência do robot (X,Y,Z) , a preencher.

Após consulta da imagem de alturas ou do ficheiro de amostragem de alturas, este ficheiro é completado com a informação dos valores de Z associados aos pontos (x,y) nele contidos (a estrutura contém ainda espaço para o registo das coordenadas dos pontos no referencial do robot após conversão usando a matriz de transformação). O ficheiro é depois devolvido à área de acesso comum dos agentes onde o agente Visão, ao qual pertence o algoritmo de encadeamento, pode ir buscá-lo logo que seja notificado para tal. Dessa forma as duas cadeias ficarão devidamente separadas e associadas correctamente quer ao objecto parcialmente oculto quer ao totalmente visível.

É claro que o facto da cadeia desse objecto estar incompleta, pode comprometer o processo de identificação. De facto, e a menos que ele possua características bem distintas de outros objectos (como por exemplo possuir um lado completamente visível com um comprimento único), o reconhecimento pode não ser possível por haver mais do que um modelo encontrado que se adapte à pouca informação obtida do objecto. Ele irá constituir então, não o processo de identificação adoptado, mas sim uma primeira tentativa de reconhecer os objectos presentes na imagem, dada a grande velocidade de execução do algoritmo. Ou, alternativamente, constituir um último recurso no caso dos outros processos (dois dos quais desenvolvi noutros capítulos) falharem alguma identificação.

6.8 Conclusão

Como se pôde verificar com os exemplos apresentados, foi possível obter o encadeamento correcto do conjunto de segmentos que definem a fronteira de um objecto e ainda conseguida a extrapolação dos mesmos para o seu real valor, uma vez encontrados os vértices. O conhecimento das coordenadas das extremidades, uma vez encontrados aqueles, permitiu ainda corrigir os ângulos que eles formavam, embora os valores prévios já fossem bastante próximos dos reais.

Uma vez encontrado um modo de obter com rigor os comprimentos dos segmentos de fronteira de um objecto assim como os ângulos que eles formam entre si, ficou aberto o caminho para um processo alternativo de identificação a 2D.

7.1 Introdução

O Agente Laser é a entidade responsável pela recepção de pedidos de outros agentes, pelo lançamento de todas as acções julgadas necessárias e pelo envio dos resultados obtidos a toda a comunidade de agentes (e não só àquele que efectuou o pedido) por forma a que todos eles possam dispor e usufruir da informação por ele obtida.

Os dois capítulos seguintes farão uma descrição exaustiva do sensor de medida de distâncias que foi construído assim como de todo o software desenvolvido para o seu controle e que constituirão a base sólida de funcionamento deste agente Laser.

Uma vez na posse de todo esse hardware e software capazes de efectuar amostragens de alturas, que no caso da resolução ser máxima leva à construção de imagens de altura ("range images"), resta a formulação de um módulo próprio que efectue a respectiva gestão consoante as necessidades num qualquer processo de manipulação robótica. Esse módulo será ainda responsável por uma série de acções posteriores a partir dessa imagem que levem à identificação a 3D dos objectos presentes, ou alternativamente, ao cálculo de um conjunto de parâmetros que permitam a manipulação segura de objectos em situação de ambiguidade.

Todo esse pacote de software foi reunido numa só máquina (um PC 486DX/33Mhz) que no seu funcionamento autónomo constituirá o Agente Laser. Tal como já foi afirmado, este agente manterá um contacto permanente com a restante comunidade de agentes (incluindo o Agente Visão responsável pelas análises de imagem a 2D) através da troca de mensagens com pedidos e ficheiros de resultados.

7.2 Explicação Breve sobre a Abordagem a Utilizar no Processo de Identificação a 3D

Um dos grandes objectivos do processo de identificação a 3D foi o de ultrapassar as limitações dos algoritmos a 2D correspondentes, permitindo resolver

situações de sobreposição de objectos e portanto de oclusão parcial, na condição porém dessas situações não serem muito complexas.

A ideia base que deu origem a toda a formulação foi a de separar, em termos de cálculo e registo de parâmetros os pixels referentes a cada um dos objectos naquela situação. Se eles, devido àquela situação, se encontrarem em cotas diferentes então poderá ser escolhido um valor de altura que defina a sua separação numa posterior análise geométrica dirigida a apenas um dos objectos envolvidos naquela sobreposição. Essa análise é conseguida com uma versão devidamente adaptada do programa de varrimento da imagem e medição de parâmetros geométricos usado no agente Visão.

Esse valor de altura pode ser facilmente obtido através da análise do histograma 3D desde que não haja partilha dos mesmos valores de altura para pixels pertencentes aos dois objectos. Esse histograma deverá consistir na acumulação do número de pixels encontrados na imagem de alturas em função desses mesmos valores de alturas codificados em níveis de cinzento (entre 0 e 255 uma vez que o ficheiro de imagem registado na placa de aquisição de imagem reserva apenas 8 bits para cada valor). Se se mantiver aquela última condição, então o valor deverá estar a meia "distância" entre os limites das formas contidas no histograma referentes aos dois objectos (Fig.1).

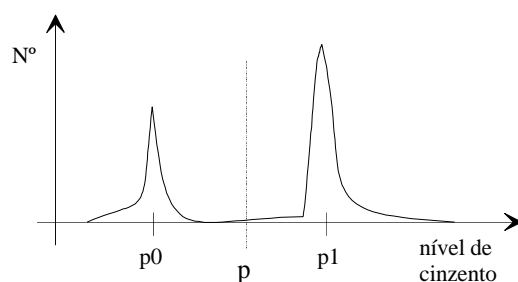


Fig. 1 - Exemplo de um histograma de alturas de duas faces planas paralelas ao plano horizontal.

A execução do programa de segmentação a 3D, aplicado apenas a um dado intervalo de níveis de cinzento delimitado por aquele valor encontrado de altura, permitirá então a obtenção de dados relativos a apenas um dos objectos, preferencialmente do objecto situado na cota mais alta uma vez que se encontra completo na imagem.

Para além da separação dos objectos em estado de sobreposição, pretende-se também identificar os objectos presentes na imagem, cuja classificação irá concorrer com aquela obtida pelo algoritmo de identificação a 2D do Agente Visão. Sempre que possível, far-se-á também a identificação dos objectos superiores em estado de sobreposição. Ora, uma vez que não se irá utilizar a descrição dos objectos por parâmetros utilizados em CAD (não só porque o volume de informação a tratar, dadas as várias possibilidades de perspectiva do objecto, seria imenso com implicações proibitivas em termos de tempo de execução, como também porque ele só contempla objectos poliédricos, o que não acontece forçosamente com o algoritmo desenvolvido) isto implica que para se obter uma grande probabilidade de identificação os objectos a submeter ao algoritmo de identificação a 3D se encontrem num dos seus estados estáveis (Fig.2). Esta restrição não é muito penalizadora uma vez que a situação mais vulgar será precisamente a de objectos espalhados na área de trabalho do robot, constituindo a sobreposição um caso esporádico a ser resolvido; e quando ocorre, o

objecto superior pode sempre ser identificado se se mantiver num dos seus estados estáveis. E mesmo no caso do estado de sobreposição implicar a rotação no espaço do objecto superior que impeça a sua identificação, poderão ser calculados parâmetros especiais que permitam a sua manipulação robótica segura, que poderá consistir somente na retirada do objecto superior para uma zona livre da área de acção do robot e visível pela câmara, para uma posterior reanálise.

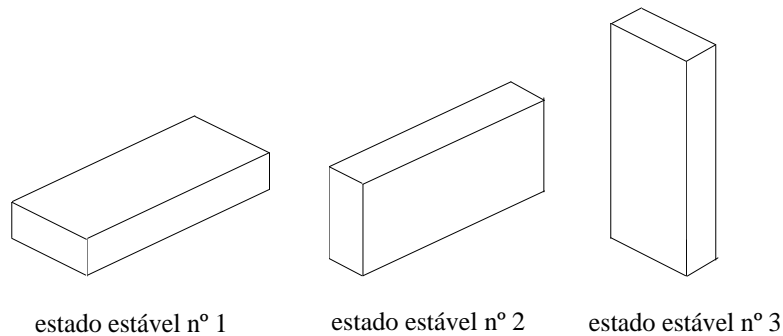


Fig. 2 - Apresentação de todos os estados estáveis de um paralelepípedo (convenção idêntica à utilizada a 2D).

Sendo assim escolheu-se um parâmetro geométrico, $sumZ2$, que é uma característica do tipo e dimensão de cada objecto, que ultrapassa a restrição dele consistir num poliedro, podendo portanto tomar qualquer forma desde que se mantenha num dos seus estados estáveis. Aquele valor consiste no momento de inércia geométrico relativamente ao plano horizontal que o sustenta. No caso de se encontrar a uma altura Z_0 do plano horizontal XOY devido a uma sobreposição, então o cálculo de $sumZ2$ que inicialmente é sempre efectuado relativamente a este plano deverá ser corrigido tendo em conta aquela altura, mantendo-se assim esta característica como um invariante geométrico dos objectos e por isso utilizável no processo de identificação.

A este valor foi ainda acrescentado um conjunto de parâmetros que facilitam a caracterização dos objectos no caso de possuírem faces paralelas ao plano horizontal (e sempre nos seus estados estáveis):

- Altura a que a face se encontra de XOY.
- Área a ela associada que consiste no nº de pixels encontrados.
- Momento de 2ª ordem do próprio histograma de alturas, $sumNr2$ relativamente ao eixo dos ZZ codificados em níveis de cinzento.

Quer as alturas quer as áreas a elas associadas são registadas num número máximo de 3, correspondendo à possibilidade de um objecto poder ter 3 faces planas paralelas ao plano horizontal. Estes valores podem ser extraídos directamente do histograma de alturas como se verá na secção referente às "Assinaturas".

Toda esta informação foi recolhida numa estrutura única a que chamei de "Assinatura" do objecto (não confundir com a assinatura de uma figura 2D) que obviamente estará associada a apenas um dos estados estáveis. Cada estado estável possuirá então a sua "Assinatura" e o conjunto de todas as "Assinaturas" possíveis de

um objecto constituirá toda a informação 3D a ele associada. Essa informação, se registada numa base de dados, poderá ser usada posteriormente como elemento de comparação para potenciais identificações.

Essa estrutura fica então com a seguinte forma explícita:

<i>sumZ2</i> :	momento de inércia geométrico
<i>sumNr2</i> :	momento de 2ª ordem do histograma de alturas
<i>Z[3]</i> :	registo de um máximo de 3 alturas
<i>nr[3]</i> :	registo de um máximo de 3 áreas

São apresentados a seguir alguns exemplos de objectos ou pares de objectos em estado de sobreposição que os algoritmos de sub-segmentação e de identificação do Agente Laser conseguem resolver (Figs. 3, 4 e 5).

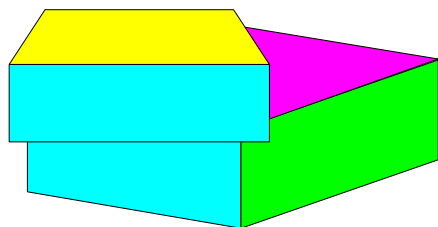


Fig. 3 - Sobreposição parcial de dois objectos poliédricos nos seus estados estáveis.

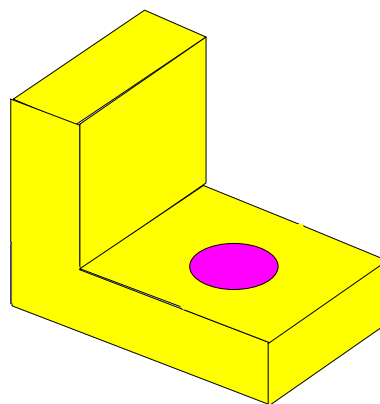


Fig. 4 - Objecto num dos seus estados estáveis, com duas faces paralelas ao plano horizontal.

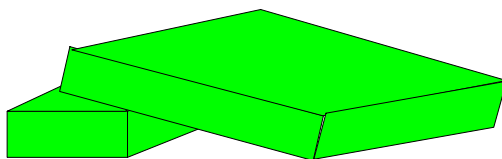


Fig. 5 - Dois objectos em estado de sobreposição parcial, encontrando-se o objecto superior numa posição considerada como não estável.

7.3 Descrição do Programa Principal do Agente Laser

No início da sua execução, este programa começa por efectuar o "display" de um ecrã de apresentação do Agente Laser, contendo a sua acção genérica, a autoria do trabalho desenvolvido e o local do seu desenvolvimento.

Segue-se imediatamente a apresentação das opções disponíveis ao utilizador juntamente com a informação do estado do agente e mensagens enviadas pelos programas em execução. Este ecrã poderá ser apagado durante aquela execução por necessidade de utilizar todo o ecrã para o registo de mensagens e pedidos de escolha de opções várias, mas é sempre mostrado de novo logo que ela termine.

Numa versão deste programa o utilizador pode intervir, efectuando a escolha duma destas opções principais (para além de outras no decurso da execução dos vários programas). Contudo é suposto que este agente funcione autonomamente pelo que normalmente ele é posto em ciclo fechado no qual se encarrega de recolher os pedidos de outros agentes e de enviar os resultados obtidos. Nesse caso todas as opções secundárias que aparecem durante essa execução, como "displays" em monitor ou registos de ficheiros em disco, são eliminadas sendo as chamadas aos programas correspondentes mantidas ou não consoante a escolha do programador.

Neste programa é feita a escolha dos algoritmos a utilizar consoante os pedidos efectuados assim como a sequência correcta das suas chamadas.

Mais adiante será apresentada a descrição de todos os programas envolvidos, quer aqueles que serão executados "on-line" quer os auxiliares ("off-line") que se encarregam da manutenção e actualização de informação a ser utilizada pelo agente. Exemplo deste último é o programa de gestão dos modelos 3D registados em disco.

Antes ainda do lançamento dos algoritmos, é da responsabilidade deste programa de gestão, a verificação da disponibilidade do sistema de aquisição de imagem que é comum ao Agente Visão e só permite a execução da acção seleccionada após verificação de "*flags*" definidas especificamente com esse objectivo:

<i>flag_Vision</i> :	endereço → 0×9FFFE
<i>flag_Laser</i> :	endereço → 0×9FFFF

A execução só é permitida quando *flag_Vision*=0 e nessa circunstância *flag_Laser* é posta imediatamente a 1 para impedir o acesso do agente concorrente à placa de aquisição de imagem, ficando esta ao dispor exclusivo do Agente Laser. Logo que termine a sua acção, *flag_Laser* é reposta a 0, disponibilizando assim o sistema de aquisição de imagem.

A escolha daqueles endereços contemplou a possibilidade daqueles agentes correrem separadamente em janelas de Windows de um mesmo PC, uma vez que eles provaram ser sempre "visíveis" de qualquer janela DOS e partilhadas por quaisquer número delas que sejam abertas naquele ambiente.

7.3.1 A Recepção de Pedidos e o Envio de Resultados

A troca de informação com os restantes agentes, seja ela composta por pedidos ou por resultados, é feita sempre através de uma área de acesso comum (tal como já acontecia com o Agente Visão). Trata-se de um directório específico do disco de uma estação de trabalho, cujo acesso físico é efectuado através de uma ligação Ethernet. A

transferência de ficheiros que contêm aquela informação, no caso da máquina onde corre o Agente Laser, é então efectuada através da execução de comandos de FTP registados em ficheiros próprios e lidos por este programa. Essa leitura em "run-time" é permitida por uma função da linguagem C (utilizada na programação dos algoritmos) que permite fazer correr ficheiros executáveis do DOS:

```
system ( "ftp <file" );
```

onde *file* corresponde ao nome desse ficheiro. Se a acção for a transferência do ficheiro que contém os pedidos, então o ficheiro é *file*=*'login3D'* e o seu conteúdo é o seguinte:

```
login3D
open turing
user pal PauloLopes
cd laser
lexec w-8
get requests.Las
lexec w-8
close
bye
```

A Fig.6 ilustra a transferência do ficheiro de pedidos *'requests.Las'* enviado por exemplo pelo Agente Visão.

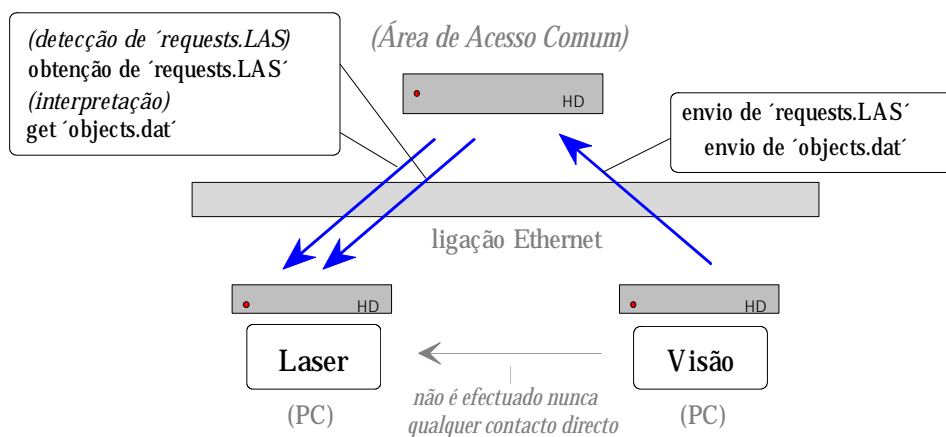


Fig. 6 - Exemplo de envio de um pedido pelo Agente Visão.

Para outras acções serão utilizados outros ficheiros de comandos. Para o envio dos resultados obtidos por este agente contidos no ficheiro *'results.Las'*, por exemplo, será utilizado o ficheiro *file*=*'s_result.Las'* que envia para a área de acesso comum aquele ficheiro. Ainda, para o envio de informação para a remoção de objectos em estado de sobreposição, cujos dados se encontram no ficheiro *'remove.Las'*, o ficheiro a ser utilizado será *file*=*'s_remove.Las'*. Ambos os ficheiros são mostrados na Fig.7.

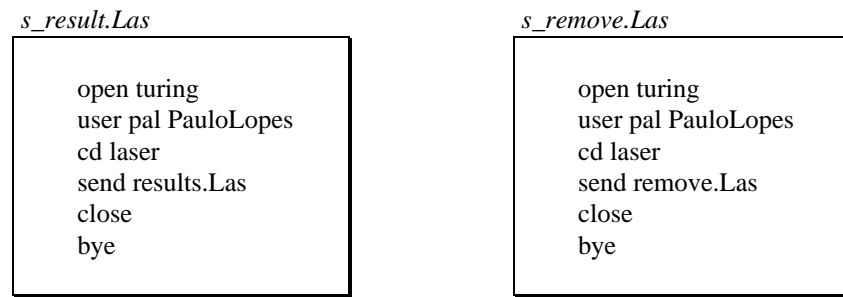


Fig. 7 - Ficheiros de comandos destinados ao programa FTP.

A Fig.8 ilustra a transferência de ficheiros para o Agente Visão (através da área de acesso comum) contendo a informação solicitada. No exemplo retratado, o Agente Visão envia um 1º ficheiro, 'result3D', contendo o nome do ficheiro com informação útil, e de seguida, este último: 'objects.dat'.

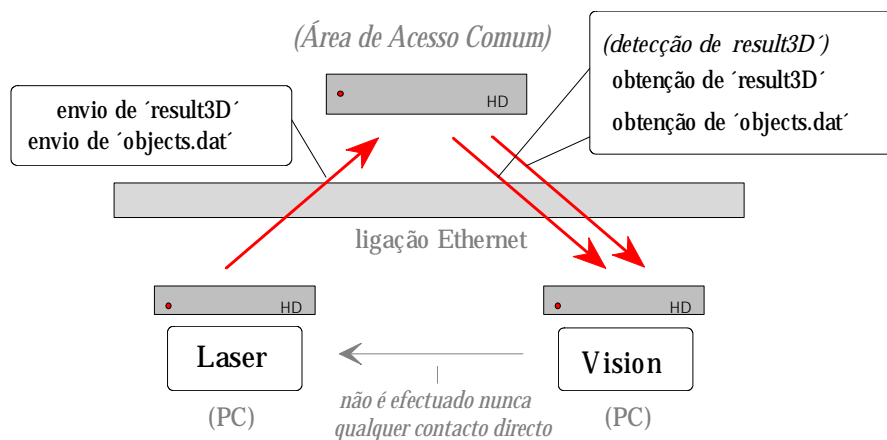


Fig. 8 - O envio dos resultados após execução do pedido.

7.3.2 As Opções de Funcionamento do Agente Laser

A tabela seguinte (1) apresenta uma lista completa de todas as opções possíveis, cada uma correspondendo a um pedido específico que é transportado no ficheiro de pedidos 'requests.Las' cuja codificação é conhecida de antemão pelos restantes agentes.

Tabela 1

Nº dos pedidos	Código	Significado	Parâmetros auxiliares (opcionais ou obrigatórios)
1	g	global	resolução do varrimento
2	l	local	resolução do varrimento + coordenadas do rectângulo delimitador
3	r	random	resolução do varrimento
4	i	identify	(não possui)
5	f	few	resolução do varrimento
6	b	build	coordenadas do rectângulo delimitador

A tabela 2 descreve em pormenor a acção do agente para cada uma daquelas opções.

Tabela 2

Código	Acção
g	Ordem de varrimento global (na sua extensão máxima) com especificação da resolução a utilizar; são registados em disco os ficheiros de amostragem de alturas (em texto e binário) cuja visualização posterior pode ser obtida; se a resolução pedida for a máxima (1), então o programa utilizado ordena a construção imediata da imagem de alturas através do uso de filtros adequados e por fim o seu registo.
l	Ordem de varrimento local sendo fornecida a resolução do varrimento a efectuar assim como os dois pares de coordenadas que definem o rectângulo delimitador; mais uma vez, se a resolução pedida for a máxima, é ordenada a construção da imagem de distâncias que embora só contenha as figuras encontradas dentro daquele rectângulo é registada no mesmo formato de 5×64 linhas com 640 pixeis cada, num total de 204.800 bytes de comprimento.

r	<p>Preenchimento do campo "<i>real_height</i>" da estrutura "<i>struct object_</i>" que constitui a estrutura de dados relativo a cada objecto 2D definido pelo Agente Visão. As estruturas estão disponibilizadas sob a forma de vector na área de acesso comum dos agentes.</p> <p>O programa chamado lê as coordenadas do centro de massa de cada objecto (em unidades de pixel) e as associa ao valor da altura correspondente lida de qualquer dos ficheiros disponíveis com informação 3D. No caso de não existir nenhum é dada ordem de varrimento laser para construção do ficheiro de alturas.</p>
i	<p>Ordem de detecção e identificação de objectos a 3D.</p> <p>Lança o programa que controla a segmentação da imagem de alturas, a sub-segmentação das figuras associadas a situações de sobreposição e por fim a identificação dos objectos presentes (ou o cálculo de parâmetros que permitam a extracção dos objectos superiores).</p>
f	<p>Atribuição de alturas a pontos pertencentes a segmentos.</p> <p>Nesta opção é feito o preenchimento do campo "<i>Z</i>" num vector de estruturas fornecido pelo Agente Visão e disponível na área de acesso comum dos agentes.</p> <p>Este ficheiro é criado pelo módulo "<i>Hough</i>" na sequência da aplicação do algoritmo de Hough aperfeiçoado e com pós-processamento e da sequenciação de segmentos e ângulos que constitui o algoritmo "<i>Chain Code</i>". Ele consiste num conjunto de coordenadas (<i>x,y, X,Y,Z</i>) nos sistemas de referência respectivamente de visão e do robot, associados a pontos chave pertencentes a uma ou mais sequências de segmentos dos quais se pretende conhecer a altura. O objectivo é obter a separação de cadeias de segmentos pertencentes a objectos diferentes numa situação de sobreposição.</p>
b	<p>Ordem específica de construção de uma nova imagem de alturas (mesmo que já exista uma). O objectivo é actualizar aquela imagem após uma acção concreta de manuseamento robótico que altere o "estado do mundo" dentro do raio de acção do robot e visível pela câmara.</p> <p>É fornecido o rectângulo delimitador da área de interesse através de dois pares de coordenadas para mais rápida execução. No final desta acção o próprio ficheiro de imagem das alturas é sempre enviado para a área de acesso comum dos agentes para eventuais consultas por aqueles.</p>

No Anexo F é apresentado o esquema de chamadas a programas e funções, efectuadas pelo programa principal do Agente Laser.

7.4 Descrição dos Programas Envolvidos e Apresentação dos Capítulos Seguintes

Os capítulos seguintes farão uma descrição pormenorizada de todos os algoritmos contidos neste Agente Laser e que permitem atingir os objectivos anunciados acima e que consistem nos seguintes:

No capítulo **8** será feita a descrição do sensor de alturas por varrimento laser, por mim desenvolvido e construído no laboratório de Visão e Robótica da FEUP, assim como todo o software de controle.

No capítulo **9** será descrito o desenvolvimento matemático que levou à medida dos valores de altura e o processo de construção das imagens de altura. Será ainda descrito todo o processo de calibração do sistema laser e ainda a análise da deformação da imagem dada pela câmara, essencial para a correcta avaliação das alturas.

No capítulo **10** será feita a apresentação dos elementos caracterizadores das figuras a 3D que irão constituir aquilo a que se convencionou chamar de "Assinaturas". Será ainda feita a análise da fiabilidade dos parâmetros que a constituem. Será ainda feita uma descrição do ficheiro de modelos 3D utilizado, no qual é feito o registo daquelas "Assinaturas", assim como do seu programa de gestão.

No capítulo **11** será descrito todo o processo de segmentação das imagens 3D, quer primária quer a 2º nível, e que leva à atribuição de valores de "Assinatura" aos objectos 3D detectados.

Finalmente no capítulo **12** será descrito todo o processo de identificação a 3D baseado naquelas "Assinaturas".

8.1 Introdução

Desde o início dos trabalhos de Tese ficou claro que o sistema que se iria construir para análises a 2D não seria suficiente para atingir completamente os objectivos traçados. Sendo assim, procurou-se desde logo idealizar e esquematizar um sensor de distâncias que fornecesse a 3ª dimensão. Este sensor, aliado a todo o software de controle e processamento, iria não só permitir complementar toda a análise a 2D como também apresentar outros resultados de interesse.

Curiosamente, embora se tivesse encontrado na literatura algumas referências a possíveis sistemas de medição [Fu87], foram muito raras as referências explícitas a sistemas laboratoriais concretos, e apenas uma referência a um sistema comercial que não preenchia os requisitos impostos pelas necessidades. Só muito recentemente se encontraram ofertas de protótipos comerciais minimamente aceitáveis e com possibilidade de aquisição [3Dscanners]. Por conseguinte, logo desde o início, e paralelamente a todo o trabalho que se viria a desenvolver para 2D, procurou-se conceber um processo de adquirir as medidas de distância. A primeira ideia, que consistia no varrimento de um ponto luminoso na cena e na medida das distância com recurso a um "array" de fotodíodos, e que é referido na secção 8.2.5, não passou da fase de esquematização atendendo a todo o sistema mecânico que envolvia.

Sendo assim, abandonou-se esse projecto, embora se aponte como interessante.

Um segundo projecto com um díodo laser foi desenvolvido (descrito na secção 8.3) e levado até ao fim. Contudo, devido a características desfavoráveis associadas ao díodo Laser utilizado, foi decidido não o adoptar na implementação do sistema de medida. Todo o trabalho desenvolvido em torno dele mantém-se de qualquer forma válido, e permitirá que no futuro possa vir a reduzir em volume todo o sistema de varrimento, caso se possa adquirir um díodo Laser adequado.

O sistema que viria de facto a utilizar, diferiu bastante dos dois anteriores tanto em termos de sistema óptico como mecânico. O seu estudo e construção decorreram paralelamente ao desenvolvimento do sistema de visão a 2D, já que ele implicava o amadurecimento de ideias.

A sua concretização veio a consumir bastante tempo e esforço. De qualquer modo pode-se considerá-la bastante gratificante pelos resultados obtidos que foram

bastante positivos, não só em termos de robustez como também em termos de fiabilidade de funcionamento.

8.2 O Estado da Arte nos Sensores de Distância

Os processos utilizados são vários e podem ser classificados em activos e passivos, consoante necessitem, ou não, de interagir de uma forma activa com o ambiente.

De qualquer forma e em geral, esses sistemas artificiais construídos imitam processos usados na natureza. No caso humano, por exemplo, a distância é avaliada por triangulação através de uma visão binocular (passiva). Outros animais como os morcegos, os golfinhos e alguns cetáceos, complementam a sua visão com o envio e a recepção de impulsos ultra-sónicos em que a distância é avaliada pelo "tempo de voo" das frentes de onda (activa).

A estes processos deverá incluir-se o da luz estruturada, classe na qual o sistema que foi implementado nesta Tese se inclui.

Seguidamente farei uma descrição de vários processos.

8.2.1 Triangulação Simples

Neste processo é utilizado uma montagem como a indicada na Fig.1. Uma fonte de luz a uma distância fixa do detector projecta um ponto luminoso no objecto; esta fonte executa um movimento de rotação em torno do eixo perpendicular à linha que o une ao detector até que o ponto luminoso seja detectado (o detector está focado numa pequena área do objecto). Quando tal acontece, o ângulo θ conhecido é utilizado numa fórmula trigonométrica simples para encontrar a distância ao objecto. esta medida é apenas pontual. Para outros valores é necessário deslocar todo o sistema em duas direcções perpendiculares (da esquerda para a direita e de cima para baixo) [Jarvis83a].

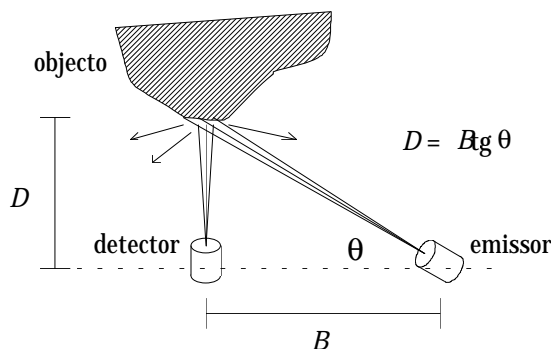


Fig. 1 - Triangulação simples.

A desvantagem deste sistema é a sua lentidão e necessidade dos seus componentes se movimentarem em conjunto no espaço para além de *per si*, ou alternativamente, do objecto, o que nem sempre é possível.

8.2.2 Triangulação por Visão Binocular (Visão Passiva)

Este é o sistema que mais se aproxima da visão humana.

Ao contrário do sistema anterior, a aquisição da informação é toda feita em simultâneo: dispõe-se de duas câmaras distanciadas de um dado valor conhecido, dirigidas para a mesma cena e geralmente paralelas (Fig.2). Cada uma dessas câmaras efectua uma aquisição de imagem em apenas alguns milésimos de segundo, ficando toda a informação registada de uma só vez e disponível para processamento imediato.

Exceptuando talvez uma eventual rectificação do ângulo formado pela duas câmaras, por meio de um motor, este sistema não requer qualquer tipo de acção mecânica o que sem dúvida constitui uma grande vantagem sobre o anterior método. Toda a acção se baseará então em efectuar correspondências entre pontos importantes nas duas imagens, dado que serão vistos de perspectivas diferentes pelas duas câmaras, eles ficarão diferentemente posicionados na matriz xy de cada uma das imagens. Conhecidas essas duas posições, assim como a distância entre câmaras e o ângulo por elas formado, é possível "descodificar" a distância a que cada ponto se encontra das câmaras (ou de qualquer outro ponto de referência) pelo uso de relações trigonométricas simples.

O mais problemático deste processo é a correspondência bem sucedida dos pontos mais importantes. Em casos de um grande número de pontos presentes ou de ocorrência de ruído, essa correspondência poderá ficar prejudicada. Por vezes, utiliza-se uma terceira câmara [Bortolozzi91] para efectuar a confirmação dessas correspondências e bem assim eliminar o efeito de ruídos, que doutra forma ficaria irremediavelmente presente.

As relações trigonométricas baseiam-se no seguinte esquema:

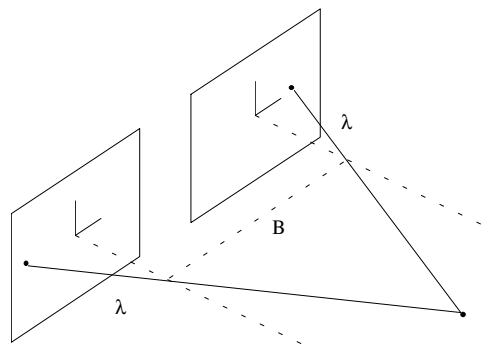


Fig. 2 - Visão stereo com 2 câmaras.

Ora, de acordo com a figura anterior,

$$\frac{X_1}{\lambda - Z} = \frac{x_1}{\lambda} \Leftrightarrow X_1 = \frac{x_1}{\lambda} (\lambda - Z)$$

do mesmo modo para x_2 ,

$$X_2 = \frac{x_2}{\lambda}(\lambda - Z)$$

sabendo que $X_2 = X_1 + B$, tem-se,

$$\begin{cases} X_1 = \frac{x_1}{\lambda}(\lambda - Z) \\ X_1 + B = \frac{x_2}{\lambda}(\lambda - Z) \end{cases} \Rightarrow B\lambda = (\lambda - Z)(x_2 - x_1) \Leftrightarrow Z = \lambda - \frac{\lambda B}{x_2 - x_1}$$

Z fica assim perfeitamente definido, dado que λ e B são conhecidos e x_1 e x_2 são mensuráveis.

Conhecido Z , as coordenadas reais (X, Y) podem ainda ser obtidas a partir de (x_1, y_1) usando a matriz de calibração da câmara 1. O posicionamento do ponto \underline{w} no mundo real fica portanto inequivocamente determinado (a correspondência fica facilitada se os pontos de interesse forem devidamente etiquetados, vértice de duas arestas, de três arestas, ponto isolado, extremo de segmento, etc.).

8.2.3 Medição das Distâncias Usando Desfasamento

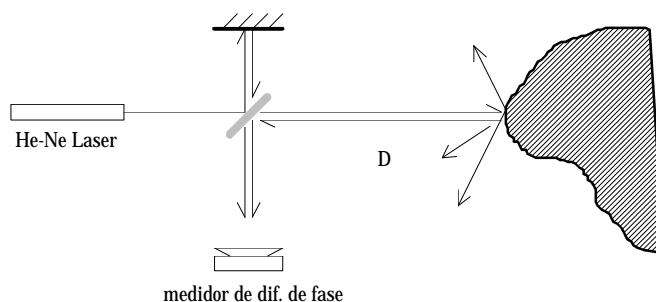


Fig. 3 - Medição do desfasamento entre 2 ondas.

Neste processo utiliza-se uma emissão contínua de luz Laser, por exemplo de He-Ne, cujo feixe é separado em dois por um "beam-splitter" como indicado na Fig.3. Um dos feixes é imediatamente dirigido ao receptor enquanto que o outro é feito incidir no objecto situado a uma distância D que se pretende conhecer.

Aquele receptor recebe assim duas ondas: uma proveniente do laser e que vai funcionar como referência, e outra obtida por reflexão difusa no objecto. Devido ao caminho óptico extra introduzido na segunda onda ($2D$) esta vai adquirir uma diferença de fase que é detectada e medida. A relação entre D e θ é a seguinte:

$$2n\pi + \theta = 2K \cdot D = \frac{4\pi D}{\lambda} \Leftrightarrow D = \frac{\theta}{2\pi} \times \left(\frac{\lambda}{2}\right) + n \cdot \frac{\lambda}{2}$$

$(n \in \mathbb{N})$

Para um dado θ existe sempre uma incerteza quanto a n visto que um caminho óptico extra múltiplo de λ (correspondente a distâncias ao objecto múltiplas de $\lambda/2$) introduz sempre um desfasamento também múltiplo de 2π , ou seja, nulo. No entanto, pode-se ajustar λ de forma a que ocorra sempre $n=0$, ou seja, o desfasamento introduzido na segunda onda seja sempre $0 < \theta < 2\pi$. Para tal é necessário arbitrar uma distância máxima a que se pretende que o sistema responda; e sendo assim a distância medida será sempre uma fracção desse valor (se distâncias maiores ocorrerem, é suposto possuírem intensidades associadas de feixe reflectido muito abaixo das normalmente esperadas e sendo assim poderão ser identificadas e rejeitadas). Por exemplo para uma distância máxima de 3m, o λ seria:

$$n = 0 \Rightarrow 3 = \frac{2\pi}{2\pi} \times \frac{\lambda}{2} + 0 \Leftrightarrow \lambda = 6\text{m}$$

situação que seria extrema e em que o desfasamento detectado seria nulo.

Atendendo agora que a velocidade da luz no ar é $c=3.10^8\text{m/s}$ teríamos uma frequência igual a:

$$f = \frac{c}{\lambda} = \frac{3.10^8}{6} = 5.10^7\text{Hz} = 50\text{MHz}$$

que é cerca de 10 milhões de vezes mais pequena que a do Laser de He-Ne ($\lambda=632,8\text{nm}$)! Ora, não se dispõe de emissores coerentes com aquela frequência (a mais próxima é a do Maser na ordem dos gigahertz ($f\sim 0,3\text{GHz}$), ainda 6.000 vezes maior.

Ela corresponde na realidade à das emissões rádio que de qualquer forma não seriam reflectidas pelos objectos e portanto o seu uso seria totalmente inútil. Existe também a questão da resolução espacial que neste caso seria extremamente reduzida e da ordem dos metros, pelo que a obtenção de uma imagem de "range" seria impossível.

Por outro lado impõe-se o uso de um Laser (p.ex. de He-Ne com $\lambda=632,8\text{nm}$) no espectro visível, dado que a sua visualização facilita o controle dos alinhamentos ópticos. Uma outra particularidade permite a obtenção de informação de alta resolução: a baixa divergência de feixe. De facto, uma divergência de apenas alguns mrad, constitui um factor essencial para se poder diferenciar pontos adjacentes, de reduzidas dimensões e a grandes distâncias. No caso do Laser de He-Ne, acresce ainda a vantagem de possuir um custo baixo.

Sendo assim, terá que se utilizar aquele feixe apenas como a transportadora da onda que realmente nos interessa, ou seja a de 50MHz, e que irá na realidade corresponder à modulação em amplitude do feixe. Sendo assim, o medidor de diferença de fase deverá primeiramente extrair o sinal modulador às duas transportadoras dos dois feixes recebidos.

Em princípio, se os dois sinais extraídos, desfasados de ϕ , tivessem a mesma amplitude, então a interferência entre eles daria:

$$\begin{cases} e_1 = A \cdot \text{sen}(\omega t) \\ e_2 = A \cdot \text{sen}(\omega t + \varphi) \end{cases}$$

$$e = e_1 + e_2 = A \cdot [\text{sen}(\omega t) + \text{sen}(\omega t + \varphi)] = 2A \cdot \left[\text{sen}\left(\omega t + \frac{\varphi}{2}\right) \times \cos\left(\frac{\varphi}{2}\right) \right]$$

Esta sobreposição das duas ondas não corresponde a uma interferência real já que se trata de dois sinais extraídos dos feixes recebidos. Aliás essa interferência nunca seria possível. Em primeiro lugar porque a diferença de amplitudes seria enorme: um dos feixes é recebido directamente do "beam-splitter" e o outro é proveniente da reflexão difusa do objecto. Em segundo lugar, porque existiria sempre o problema da coerência espacial longitudinal do Laser que normalmente não excede 1m, distância provavelmente inferior àquela a que se encontra o objecto pelo que não seria possível obter um padrão fixo de interferência. De qualquer modo a amplitude da segunda onda não teria significado já que ela depende tanto da distância do objecto como da sua reflectância. Sendo assim nunca seria possível extrair φ do sinal de interferência.

Os dois sinais teriam então que ser extraídos em separado e o segundo amplificado de forma a exhibir a mesma amplitude. Só deste modo e após mistura dos sinais se obteria o factor $\cos(\varphi/2)$ do qual se obteria φ e finalmente D .

É este o caso do sistema construído no SRI [Nitzan77] que utiliza um Laser de He-Ne com 15mW e um modulador de 9MHz. A distância máxima mensurável é de 5m. A imagem de "range" obtida possui uma resolução de 128×128 e demora cerca de 2h a ser construída. Embora o processo seja interessante, ele não poderá ser utilizado na maior parte das aplicações dado o seu grande tempo de execução. A título de comparação refira-se que no sistema desenvolvido nesta Tese, conseguiu-se uma resolução de 640×512 (20× superior) com um tempo de execução de apenas 5min (na opção de funcionamento mais morosa), ou seja cerca de 480× mais rápido!

Este método apresenta ainda outras desvantagens. Requer por exemplo um sistema de amplificação óptica do feixe recebido do objecto - o fotomultiplicador - que actua ainda antes da amplificação electrónica de sinal. Este dispositivo, que deve possuir uma grande sensibilidade, é bastante dispendioso, pesado, volumoso e extremamente frágil. Este sistema, pelas evidências apresentadas, não seria de forma alguma adequada à robótica de manipulação e montagem.

8.2.4 Medição da Distância por Tempo de Vôo ("time-of-flight range finder")

Um processo mais directo de medida de distância seria a da medição directa do tempo de retorno de um impulso ultra-sonoro. Sendo $v=340\text{m/s}$ a velocidade do som no ar e D a distância a que se encontra um obstáculo, então o "tempo de vôo" será de $2D/v$. Para distâncias da ordem dos metros teremos tempos na ordem dos centésimos de segundo, perfeitamente mensuráveis usando a electrónica convencional. O uso de ultras-sons, contudo, carece de resolução espacial uma vez que a divergência do feixe pode

atingir os 60°. De qualquer modo, por muito que se aperfeiçoasse os emissores, não seria nunca possível "resolver" dimensões abaixo do comprimento dos ultra-sons. Por exemplo para uma frequência de 50KHz o comprimento de onda seria de 6,8mm, que não seria de forma alguma aceitável para uma imagem de "range".

Sendo assim, a sua aplicação fica limitada à detecção de obstáculos, por exemplo, na robótica móvel.

Para a construção de imagens de "range" utilizáveis já no domínio da robótica de manipulação, há que recorrer de novo a emissões coerentes de luz e de preferência dentro do espectro visível (a opção dos 632,8nm do Laser de He-Ne é a normalmente escolhida). A medição do tempo de vôo torna-se no entanto mais crítica neste caso uma vez que os tempos envolvidos são da ordem dos pico-segundos, cujo tratamento é impossível com a electrónica convencional. Mais concretamente, para uma incerteza de cerca de 5mm na distância medida, requiere-se uma precisão na ordem dos 50ps [Jarvis83a]. Todos os circuitos utilizados deverão obedecer a este condicionalismo, ou seja, possuir respostas rápidas desta ordem de grandeza. O preço de tais circuitos é proibitivo e de difícil acesso no mercado. Para além da aquisição de tais circuitos de electrónica rápida, havia que construir toda uma aparelhagem de medida adequada, que detectasse o sinal de retorno, filtrasse, ampliasse e finalmente medisse. Tudo isto implicaria um trabalho volumoso que poderia por em risco os trabalhos de verdadeiro interesse na Tese, já que ele sairia fora do âmbito destes. Implicaria ainda a aquisição e montagem de um Laser pulsado e respectivo controle assim como de um fotomultiplicador de alta sensibilidade para detectar e pré-amplificar o impulso luminoso reflectido (tal como no sistema anterior). Tanto o grande volume de trabalho que isto implicaria, como os custos proibitivos de tal montagem e ainda as grandes desvantagens que este processo partilha com o anteriormente descrito, fizeram com que este se fosse rejeitado nesta Tese.

A literatura pode mostrar, contudo, que alguns protótipos foram já conseguidos. Por exemplo, no Jet Propulsion Laboratory, um Laser de arsenieto de gálio, pulsado, com emissão na zona do infra-vermelho ($\lambda=840\text{nm}$) foi utilizado conjuntamente com um fotomultiplicador de alta sensibilidade. Um sistema mecânico de varrimento ligado a um espelho projectava o feixe em diversos pontos da área a analisar e simultaneamente recebia o reflexo do impulso emitido, coaxialmente. A resolução conseguida nas imagens de "range" era de 128×128 e demorava cerca de 3min a ser obtida.

Mais recentemente, para além dos aparelhos de electrónica rápida, foi introduzido o uso de transputers para um também rápido processamento computacional. Este melhoramento veio reduzir drasticamente o tempo de obtenção de uma imagem de "range" para as décimas de segundo, permitindo assim análises em tempo real. O projecto, desenvolvimento e implementação de tal dispositivo de alta tecnologia absorveria concerteza um considerável número de investigadores, dando contributos nas várias áreas diferentes de conhecimento envolvidas.

8.2.5 Medição da Distância com um "Array" Linear de Fotodíodos (Triangulação)

Neste sistema é utilizada uma fonte de emissão luminosa coerente, de preferência no espectro visível (p.ex. um díodo laser + uma lente colimadora para focalizar o feixe num ponto da cena, ou um laser de He-Ne) e um "array" linear de fotodíodos e constitui a 1ª escolha de implementação que no entanto viria a ser abandonada em favor da triangulação.

O funcionamento deste sistema é o seguinte: a fonte de luz projecta um ponto luminoso na cena de que resulta uma reflexão difusa (desde que o objecto ou área atingida não seja espelhada). Parte dos raios ópticos resultantes dessa reflexão atravessam então uma lente colocada paralelamente ao emissor que os faz convergir num plano de formação de imagem (de acordo com a lei dos focos conjugados) onde se encontra um "array" linear de fotodíodos (Fig.4). A imagem formada, ainda que eventualmente desfocada, será obviamente um outro ponto luminoso, imagem do primeiro, que é projectado no "array". Um fotodíodo registará então um máximo de intensidade e a sua posição no "array" dará a medida indirecta da distância D que se pretende conhecer.

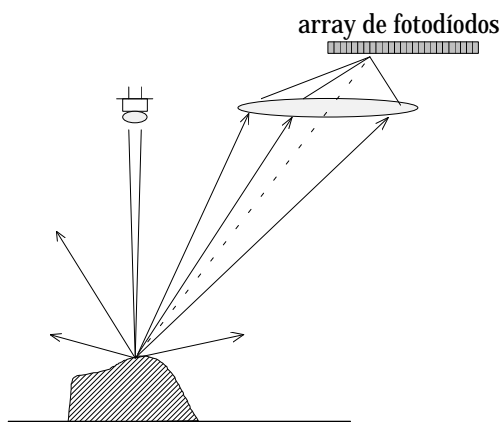


Fig. 4 - Modo de funcionamento do sistema óptico.

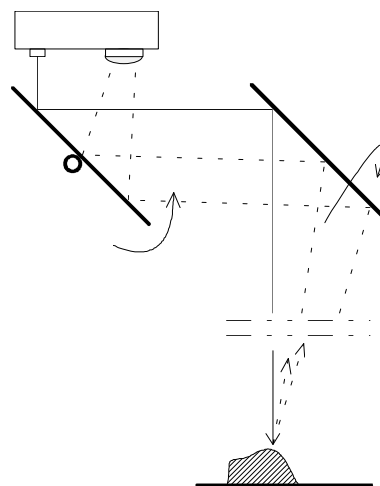


Fig. 5 - Esquema do sistema mecânico responsável pelo varrimento do feixe.

Como se poderá verificar através da figura anterior, quanto mais próximo estiver o objecto, mais longe do ponto A se irá formar a imagem do ponto luminoso. Se d for a distância entre as lentes e f for a distância da lente 2 ao "array" (poderá ser diferente da distância focal) e Δx for a espessura de cada fotodíodo, então D será dado por:

$$\frac{d}{D} = \operatorname{tg}\theta = \frac{x}{f} \Leftrightarrow D = \frac{d \cdot f}{x} = \frac{d \cdot f}{i \cdot \Delta x}$$

onde i é a posição do fotodíodo com o máximo de intensidade.

O facto dos fotodíodos possuírem uma largura finita introduz forçosamente uma imprecisão no resultado. E embora ela possa ser atenuada fazendo uma análise da intensidade recebida nos fotodíodos vizinhos por forma a posicionar o máximo entre

dois centros de fotodíodos e apresentar um i decimal, o facto é que existe sempre um erro de posicionamento em x derivado do referido carácter descontínuo do "array".

No caso mais desfavorável i é inteiro e o erro é de $0,5 \times i$ o que para um valor de $i \gg 1$, dá um erro de:

$$D = \frac{d \cdot f}{i \cdot \Delta x} \Rightarrow \frac{\Delta D}{D} \approx \frac{\Delta i}{i}$$

Se $i=30$ (p.ex. num "array" de 64 fotodíodos), $\Delta D/D \approx 0,5/30 = 0,017 \rightarrow 2\%$, o que para um $D=1,5\text{m}$ dará uma imprecisão de $\sim 3\text{cm}$. O erro relativo de D varia obviamente com o valor de i , o que significa que para grandes distâncias (i pequeno) o sistema de medida perde fiabilidade. De notar que o erro absoluto que já aumentava com a distância ($\Delta D = D \cdot \Delta i / i$) virá ainda agravado com o facto de $\Delta i / i$ não ser constante e aumentar drasticamente com a diminuição de i .

Uma possível solução para minimizar esse erro seria o de escolher um "array" com maior número de fotodíodos (p.ex. 256) embora isso pudesse elevar os custos de montagem exponencialmente. Por outro lado, as distâncias d e f teriam que ser cuidadosamente escolhidas de acordo com uma gama pré-definida de possíveis valores de D que não poderia ser muito larga. De preferência deveria também poder-se escolher a largura Δx dos fotodíodos (o que na realidade não é possível) de modo a associar os pontos extremos do "array" às distâncias limites dessa gama. Desta forma explorar-se-ia ao máximo a extensão física do "array" minimizando assim o erro relativo $\Delta i / i$.

A imagem completa de distâncias seria obtida por varrimento do ponto luminoso com um sistema de espelhos rotativos tal como apresentado na Fig.5 (exemplos deste tipo de sistema podem ser encontrados em [Nitzan88] e em [Rioux89]).

Das grandes desvantagens deste sistema saliente-se a necessidade de alguma mecânica com peças móveis que implica uma construção cuidada impossível de executar sem o devido apoio oficial e um controle electrónico não muito imediato. O seu tamanho pode também revelar-se incómodo, mas mais grave que isso poderá ser de uma robustez bastante duvidosa. Para além de tudo isso acresce a necessidade determinar com rigor a posição i para a qual ocorra um máximo de intensidade o que pode não ser um processo fácil principalmente se houver uma luminosidade ambiente apreciável.

Numa variante deste processo, [Marszalec92] utiliza um "array" de Leds associado a uma lente colimadora em substituição do díodo laser. A activação sucessiva daqueles Leds direcciona o feixe luminoso para várias direcções ao longo de uma linha na cena, simulando assim o processo de rotação mecânica. Este sistema que encontra aplicações satisfatórias em robótica móvel onde o varrimento linear é suficiente, só seria aplicável em robótica de manipulação se o referido "array" fosse substituído por uma matriz de Leds que, a existir no mercado, teria um custo inabarcável. Para além disso, dada a impossibilidade de focagem do feixe (o sistema óptico estaria ligado rigidamente à matriz tal como acontece com o do "array"), não seria possível obter boas resoluções na imagem de alturas adquirida por este processo (não havendo sequer controle no seu valor), tornando-o assim totalmente inadequado.

Sendo assim e embora estes sistemas possam apresentar alguns atractivos, não foram considerados para desenvolvimento e montagem tendo a minha preferência ido para outro sistema mais robusto e de medições mais rigorosas.

8.2.6 Medição da Distância através da Projecção de n Linhas Luminosas Paralelas (Triangulação)

O mesmo princípio da triangulação pode ser utilizado na projecção de um padrão de luz conhecido, por exemplo n linha paralelas. Como resultado da sua projecção num objecto, aquele padrão irá ser deformado de acordo com a sua topografia, ou seja, com as várias alturas pontuais em toda a superfície atingida pela luz.

Se for possível associar univocamente cada troço de linha deformada com a remanescente projectada directamente na mesa de suporte onde o objecto se encontra, então poder-se-á, por triangulação, medir a altura a que se encontram todos os pontos do objecto atingidos por aquela luz estruturada. Aquela associação, contudo, nem sempre é possível como será visto em 8.4.1. A grande vantagem deste processo é dispensar peças móveis e o respectivo controle. Uma aplicação deste processo pode ser encontrada em [Will71].

8.2.7 Medição da Distância através de um Gradiente de Intensidade Luminosa

A dificuldade de correspondência referido em 8.2.6 pode ser ultrapassada se for projectada na cena um padrão contínuo de luz com um dado gradiente fixo [Silva94]. Supondo que a reflectância de objectos e fundo é igual, é então possível fazer as correspondências já referidas e necessárias ao cálculo das distâncias.

Em situações reais aqueles valores de reflectância vão variar o que inviabiliza o estabelecimento de correspondências, a menos que ela possa ser determinada para cada ponto da imagem. Isso pode ser conseguido por projecção posterior de um padrão contínuo de luz de intensidade uniforme. O quociente entre os valores de intensidade obtidos nestas duas situações para cada ponto da cena, irá então permitir estabelecer as correspondências.

Se a experiência for executada em condições de luminosidade ambiente não nula, há ainda que proceder à aquisição de uma 3ª imagem, cujos valores de intensidade deverão ser subtraídos à da 1ª antes de se proceder ao cálculo dos quocientes com as da 2ª.

Para que este processo seja eficiente, há necessidade de garantir uma boa qualidade dos padrões de luz utilizados. A utilização de um diapositivo que possua uma variação gradual de transmitância numa das direcções é capaz de conferir à luz projectada o referido gradiente; mas isso requer a utilização de um projector de diapositivos que possui um volume apreciável. Para além disso só consegue iluminar pequenas regiões e a curta distância o que, por exemplo, não vai de encontro aos requisitos do trabalho desta Tese. Em algumas aplicações, este processo não poderá, portanto, ser adoptado.

8.2.8 Medição da Distância através da Projecção de uma Linha Luminosa com Varrimento (Triangulação)

Este processo, que foi o adoptado para implementação, apresenta vantagens relativamente aos apresentados em 8.2.5 e 8.2.6. Em relação a este último levanta a ambiguidade de correspondência entre pontos do objecto iluminados pelo feixe e as respectivas linhas de referência não sujeitas a deformação, pois existe apenas um plano de luz e por conseguinte uma só linha de referência. Em relação ao penúltimo, ele diminui o número de graus de liberdade de 2 para 1, simplificando consideravelmente quer a construção quer o seu controle, para além de conferir a todo o sistema uma maior robustez física.

Como se verá adiante, este processo é ainda totalmente imune à luminosidade ambiente, especialmente se, como foi o caso da minha implementação, for acrescentada à câmara um filtro vermelho que atenua a essa luminosidade em favor da luz laser de $\lambda = 632,8\text{nm}$. Ele é ainda especialmente adequado a cenas de dimensão apreciável como o são as dos ambientes de robótica de manipulação.

Um sistema semelhante ao desenvolvido, embora difira bastante na tecnologia utilizada, na aplicação e nos requisitos de resolução (muito inferior àquele conseguido no nosso sistema), pode ser encontrado em [Gruss92].

8.2.9 A Discriminação de Pontos por Variação do Plano de Observação da Imagem

Qualquer ponto do espaço situado à frente de uma lente convergente, vai ter a ele associada uma imagem real (desde que a sua distância à lente não seja inferior à distância focal, circunstância em que a imagem seria virtual) projectável num plano a uma distância dada pela equação dos focos conjugados: $1/s_o + 1/s_i = 1/f$ (onde s_o é a distância objecto e s_i a distância imagem). No caso de ser possível a focagem do ponto pelo ajuste de s_i , então, conhecido f , s_o seria imediatamente calculável. Se o plano de observação não for o plano da imagem, como acontecerá no caso de haver mais do que um ponto a medir e não for possível ajustar s_i para todos eles, as respectivas imagens não virão focadas, mas a sua distância lateral ao centro óptico será mensurável.

Nesta técnica, são obtidos dois conjuntos de imagens de um conjunto de pontos existentes na cena por alteração da distância do plano de observação das imagens à lente convergente. Para cada um deles são medidas as duas distâncias laterais ao eixo óptico, cuja diferença é designada por discrepância. Este valor dá uma medida directa da distância a que cada ponto se encontra da lente [Andersen91] [Siegel92] [Dias92].

Neste processo é exigível a correcta correspondência de pontos nas duas imagens obtidas por alteração da distância focal o que nem sempre é possível. Para além disso, a medida da disparidade é dificultada no caso de pontos distantes ou próximos do eixo óptico, devido à redução drástica do seu valor. Os erros na medição da distância são por isso variáveis, atingindo valores eventualmente inaceitáveis naquelas situações particulares.

8.2.10 Análise de Textura

A textura pode ser uma boa fonte de informação sobre a forma de uma determinada superfície. Se houver uma densidade superficial constante dos elementos que a constituem, então o gradiente dessa densidade medido na imagem, dá a inclinação dessa superfície relativamente à câmara.

Se por outro lado for conhecida a forma dos elementos, então a análise da sua imagem distorcida dá a informação da orientação local da superfície. Por exemplo, se a textura for composta por círculos, então a sua imagem será um conjunto de elipses. O comprimento relativo e inclinação dos eixos medidos em cada elemento, fornecerá então a pretendida inclinação local [Shirai83].

Embora as distâncias a que se encontram os elementos da câmara possam ser calculadas a partir da distância fixa entre eles e das várias inclinações, elas envolvem um processo de integração que não permite fornecer os seus valores independentemente um dos outros e de uma forma imediata. Para além disso este processo só funciona obviamente para superfícies adequadamente texturadas.

A sua aplicação em Robótica de Manipulação, especialmente para ambientes não completamente estruturados, fica portanto completamente posta de parte.

8.3 O Díodo Laser TOLD 9200

8.3.1 Descrição do Funcionamento do Circuito de Alimentação

Como fonte para o sistema de iluminação estruturada foi primeiramente considerado um díodo laser: o díodo TOLD 9200 da Toshiba. Para além de apresentar uma saída de luz razoavelmente coerente no comprimento de onda visível com $\lambda = 670\text{nm}$, as suas pequenas dimensões faziam prever uma fácil instalação e manuseamento. Para que a sua emissão luminosa atingisse os 3mW, bastava que a corrente polarizadora que o alimentava atravessasse o limiar dos 80mA (para uma temperatura de 25°).

Dados os requisitos de funcionamento do díodo laser TOLD 9200, foi adoptada uma fonte de corrente estabilizada e ajustável para os dois valores extremos de corrente pretendidos: 25mA e 100mA.

O circuito desenvolvido e apresentado na Fig.6 consta de duas partes distintas: uma consiste na alimentação primária, ou seja, na transformação de tensão da rede (220V~) para 12V alternos, rectificadas a onda completa com uma ponte de díodos e filtrada por um condensador de 1mF que garante o fornecimento de corrente ao circuito posterior quando a tensão à saída daquela ponte baixa de $\sqrt{2}.12\text{V}$ para 0V. Obtém-se assim um fornecimento de tensão contínua e quase sem flutuação. O condensador de 100nF é ainda incluído para eliminar picos de tensão assim como um fusível para 125mA para protecção. A tensão de saída é do tipo da apresentada na Fig.7, embora a flutuação na tensão esteja exagerada para fins demonstrativos.

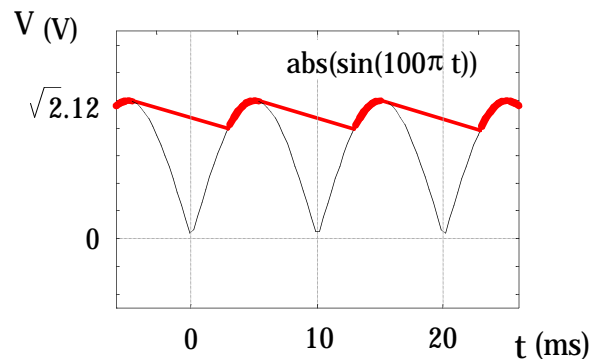


Fig. 7 - Rectificação de onda completa na alimentação do díodo laser.

A 2ª parte do circuito utiliza já os $\sqrt{2.12V}$ contínuos, garantidos pela parte descrita atrás, e consiste essencialmente num transistor (BD 137) em cujo colector é colocado o díodo laser (Fig.6). O funcionamento do laser é garantido pelo fluxo de corrente no colector. Atendendo a que se pretende um funcionamento estabilizado e ajustável, utilizou-se uma malha de realimentação como ilustra a figura, cujos elementos principais são o amplificador operacional 741, díodo Zener para 3,3V e a resistência variável de 100Ω colocada no ramo entre o emissor do transistor e a terra. O seu funcionamento é o seguinte: em regime linear do 741, o díodo Zener impõe uma tensão na entrada 2 igual à da 3 atendendo ao seu elevado factor de amplificação. Esse valor (3,3V) deverá também ser a tensão no emissor do transistor dada a ligação deste ponto àquela entrada 2 através da resistência de $10K\Omega$ (ramo de realimentação). As resistências R_1 e R_2 são então escolhidas de acordo com os limites de corrente pretendidos já que $I_C \approx I_E$ (correntes respectivamente no colector e no emissor). São eles,

$$I_{\min} = 3,3 / (100+33) = 25\text{mA} \quad \text{e,}$$

$$I_{\max} = 3,3 / 33 = 100 \text{ mA}$$

A corrente que atravessa o laser fica assim definida pelo valor de tensão do díodo Zener e da resistência no emissor do transistor.

Qualquer flutuação nos valores de tensão à entrada do 741 produz um efeito que a contraria, uma vez que existe a malha de realimentação negativa e portanto aquele nunca atinge a saturação quer para $+V_{cc}$ quer para $-V_{cc}$. Por exemplo, um ligeiro aumento na tensão de entrada 3 (V_{in}^+) produz um aumento da corrente de base e igualmente portanto na corrente do emissor. Esse aumento faz elevar a tensão do emissor que é realimentada no 741 (entrada 2 referente a V_{in}^-) compensando assim o referido aumento de V em 3, e mantendo o regime linear do 741. Esta realimentação garante que o emissor se encontre sempre à tensão de 3,3V determinados pelo díodo Zener, estabilizando assim a corrente que alimenta o laser. O ajuste da corrente é efectuado no potenciómetro R_2 .

São ainda inseridos dois díodos extra (D_2 e D_3) no circuito. O Led D_2 destina-se a indicar o seu funcionamento enquanto que D_3 se destina a eliminar tensões momentâneas com polarização inversa no laser que o poderiam danificar.

A Fig.8 mostra a fonte de alimentação na sua versão final.

Fig. 6 - Esquema do alimentador e estabilizador do díodo laser.

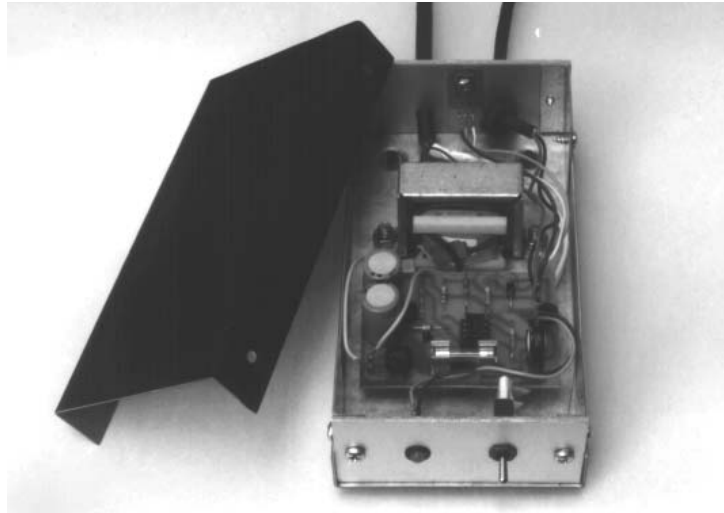


Fig. 8 - Montagem final da fonte de alimentação do díodo laser.

Finalmente, o díodo laser é ligado a uma placa de alumínio como a mostrada na Fig.9, com o duplo objectivo de permitir uma boa dissipação de calor e de conferir ao conjunto uma razoável robustez.

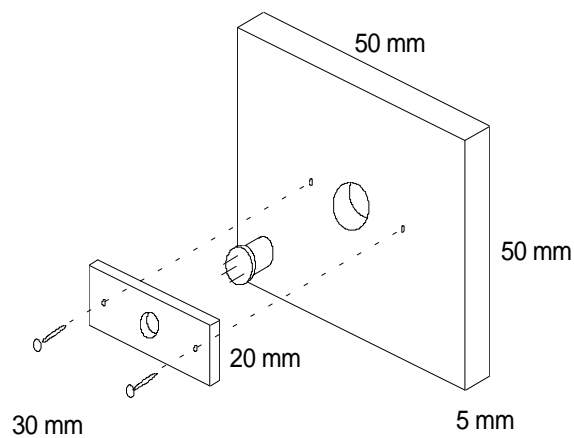


Fig. 9 - Montagem robusta do díodo laser na placa dissipadora.

8.3.2 A Viabilidade do Uso do Díodo Laser TOLD 9200

Ao contrário do que fora suposto aquando da encomenda deste laser, ele não apresenta um feixe de saída colimado, possuindo inclusivé divergências angulares de feixe diferentes nas duas direcções principais perpendiculares à direcção de emissão. Os valores são aproximadamente de 7° e 34° nessas direcções. A possibilidade de colimação por um sistema óptico externo foi posta de parte pela complexidade de elementos e suportes que envolveria e que implicaria grande volume e baixa robustez. Eficiente seria um colimador de reduzidas dimensões ligado já de origem ao laser pelo que seria necessário nova aquisição (e a consequente rejeição do TOLD 9200). Deste modo a fonte de alimentação desenvolvida manteria a sua utilidade. Foi, contudo, considerado que os 3mW de potência eram insuficientes para fazer face às grandes distâncias existentes na montagem experimental, da ordem dos metros (qualquer pequena divergência do feixe, sempre presente apesar do uso de uma lente colimadora externa, originará sempre um espalhamento da luz na cena pelo que as zonas de incidência poderiam, dada aquela baixa potência, não ser salientadas contra o fundo e portanto não ser registadas na imagem captada pela câmara). Ainda, o uso de elementos refractivos na montagem, produziam uma perda de intensidade por reflexão nas várias interfaces.

Foi portanto rejeitada de todo esta opção e adoptou-se um laser de He-Ne de 30mW, colimado, a funcionar para $\lambda=632,8\text{nm}$. A fonte desenvolvida mantém contudo a sua utilidade em aplicações futuras que venham a envolver o uso de um díodo laser.

8.4 O Sistema Óptico

8.4.1 A Luz Estruturada

O varrimento de um ponto luminoso em toda a cena envolveria um sistema mecânico com dois graus de liberdade. Um tal sistema seria decerto volumoso, de controle complexo e possuiria baixas estabilidade e robustez. Estas características aliadas ao grande tempo de aquisição de dados que tal varrimento implicaria, fizeram com que esse sistema fosse posto de lado.

O outro extremo no uso de luz estruturada seria a projecção imediata e simultânea de um padrão de linhas paralelas em toda a cena. Este sistema possui a vantagem de não possuir peças móveis pelo que não existe qualquer controle a ser efectuado. Para além disto simplificar o seu desenho, o tempo de aquisição das distâncias seria também muito mais reduzido. Contudo, a imposição de uma dada densidade nas medidas de distância provocaria sobreposições múltiplas de linhas de referência e linhas desviadas por incidência em objectos. Assim, uma dada linha desviada não poderia fornecer valores de distância já que não seria possível estabelecer a relação de correspondência com a linha de referência respectiva. Tal ambiguidade

impede a concretização do processo de medidas de distância. O único processo de evitar tal ambiguidade seria conhecer *a priori* a altura máxima existente na cena e escolher um espaçamento entre linha que evitasse a sobreposição. Se por um lado isto seria demasiado restritivo já que não existe tal conhecimento *a priori*, por outro importaria um espaçamento entre linhas incompatível com a densidade de medidas pretendida. Por último, é de referir que a projecção de linha igualmente espaçadas poderá envolver um sistema de alguma complexidade e volume, pelo que a sua construção seria difícil e o seu uso impraticável.

O sistema escolhido corresponde a uma solução intermédia entre as duas referidas atrás: projecção de apenas um linha por cada aquisição de imagem, com a efectuação de medidas de altura ao longo dela e varrimento dessa linha na direcção perpendicular. Um tal sistema possui então apenas um grau de liberdade o que simplifica quer a sua construção quer o seu controle e as medidas não estão sujeitas a qualquer tipo de ambiguidade.

8.4.2 O Sistema Óptico Implementado

O esquema do sistema construído é o apresentado na Fig.10. Ele é constituído por uma fonte laser de He-Ne com 30mW de potência e divergência de feixe inferior a 2mrad, um espelho orientável a 3D (Fig.14) para direccionar o feixe para o espelho rotativo (Fig.12), uma lente esférica com a distância focal $f=1,5m$, uma lente cilíndrica e o espelho rotativo já referido cujo movimento, necessário para fazer varrer a linha em toda a zona de trabalho, é garantida por um motor passo-a-passo (Fig.11).

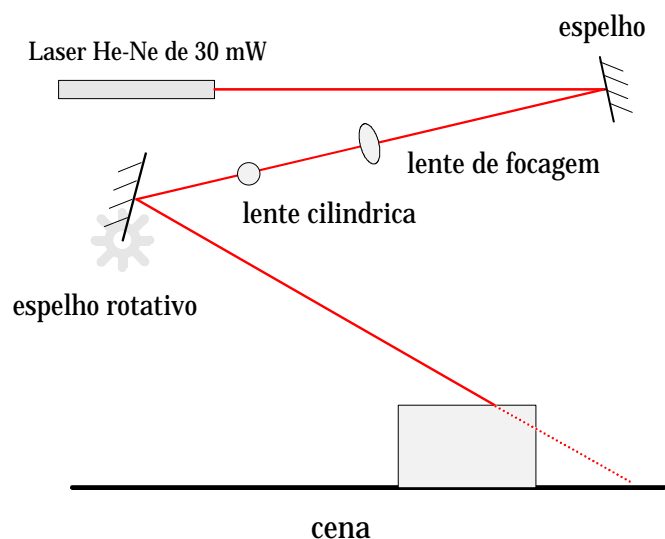


Fig. 10 - Desenho esquemático do sistema óptico implementado.

Apesar da pequena divergência de feixe, este, após o trajecto de cerca de 3m desde a fonte laser até à zona de trabalho, adquire já um diâmetro não desprezável de cerca de 7mm. A lente esférica incluída destina-se precisamente a focá-lo para a cena.

A lente cilíndrica, por seu lado, provoca o efeito contrário, ou seja, a divergência do feixe mas apenas numa direcção, obtendo-se assim um plano de luz coerente cuja intersecção com o plano de trabalho é precisamente a linha laser pretendida.

O motor, ligado ao 2º espelho, e que será descrito numa secção adiante, garante que aquela linha percorra toda, ou parte da cena.

8.4.3 A Geometria da Zona de Trabalho e suas Consequências

A instalação de todo aquele sistema foi dificultada pela geometria do espaço ocupado quer pelo robot quer pela mesa de trabalho onde os objectos a manipular se iriam encontrar. Aqueles situavam-se num espaço delimitado por uma armação de forma paralelepipedica, não alinhada com aqueles, e constituída por 4 barras verticais e 3 conjuntos de 4 barras horizontais. Cada um daqueles conjuntos formava um rectângulo horizontal situado a uma dada altura do solo.

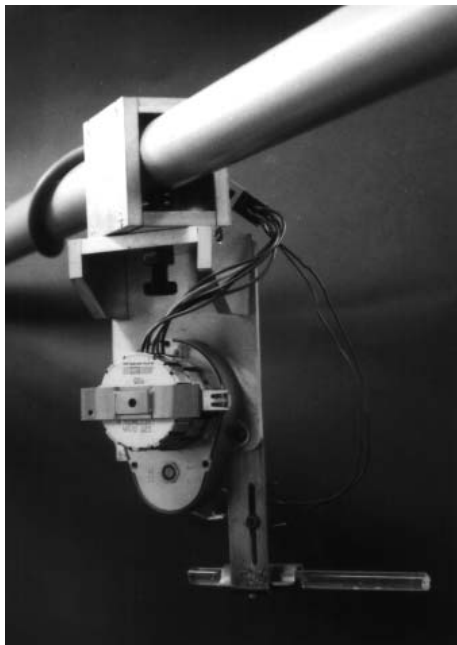


Fig. 11 - O dispositivo responsável pelo varrimento ("scanner"); perspectiva do motor passo-a-passo

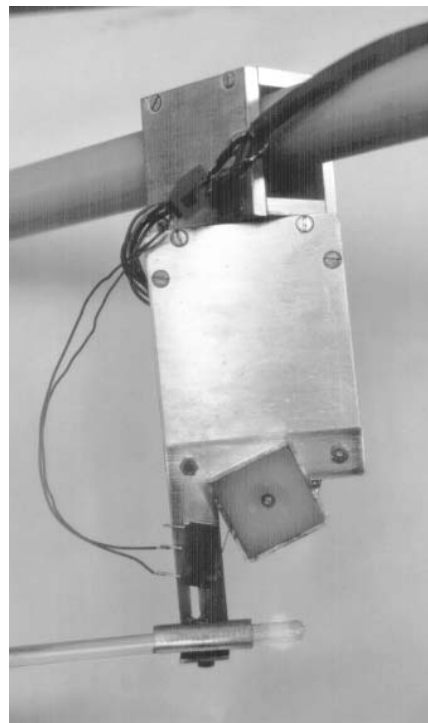


Fig. 12 - O dispositivo responsável pelo varrimento ("scanner"); perspectiva da lente cilíndrica e do espelho rotativo.



Fig. 13 - A câmara utilizada e o respectivo suporte orientável.

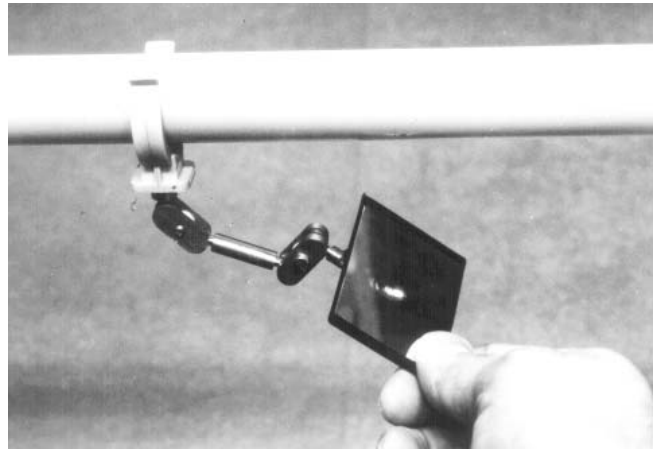


Fig. 14 - Espelho orientável em 3D.

Como se pode ver nas Figs. 11 e 13, quer a câmara vídeo quer o motor necessitaram de um suporte especialmente desenhado para permitir um alinhamento por rotação destes elementos diferente da direcção das barras onde eles se fixavam.

A colocação final de todos o elementos é a seguinte: o laser encontra-se na horizontal fixado a uma das barras horizontais inferiores da armação. Nessa mesma barra e alinhado verticalmente com a lente esférica e o espelho rotativo é colocado o espelho orientável. Aqueles dois elementos encontram-se respectivamente na barra horizontal intermédia e na superior. A localização do espelho rotativo permite incidências do feixe na mesa de trabalho em torno dos 45° . A colocação da lente cilíndrica perto do espelho rotativo (e imediatamente antes deste em termos de trajectória do feixe) foi a única possível dada a divergência imediatamente provocada na luz laser.

8.5 O Sistema Mecânico

8.5.1 O "Scanner" e a Opção do Motor Passo-a-passo

Já de posse da versão final do suporte mecânico para o motor e a lente cilíndrica, foram feitas experiências com vários motores DC e AC, mas cedo se tornou claro que a única opção aceitável seria a de um motor passo-a-passo.

Esta escolha deveu-se a vários motivos. Em primeiro lugar havia que reduzir drasticamente a velocidade angular do espelho rotativo. Tal foi conseguido com uma caixa de desmultiplicação ligada ao motor. Porém, verificou-se que a aquisição de

imagem de uma linha laser em movimento provocava efeitos de tremura. Tal efeito devia-se ao facto do registo de intensidade nos vários elementos discretos da imagem em memória, ou pixels (a partir dos valores de tensão recebidos por cada elemento do "array" de fotodíodos de que é constituído o sensor da câmara CCD utilizada) não ser simultâneo; para além de que se verificava o registo primeiramente das linhas ímpares e só posteriormente das pares, Ora, existindo um intervalo de tempo fixo entre cada conjunto de linhas (pares e ímpares), torna-se claro que eles iriam dar uma imagem da mesma linha em dois instantes diferentes, entre os quais a linha se deslocara lateralmente. A mostra simultânea dessas duas imagens provocava o referido efeito de tremura observado. A esta constatação seguiu-se a escrita de um "filtro" que eliminava as linhas ímpares; mas não resolveu a necessidade de provocar deslocamentos constantes da linha laser entre cada duas aquisições de imagem, nem tão pouco permitiu que esse deslocamento fosse ajustável e independente do tempo de medida de alturas para cada imagem.

Sendo assim, as soluções anteriores foram preteridas em favor do motor passo-a-passo que, por permitir o seu controle por software, permitia também o seu funcionamento alternado com o processo de medida; ou seja, o incremento da linha laser só seria efectuado após terminado todo o processo de cálculo e registo em memória das alturas, pelo que se conseguia sempre obter um deslocamento lateral aproximadamente constante entre cada duas aquisições. O problema do desfaseamento temporal entre a aquisição das linhas pares e ímpares ficou também ultrapassado dado que no instante de registo da imagem da linha esta se encontrava sempre imóvel.

8.5.2 O Motor Passo-a-passo, a Caixa de Desmultiplicação e o Controlador SAA1027

Das soluções anteriores apenas foi adoptada a caixa de desmultiplicação ou redutora. Esta, quando acoplada ao motor permite reduzir num determinado factor o ângulo de rotação. Esta redução é essencial para se obter deslocamento pequenos da linha laser para cada sinal de incremento enviado ao controlador do motor. A escolha daquele factor permite ainda ajustar o deslocamento para um valor pretendido e depende do seu valor de ângulo de rotação mínimo.

O motor adquirido foi um Saia de 4 fases, alimentado a 12V e com um ângulo de rotação mínimo de $7,5^\circ$. Para estes valores, o deslocamento mínimo da linha laser na cena, é de $\sim 7,5 \times 1,70 \times \pi / 180 / \text{sen}(45^\circ) \sim 0,3\text{m}$, ou seja, demasiado elevado para ter alguma utilidade. Contudo, se ao motor for acoplado um redutor de $500\times$, o deslocamento lateral passa a $0,6\text{mm}$ o que já é mais do que suficiente para permitir obter medidas densas de alturas.

Foi também adquirido um circuito integrado SAA1027 para controle imediato do motor. A sua função é gerar combinações binárias de tensão de 12V nas 4 fases, numa sequência tal que induza incrementos elementares de apenas $7,5^\circ$, tal como especificado atrás. O esquema de ligações encontra-se na Fig.15.

Finalmente foi incluído um interruptor de pressão na armação do "scanner" com o objectivo de detectar o ponto em que o espelho atinge a posição de referência ("home position") quando é efectuado um movimento de rotação invertido.

8.5.3 O Circuito Associado ao Controlador; os Sinais de Saída e de Entrada

Aquele circuito integrado por si só não é suficiente para efectuar o controle do motor. Atendendo a que ele será efectuado por software, ou seja, pelo computador, há pelo menos a necessidade de desacoplar os dois sistemas eléctricos para que um mau funcionamento de qualquer espécie no circuito exterior não danifique o hardware do computador. A solução implementada foi a de utilizar desacopladores ópticos 4N33, que como o próprio nome indica, efectua a transmissão do sinal binário entre os dois circuitos pela emissão luminosa através de um Led e um fototransistor, ambos incluídos naquele circuito integrado. Estes circuitos resolvem também o problema de ambos os circuitos funcionarem a tensões diferentes: o do motor (externo) utiliza 12V enquanto que o computador utiliza 5V. Estes circuitos foram utilizados para todos os sinais trocados. São eles o sinal de incremento ou "clock", o inicializador ou "reset", o controle do sentido de rotação e o sinal de fim-de-curso referido na secção anterior. Os três primeiros sinais são efectuados no sentido computador→controlador e o último no sentido inverso. São portanto 4 sinais para os quais são utilizados os 4N33. As Figs. 16 e 17 apresentam o esquema de um deles e o esquema de ligações. A Fig.18 mostra a montagem efectuada.

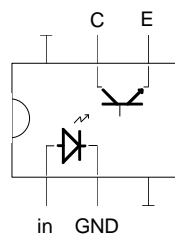


Fig. 16 - Esquema e "pin-out" de um acoplador óptico 4N33.

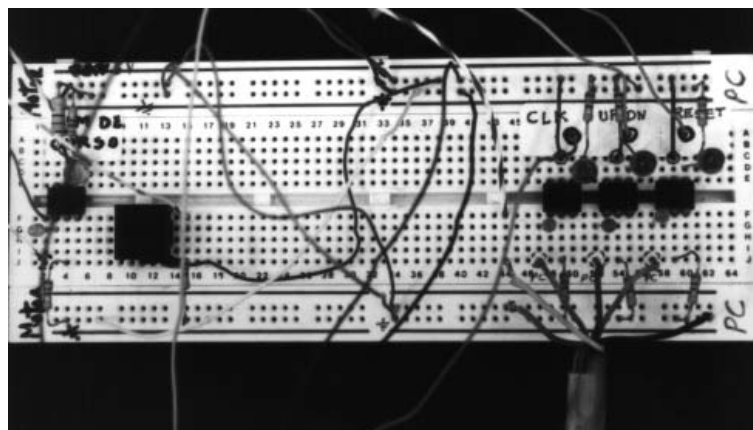


Fig. 18 - Fotografia dos acoplamentos ópticos.

O cabo mostrado na parte inferior direita transporta os sinais enviados de e para o computador.

Fig. 15 - Esquema de ligações do controlador SAA1027.
F1 a F4 são as ligações às 4 fases do motor passo-a-passo.

É de referir ainda que foi utilizado um comparador construído a partir de um amplificador operacional 741 para o sinal de clock à saída do respectivo 4N33 uma vez que a inclusão deste elemento provoca ruído no sinal. O valor de tensão para comparação é de 6V conforme a Fig.19.

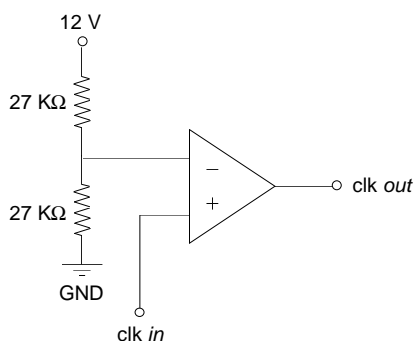


Fig. 19 - Comparador com um 741 para filtrar o sinal de clock.

Foi também incluído um inversor binário (c.i. 4011 - quad 2-input NAND) para inverter o sinal de "clock" enviado ao controlador SAA1027 uma vez que o controle deste é "active low".

Embora o software saiba em qualquer instante a "posição" em que o motor se encontra por armazenamento do nº de impulsos de clock enviados numa variável interna (*counter*), o sinal de fim-de-curso é essencial para proteger o hardware de algum mau controle daquela variável.

Foi ainda construído um contador BCD de 4 dígitos para contagem e mostra ("display") do nº de incrementos enviados pelo computador ao controlador do motor. Este contador efectua a contagem nos dois sentidos para corresponder à possibilidade real do motor poder rodar no sentido inverso. Mais uma vez, embora haja um detector de fim-de-curso para além da manutenção em memória do nº de incrementos (com os valores negativos também contabilizados), apresenta-se útil o controle visual da evolução da rotação do motor. As figuras seguintes (20, 21 e 22) ilustram não só o esquema de ligações e circuitos integrados usados (são eles o contador CMOS 4029, o conversor BCD-7segm 4511 e o "display" de 7segm LA6780), como também o circuito nas suas fases experimental e final.

Fig. 17 - Esquema de ligações dos 4 c.i. 4N33 utilizados.

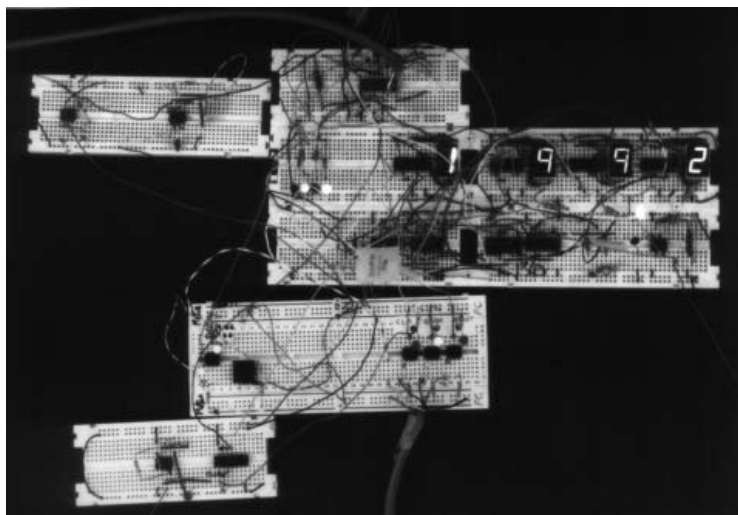


Fig. 21 - Os contadores 4029, os conversores bin-7segm e os "displays" respectivos.



Fig. 22 - A montagem final de todo o circuito de controle.

8.5.4 A Comunicação com o Computador: A Placa de I/O Digital de 24 bits, PIO-12

A comunicação com o computador foi efectuada com uma placa I/O digital de 24 bits, PIO-12, da Metrabyte, cuja descrição detalhada pode ser encontrada no manual "Data Acquisition and Control" daquela marca.

Ela consiste essencialmente numa P.P.I. ligada a três portos de entrada/saída de 8 bits cada; a selecção de entrada ou saída para cada porto é efectuada pelo envio de uma "palavra" de controle ("byte") para um endereço específico (= endereço base +3). Para uma configuração em que o 1º porto (PA) é programado para saída (transmitindo os sinais de "clock", "up/down" e "reset") e o 2º porto (PB) para entrada (recebendo o sinal de fim-de-curso do espelho rotativo), a "palavra" de controle é $1000\ 0010_B = 130_D$, que no início de cada utilização deve ser escrita no endereço $300_H + 3 = 303_H = 771_D$.

Fig. 20 - Os contadores 4029, os conversores bin-7segm e os "displays" respectivos (LA6780).

Refira-se ainda que a placa é ligada ao exterior através de uma ficha de 38 pinos, sendo os pinos 37, 36 e 35 ligados respectivamente ao "clock", "up/down" e "reset" e o pino 10 ligado ao sinal de fim-de-curso. Os pinos 20 e 21 transportam os sinais respectivamente de 5V e da terra da fonte de alimentação do computador. Na ligação ao cabo série foram utilizados as seguintes correspondências de cor:

Azul	Sinal de Clock
Vermelho	sinal do sentido de rotação "Up/Down"
Verde	sinal de "Reset"
Laranja	5V (fonte de alimentação do computador)
Preto	0V (fonte de alimentação do computador)
Castanho	não ligado

Ainda de acordo com o esquema de ligações daquela ficha, os 3 sinais de saída vão corresponder respectivamente aos bits 0, 1 e 2 de PA e o sinal de entrada ao bit 0 de PB.

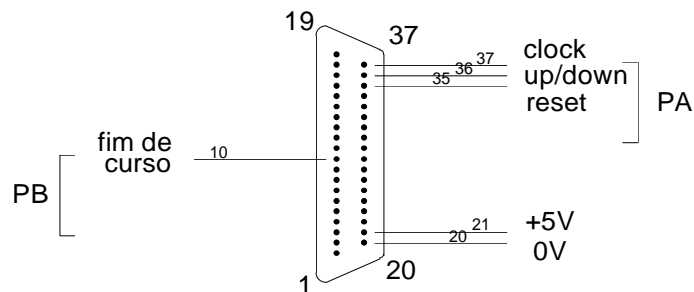


Fig. 23 - O esquema de ligações da tomada de 25 pinos da placa PIO-12.

Para enviar um controle através de PA basta manipular uma variável ("byte"), que em qualquer instante traduz o valor "segurado" pela P.P.I. e transmitido para o exterior através da ficha de 38 pinos. Esse "byte" é sempre mantido em memória; sempre que é necessário enviar um sinal determinado (Fig.24) basta alterar o bit correspondente e reenviar o "byte" pelo porto de saída PA (endereço = endereço de base + 0).

8.6 O Software de Controle

8.6.1 O Controle de Baixo Nível

O envio/leitura de todos estes sinais é efectuado com as instruções de C "inportb(endereço,byte)" e "outportb(endereço,byte)", em que "endereço" toma o valor

de base 300_H+3 se for pretendido enviar o byte de controle, 300_H para aceder ao porto A (PA), 300_H+1 para o porto B (PB) e 300_H+2 para o porto C (PC).

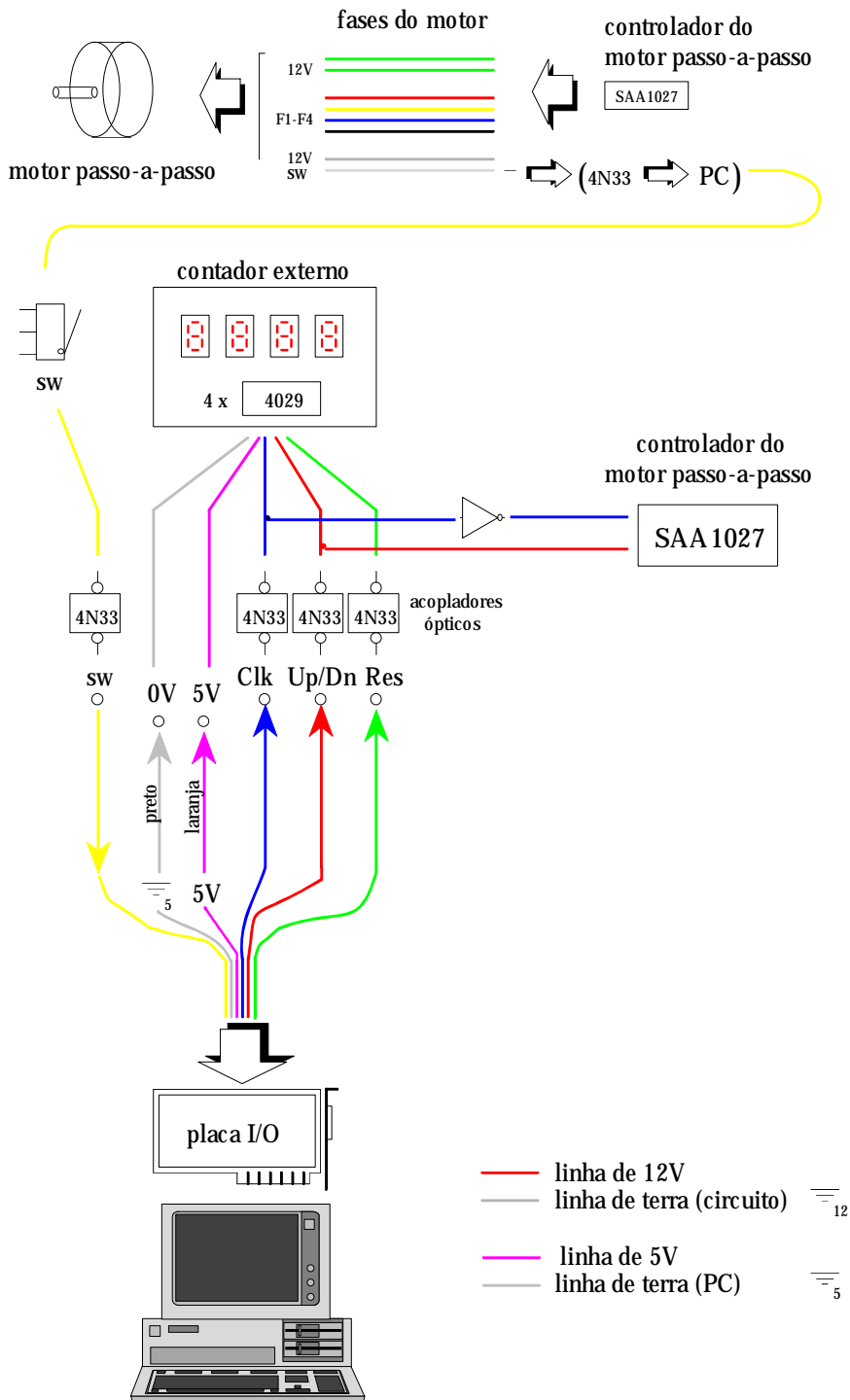


Fig. 24 - Esquema completo de circuitos e ligações entre o computador e o motor passo-a-passo.

Usando apenas estas duas funções foram construídas em C os três comandos (clock, up/down e reset) e a função de leitura "receive()", bastando para tal actuar ao nível do bit no byte enviado/recebido. Apresenta-se de seguida a listagem destas funções

na qual são apresentados num primeiro nível, o controle de bit como referido atrás: "set_bit()" e "reset_bit()"; num segundo nível e por indicação explícita do bit a alterar e para que valor (0 ou 1), temos as funções seguintes:

- clock_0() e clock_1()
- up_down_0() e up_down_1()
- reset_0() e reset_1()
- initboard()
- receive()

Por associação destas últimas consegue-se obter os comandos:

- clock()
- up()
- down()
- reset()

que com "initboard()" e "receive()" já apresentados atrás, formam as primitivas de controle do motor passo-a-passo através do controlador e do circuito associado já descrito e portanto da linha laser incidente na cena.

Por associação destas conseguem-se ainda comandos de maior nível: "go_back()" que faz retornar a linha laser até à posição para a qual a variável interna "counter" atinge o valor 0; "search_home()" que efectua a procura da posição de referência fazendo uso do sinal de fim-de-curso activado no interruptor colocado no "scanner", para os casos em que o valor de "counter" não é fiável ou conhecido, e finalmente, "steps(int,int)" que envia o número especificado de sinais de clock, que de acordo com o 2º parâmetro, faz a linha laser progredir ou recuar aquele nº de passos (Fig.25).

O controle da linha torna-se assim transparente e imediato, não sendo mais necessário actuar ao nível dos bits nem de conhecer os meandros da transmissão de sinais.

8.6.2 A Obtenção de Imagens de Distância com Densidade de Pontos Constante

Os comandos directos anunciados atrás permitem o controle total do motor passo-a-passo de uma forma completamente transparente.

O processo de construção de uma imagem de distância poderia então consistir em aquisições de imagem da linha laser, na amostragem de pontos ao longo de cada linha, o cálculo das respectivas distâncias associadas seguido de registo em memória e finalmente de um incremento de i impulsos de clock, antes do ciclo fechar e tudo repetir.

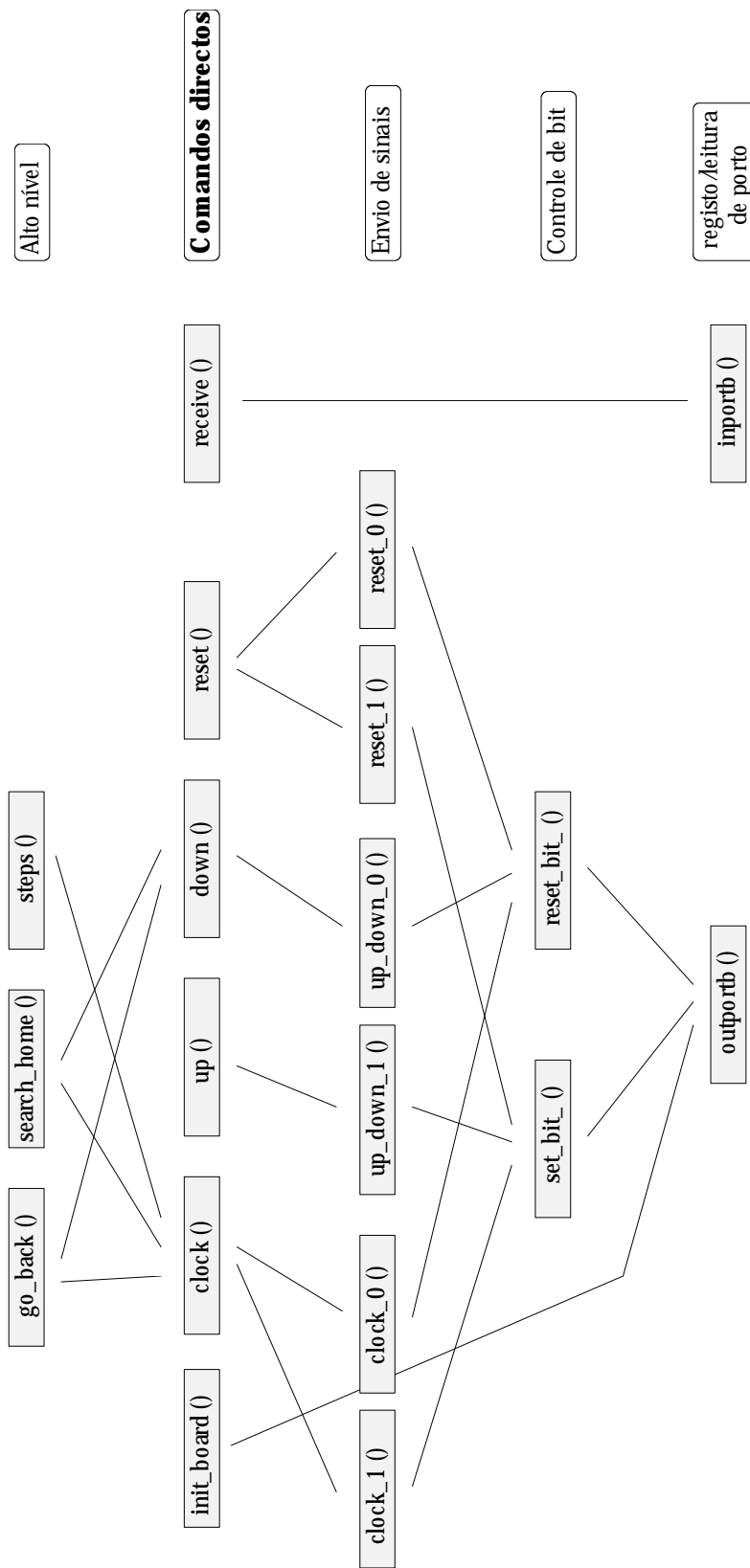


Fig. 25 - Estrutura hierárquica de funções de controle.

Será contudo fácil de constatar que o deslocamento lateral da linha laser para cada conjunto de i impulsos varia com o ângulo inicial de incidência do feixe na cena (Figs. 26 e 27). No caso particular da montagem efectuada, esse deslocamento aumenta sempre com θ de que resulta uma densidade de pontos na imagem de distâncias cada vez menor. Este efeito não é de modo algum desejável, pelo que houve a necessidade de ajustar o nº de impulsos a enviar consoante o valor θ do feixe. Mais concretamente, foi definido um dado valor para deslocamento lateral elementar pretendido e em função do valor de θ do feixe, foi calculado o nº de impulsos a enviar ao controlador para se obter aquele deslocamento. Sendo assim, e uma vez que os deslocamentos laterais podem agora ser numerados, da fórmula que for obtida poder-se-á construir uma tabela de valores discretos a utilizar em cada instante consoante o nº total de deslocamentos laterais elementares já efectuados a partir da posição 0. Tal fórmula vai depender não só do valor escolhido para aquele deslocamento mas também da configuração de todo o sistema, ou seja das várias distâncias como a altura h do espelho rotativo à mesa de trabalho, ou a distância d medida na horizontal do início da mesa de trabalho a que corresponde a posição 0, até à vertical do espelho rotativo (Fig.26).

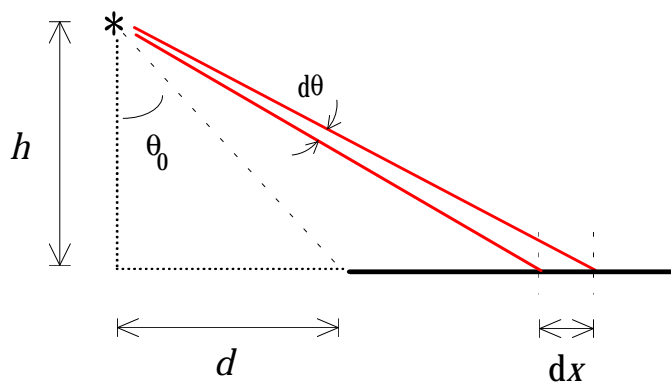


Fig. 26 - Deslocamento da linha laser em dx como resultado da rotação $d\theta$ do espelho.

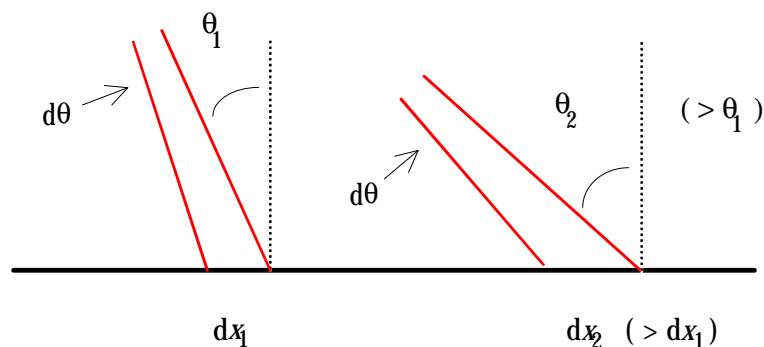


Fig. 27 - O valor do deslocamento dx como resultado da rotação $d\theta$ do espelho depende do ângulo de incidência θ (e portanto da posição absoluta x).

A formulação é então a seguinte: $\text{tg}\theta = \frac{(x'+d)}{h}$

por diferenciação obtém-se: $\frac{1}{\cos^2\theta} \times d\theta = \frac{h}{h^2 + (x'+d)^2} \times dx'$

onde $\underline{dx'}$ é o deslocamento elementar pretendido e $\underline{d\theta}$ é a rotação infinitesimal que lhe dá origem. Este pode ser dado por:

$$d\theta = \text{nr_steps}(x') \times \frac{\text{angle_per_step}}{\text{gear}}$$

onde "nr_steps(x')" é o nº de impulsos a enviar para se obter aquele $\underline{d\theta}$, "angle_per_step" é o menor ângulo de rotação do motor que no nosso caso é de $7,5^\circ$ e finalmente "gear" é o factor de desmultiplicação da caixa reductora. Por substituição obtém-se:

$$\text{nr_steps}(x') \times \frac{\text{angle_per_step}}{\text{gear}} = \frac{h}{h^2 + (x'+d)^2} \times dx' \Leftrightarrow$$

$$\Leftrightarrow \text{nr_steps}(x') = \left(\frac{\text{gear}}{\text{angle_per_step}} \right) \times \frac{\frac{dx'}{h}}{1 + \left(\frac{x'+d}{h} \right)^2} = \text{ratio} \times \frac{\frac{dx'}{h}}{1 + \left(\frac{x'}{h} + \frac{d}{h} \right)^2}$$

Ora, $\text{angle_per_step} = 7,5\pi/180$ e $\text{gear} = 500$ pelo que,

$$\text{ratio} = \text{gear}/\text{angle_per_step} = 12000/\pi \text{ passos/rad}$$

e $h \approx 1160\text{mm}$, $d/h = \text{tg}(0,361737) \approx 0,3784$ pelo que finalmente,

$$\text{nr_steps}(x') = \frac{12000}{\pi} \times \frac{\frac{dx'}{1160}}{1 + \left(\frac{x'}{1160} + 0,3784 \right)^2}$$

Aquele valor (0,361737) é obtido após calibração do medidor laser (ver a secção correspondente) já que \underline{d} é difícil de medir com precisão, directamente.

Ainda, $\underline{x'}$ é agora discreto e igual a $i \times \underline{dx'}$ com $i \in N_0$ (ele corresponde ao nº total de deslocamentos laterais constantes $\underline{dx'}$ já efectuados) e $\underline{dx'}$ é convencionado ser igual a 1mm, pelo que,

$$\text{nr_steps}(i) = \frac{12000}{\pi} \times \frac{1}{1160} \frac{1}{1 + \left(\frac{i}{1160} + 0,3784\right)^2}$$

Atribuindo valores a $i \in N_0$ obtém-se uma tabela de valores inteiros para "nr_steps()":

i	nr_steps
0	2,880 → 3
1	2,878 → 3
2	2,877 → 3
3	2,876 → 3
...	...
215	2,500 → 3
...	...
300	2,342 → 2
...	...
450	2,075 → 2

Estes valores ficam então armazenados em disco.

Como se pode constatar esta tabela envolve apenas dois valores distintos pelo que na transição a imagem de distância sofre um aumento brusco de densidade de pontos (correspondente à diminuição de 3 para 2 passos do motor para cada ordem de deslocamento lateral aproximadamente constante). Esta variação poderia ser mais suave e ocorrer mais frequentemente se da tabela constassem mais valores (p.ex. 4, 3, 2 e 1). Tal seria conseguido com uma desmultiplicação superior à usada: 750× ou mesmo 1000×. Contudo, dada a qualidade das imagens de distância obtida, tal não foi concretizado e "gear" manteve o seu valor de 500×.

8.6.3 A Construção de Novos Comandos de Alto Nível

Uma vez de posse daquela tabela de passos em função do nº de deslocamentos laterais constantes, estamos em condições de construir funções de controle ainda de maior nível e nas quais não se faça mais referência ao nº de passos a enviar mas sim de deslocamentos. Para tal é necessário manter também em memória (para além do "byte" de controle do motor enviado através de PA e do nº de passos já efectuados por aquele desde a posição de referência) o nº de deslocamentos total (com os valores negativos também contabilizados) já efectuados pela linha laser. Assim, se se pretender que numa determinada posição da linha, ela efectue um incremento $\underline{dx'}$ (deslocamento lateral

constante) basta "ler" o valor da tabela cuja entrada é aquele n° total a que se pode convencionar chamar n_shifts . A função que executa esta acção obtém aquele valor por consulta a um vector (Fig.28). O número de deslocamentos laterais de valor constante registado em n_shifts é então incrementado em 1 unidade. O n° total de passos mantido em $counter$ (importante para em qualquer instante se conhecer o ângulo de inclinação do feixe sobre a cena) é também actualizado, agora pela soma do valor lido no vector referido.

Uso:	<code>one_shift(n_shifts)</code>
Nº de passos enviados ao motor:	<code>steps1[n_shifts]</code>
Actualização da variável " n_shifts ":	<code>n_shifts += 1</code>
Actualização da variável " $counter$ ":	<code>$counter$ += steps1[n_shifts]</code>

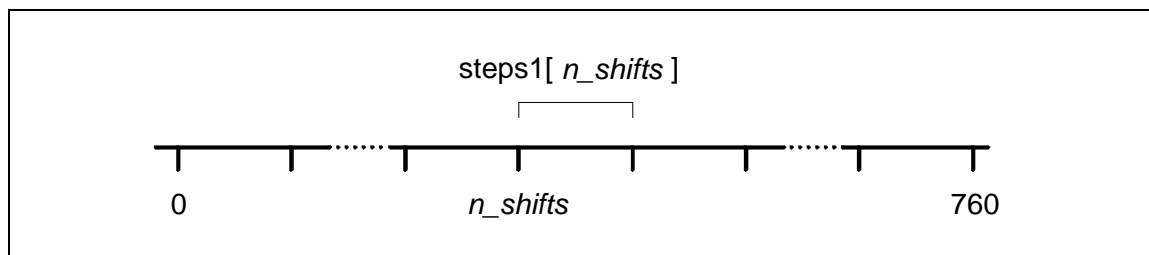


Fig. 28 - Obtenção do n° de impulsos a enviar ao motor com " $steps1[]$ " em função da posição corrente da linha dada por n_shifts , para se obter o próximo deslocamento constante na horizontal.

Para maior velocidade de execução esta tabela é lida do disco apenas uma vez e também mantida em memória (vector " $steps1[]$ ").

Uma outra função de alto nível e de maior importância que a anterior foi ainda construída, e denominada de " $goto_pos()$ ". Ela destina-se a permitir o deslocamento da linha laser para uma posição absoluta (em termos de n_shifts) qualquer que seja a sua posição inicial. Supondo que ela se encontra na posição 0 e se pretende movê-la para a posição n , então o n° de passos a enviar ao motor será o somatório de todos os valores da tabela acima, desde 0 até $n-1$, ou seja,

$$\sum_{i=0}^{n-1} steps1[i]$$

Para evitar efectuar cálculos sempre que se chama " $goto_pos()$ ", constrói-se outra tabela com os valores do n° total de passos a enviar para que a linha se mova de 0 para a posição absoluta n correspondente ao n° de deslocamentos laterais de valor constante que se pretende efectuar, para todos os valores de $n \in \mathbb{N}_0$. O valor a registar para cada uma daquelas entradas n é o anunciado acima.

Esta tabela é também armazenada em disco e mantida em memória sob a forma de vector.

Considerando agora que a linha não parte da sua posição de referência mas da posição n_shifts e se pretende que se mova para uma nova posição $location$, basta ler na tabela os valores correspondentes à entrada $location$ e n_shifts e subtraí-las (Fig.29).

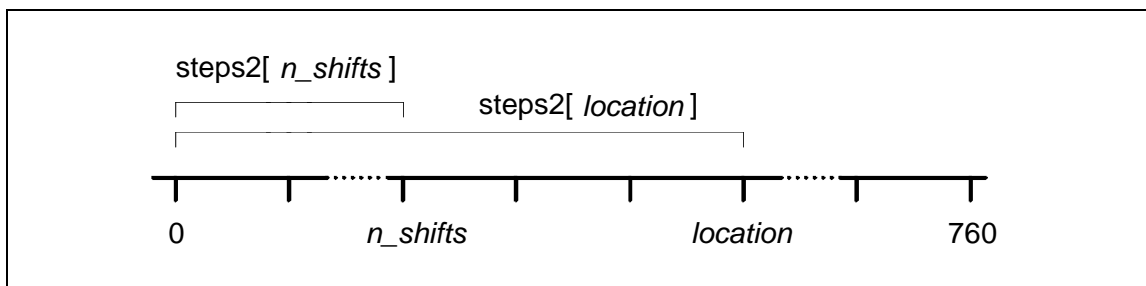


Fig. 29 - Obtenção do nº total de impulsos a enviar ao motor para a linha se deslocar da posição absoluta n_shifts em que se encontra para a nova posição absoluta $location$.

$$\begin{cases} steps2[n_shifts] = \sum_{i=0}^{n_shifts-1} steps1[i] \\ steps2[location] = \sum_{i=0}^{location-1} steps1[i] \end{cases}$$

O valor obtido daquela subtracção corresponde ao nº de incrementos a enviar ao motor.

Mais uma vez a variável n_shifts que contém o nº total de deslocamentos laterais de valor (aproximadamente) constante, efectuados desde a posição de referência, é actualizada pela soma de $(location - n_shifts)$. O nº total de passos do motor é também actualizado pela soma do valor em memória pelo nº de incrementos referido acima.

Em resumo:

Uso:	<code>goto_pos(location)</code>
Nº de passos enviados ao motor:	<code>passos = steps2[location] - steps2[n_shifts] </code>
Actualização de " n_shifts ":	<code>n_shifts += ou -= location - n_shifts </code>
Actualização de " $counter$ ":	<code>counter += ou -= passos</code>

Para completar este novo conjunto de funções de alto nível, foi ainda construída a função " $shifts(n)$ " que permite efectuar n deslocamentos laterais constantes, em qualquer dos dois sentidos (Fig.30). Na realidade, esta é apenas uma variação de "`goto_pos()`" já que ela pode ser traduzida por "`goto_pos(n_shifts + n)`". Foi no entanto mantida por uma questão de facilidade de linguagem quando de uma qualquer programação.

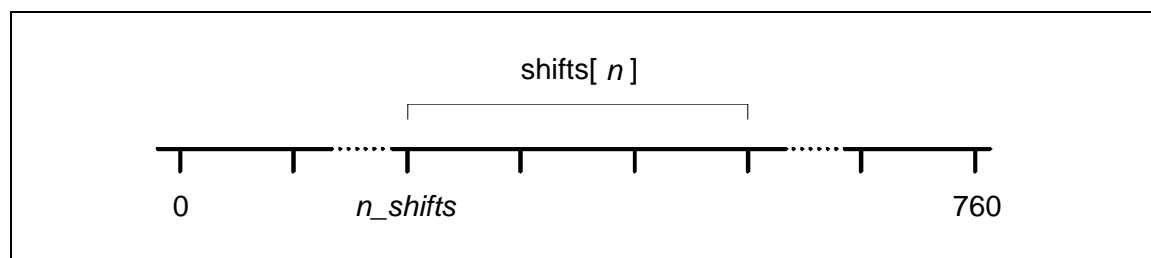


Fig. 30 - A nova função *shifts()* calcula o nº de impulsos a enviar ao motor em função do nº de deslocamentos horizontais constantes pretendidos e da posição absoluta prévia, *n_shifts*.

Com estas três funções é possível efectuar o completo controle da linha laser e portanto do medidor de distâncias e vão constituir o conjunto de ferramentas com o qual é posteriormente efectuada a aquisição de imagens de distâncias.

8.7 Resultados Experimentais

Para que se possam obter imagens de distâncias de boa qualidade é essencial que todo o sistema óptico e mecânico, que constituem o hardware do sistema de medida de distâncias, seja robusto e fiável. E agora que já se dispõe de um conjunto de comandos de alto nível para controle da linha laser, resta a realização de testes experimentais que atestem aquela robustez, quer pela medida de passos perdidos pelo controlador do motor, quer pela repetibilidade de operações de posicionamento ou varrimento, ou ainda pela análise da independência do "caminho" tomado para atingir determinadas posições. Esses testes foram realizados e os resultados serão apresentados a seguir.

8.7.1 Testes de Posicionamento da Linha Laser

8.7.1.1 Varrimento com retorno; voltas completas

Como primeiro teste na resposta do hardware em termos de perda de "steps" ou passos enviados ao controlador do motor e simultaneamente para a verificação dos parâmetros intrínsecos do sistema mecânico (motor + caixa de desmultiplicação) foram efectuados ensaios variados de varrimento eventualmente com retorno:

1. voltas completas (24000 passos)
2. voltas completas com retorno
3. 1/6 de volta (4000 passos) com retorno
4. 1/3 de volta (8000 passos) com retorno
5. 1/2 de volta (12000 passos) com retorno

De acordo com as especificações dos fabricantes, o motor efectua uma rotação de $7,5^\circ$ por cada sinal de incremento no controlador, e a caixa de desmultiplicação possui um factor de $500\times$. Sendo assim uma volta completa deverá corresponder a $500\times 180^\circ/7,5^\circ=24.000$ passos. As primeiras duas experiências destinam-se a confirmar este resultado.

1. volta completa (24.000 passos):

Experiência n°	n° de passos perdidos
1	18
2	-6
3	48
4	-3
5	9
6	9
7	21

Experiência n°	n° de passos perdidos
8	-9
9	12
10	3
11	-3
12	35
13	2
14	3

n° médio de passos absolutos perdidos = **12,9** *passos*
 (desvio absoluto médio)
 erro relativo = **0,054%**

2. volta completa com retorno:

Experiência n°	n° de passos perdidos
1	0
2	0
3	0
4	0
5	0

Experiência n°	n° de passos perdidos
6	0
7	0
8	0
9	0
10	0

n° médio de passos absolutos perdidos = **0** *passos*
 (desvio absoluto médio)
 erro relativo = **0%**

3. 1/6 de volta (4.000 passos) com retorno:

Experiência n°	n° de passos perdidos
1	3
2	4
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0

Experiência n°	n° de passos perdidos
11	0
12	0
13	0
14	0
15	0
16	0
17	0
18	0
19	0
20	0

n° médio de passos absolutos perdidos = **0,35** *passos*
 (desvio absoluto médio)
 erro relativo = **0,009%**

4. 1/3 de volta (8.000 passos) com retorno:

Experiência n°	n° de passos perdidos
1	0
2	6
3	-2
4	9
5	0
6	0
7	-6
8	0
9	9
10	0

Experiência n°	n° de passos perdidos
11	-3
12	0
13	5
14	0
15	0
16	0
17	0
18	0
19	0
20	0

n° médio de passos absolutos perdidos = **2** *passos*
 (desvio absoluto médio)
 erro relativo = **0,025%**

5. 1/2 de volta (12.000 passos) com retorno:

Experiência n°	n° de passos perdidos
1	-7
2	-12
3	-3
4	0
5	0

Experiência n°	n° de passos perdidos
6	0
7	0
8	0
9	6
10	-3

n° médio de passos perdidos = **3,1** *passos*
 (desvio absoluto médio)
 erro relativo = **0,026%**

Estes resultados confirmam o n° 24.000 como sendo o n° de passos necessários para o veio da caixa de desmultiplicação ligada ao espelho efectuar uma volta completa. Confirma-se também o grande rigor de posicionamento cujo erro relativo ronda normalmente os 0,02% (2 passos por dez mil) e não excede os 0,054%.

8.7.1.2 Varrimento e retorno, várias vezes

Para testar a repetibilidade e o rigor de posicionamento foram feitas várias experiências nas quais eram enviados ao controlador do motor vários conjuntos de incrementos intercalados com retorno à posição inicial, ou seja, com o envio do mesmo n° de incrementos com ordem de os efectuar no sentido negativo.

Os resultados obtidos foram os seguintes:

n° de ciclos	n° de passos / ciclo	total de passos	n° de passos perdidos	Δt (s)	v (passos /s)
10	600	12.000	18	45	267
10	300	6.000	9	22	273
20	600	24.000	10	91	264
20	800	32.000	7	122	263
30	500	30.000	6	115	261
10	100	2.000	1	8	250
30	100	6.000	4	23	261
30	200	12.000	13	46	261
50	800	80.000	8	300	267
1	800	1.600	6	6	267

As temporizações introduzidas no sinal de clock permitem um varrimento à velocidade de 263 passos/s (cerca de 263 mm/s), conforme medidas efectuadas e apresentadas no quadro. Para esta velocidade de varrimento obtive perdas desprezáveis em número de passos do motor. Isto implica que o posicionamento da linha é bastante rigoroso: a avaliar pelo conteúdo da tabela referente a uma série de experiências, poder-se-ia concluir que para um posicionamento a uma "distância" de 400 passos, se perderiam em média 0,17 passos, ou seja cerca de 0,2 mm!

8.7.2 Uso das Funções de Deslocamento Horizontal Constante, Avaliação do Rigor de Posicionamento (sem dependência no caminho) e Repetibilidade

Como já ficou visto, o aumento gradual na inclinação do feixe laser sobre a cena, origina um deslocamento cada vez maior na horizontal para o mesmo número de passos enviados ao "driver" do motor. Sendo assim e uma vez que o deslocamento lateral da linha para cada passo é imperceptível (inferior a 1 mm), decidi convencionar um deslocamento constante mínimo de 1mm e calcular o número de passos necessários para atingir esse valor para qualquer inclinação do feixe, como já foi visto acima. Cada um desses deslocamentos constantes poderá conter 2 ou 3 passos (para o desmultiplicador usado) consoante a inclinação. Três funções foram construídas que utilizam esses deslocamentos: *'goto_pos()'*, *'one_shift()'* e *'shifts()'* como já foi visto.

Foram estas funções de maior nível que permitiram testar com mais rigor o desempenho do hardware (já que o controle da linha será sempre efectuado em termos de "shifts" em vez de passos). Refiro-me concretamente (1) à capacidade de posicionamento da linha numa dada abcissa sem erro significativo, qualquer que seja o caminho utilizado para o atingir e (2) à repetibilidade em qualquer desses posicionamentos.

Com esse objectivo em mente, foram desenvolvidos dois testes que a seguir se descrevem resumidamente:

O 1º teste :

- a) lê os ficheiros necessários para a escolha do nº de passos a efectuar para cada deslocamento em função do nº de passos já efectuados desde a posição inicial de referência.
- b) executa um nº inicial de deslocamentos (*shift0*).
- c) executa conjuntos isolados de deslocamentos em ciclo fechado até que o experimentador decida interromper a acção (quando a linha tiver atingido a abcissa pretendida) altura em que a experiência ficará completada.
- d) efectua uma aquisição de imagem para medir essa abcissa.
- e) retorna à posição inicial (o nº de passos necessários para a atingir encontra-se memorizado) o que permite verificar se há perda de sinais de clock ou falta de robustez do hardware.

Este teste é útil quando o nº de conjuntos de deslocamentos a enviar é pequeno.

Quando o nº de conjuntos de deslocamentos a enviar é elevado, a simples repetição da experiência (para verificar se a mesma abcissa é atingida e portanto se o sistema possui boa repetibilidade) requer grande esforço na introdução de dados. A título de exemplo refiro o teste nº4 no qual em cada uma das dez experiências se introduzem 141 inteiros perfazendo 1410 operações de entrada de dados. Por essa razão foi criado o 2º teste apresentado a seguir.

O 2º teste :

- aceita os conjuntos de "shifts" uma só vez, que deverão ser enviados repetidamente \underline{n} vezes.
- permite, antes da repetição da experiência, ajustar o posicionamento inicial da linha laser, possuindo um ajuste tanto grosso como fino.
- após o envio automático dos \underline{n} conjuntos de deslocamentos previamente memorizados, o programa efectua uma aquisição de imagem.
- chama ainda uma função interactiva que faz percorrer um ponto luminoso no ecrã do monitor onde se visualiza a imagem da cena captada pela câmara e a linha laser projectada; permite ler a abcissa atingida por ela.
- finalmente retorna a linha à sua posição inicial tal como o programa anterior já fazia, o que mais uma vez permite verificar a robustez do sistema.

Com base nestes 2 testes, elaborou-se um conjunto de experiências que se apresenta na Tabela 1 e cujos resultados aparecem na Tabela 2.

Tabela 1

Teste nº	nº de conj. de shifts	Conteúdo dos conjuntos
1	1	140 <i>shifts</i>
2	7	20 + 20 + 20 + 20 + 20 + 20 + 20 <i>shifts</i>
3	9	10 + 1 + 1 + 3 + 20 + 5 + 10 + 50 + 40
4	140	1 + 1 + 1 + ... + 1
5	11	50 + 3 + 10 + 1 + 5 + 40 + 1 + 7 + 3 + 18 + 2
6	14	10 + 10 + 10 + ... + 10
7	19	5 + 10 + 5 + 10 + ... + 5 + 10 + 5
8	6	250 - 170 + 90 - 45 + 22 - 7
9	8	250 - 170 + 90 - 45 + 22 - 7 - 30 + 30

Tabela 2

Teste n°	Posições atingidas (abscissas na imagem)									
1	268	261	263	265	266	263	259	261	263	265
2	263	265	259	260	264	265	262	264	263	265
3	256	260	263	264	262	264	265	261	263	265
4	257	259	260	259	259	258	258	258	258	258
5	260	260	260	260	260	260	261	261	261	261
6	263	263	263	263	263	263	263	263	263	263
7	265	268	262	262	262	262	262	262	262	262
8	230	226	228	223	-	-	-	-	-	-
9	261	260	255	255	255	255	254	254	254	255

Pela análise da Tabela 2 várias conclusões poderão ser tiradas:

1. Para cada teste e ao fim de 10 experiências, verifica-se que a linha laser atinge praticamente o mesmo ponto x (abscissa da imagem), com flutuações que vão desde 3~4 pixels de imagem (exp^a 1) até 0 (exp^a 6). Ora 3 pixels correspondem aproximadamente a 1 deslocamento de valor constante e também a 1mm na área de trabalho, o que é um valor de erro bastante satisfatório.
2. A discrepância de teste para teste aumenta ligeiramente em relação ao caso anterior; mas esse facto poderá estar relacionado com a incerteza no posicionamento inicial da linha laser (posição de referência) que se estima em cerca de 2 deslocamentos, ou seja 4~6 steps e ~2mm na cena. Fica portanto provada a grande precisão atingida e a independência do posicionamento no caminho utilizado para o atingir.
3. De notar o grande erro de posicionamento nas experiências do teste n°8; isto deve-se ao facto de este conter várias alterações no sentido de rotação e de existir um efeito de histerese no mecanismo de desmultiplicação do motor, decerto devido a folgas. Este efeito é simultaneamente confirmado e eliminado no teste n°9, igual ao n°8 mas ao qual são acrescentados 2 conjuntos finais de deslocamentos: -30 e 30; sendo este último suficiente para compensar a referida folga que é da ordem dos 20 deslocamentos. Neste teste n°9 os posicionamentos são já tão precisos quanto os dos testes anteriores, apesar de conter várias inversões no sentido de rotação do motor.

Na tabela não constam os erros de retorno à posição inicial após cada experiência, dado que eles eram praticamente inexistentes ($< 1\text{passo} \approx 1/3\text{deslocamento}$), o que atesta a grande repetibilidade do sistema de varrimento.

Pela análise destes resultados experimentais pode-se concluir que o sistema de varrimento é de facto bastante preciso.

Embora um erro de posicionamento de 2 ou 3 deslocamentos possa parecer excessivo para a construção de imagens de "range" (distância ou altura) nas quais seja utilizada a resolução máxima, ele não tem contudo qualquer efeito negativo, já que o algoritmo de aquisição regista a posição (x,y) na qual a medida Z foi efectuada (como será visto adiante) tornando-o imune a pequenos desvios. Sendo assim é dispensável o posicionamento exacto da linha laser na cena. O único efeito que poderia surgir seria o de o sistema não conseguir garantir deslocamentos de largura constante; e da sua variação quando do varrimento, resultar uma flutuação na densidade dos pixeis de distância (pixeis de "range").

Como se pôde verificar nos testes apresentados, nem essa situação poderá ocorrer dada a grande precisão de posicionamento verificada. Isto foi fruto quer de um desenho cuidado do sistema físico, que lhe conferiu robustez, quer da correcta filtragem e transmissão de sinais de clock do computador ao driver do motor (que evita a perda de passos), quer ainda da construção cuidada de funções de posicionamento de alto nível.

8.8 Conclusão

Pelos resultados apresentados acima pode-se concluir que o sistema construído é de facto robusto e fiável e pode portanto ser utilizado na aquisição de imagens de distância.

A estes resultados não é estranho o cuidado com que todo o sistema foi desenhado desde o seu início, não só em termos de hardware mas também na correcta transmissão e filtragem de sinais entre o computador, responsável por todo o controle, e o circuito associado ao motor passo-a-passo.

É claro que para a obtenção de boas imagens de distância é também necessária uma correcta medida dos valores de altura e é sobre esse assunto que se irá debruçar o capítulo seguinte. Mas pelo menos em termos de controle e de posicionamento estão reunidas condições de sucesso nesse objectivo.

9.1 Introdução

Já de posse de um sistema de posicionamento da linha laser fiável e de fácil controle, há que detectar as situações de obstrução do feixe provocado pelos objectos e medir correctamente as alturas.

Embora o processo de medida pareça imediato já que ele se baseia na triangulação, ou seja, na medida do desvio provocado na linha laser por obstrução de um objecto e como consequência da inclinação do feixe, o facto é que a correcta medida de alturas envolveu um grande número de acções distintas desde a filtragem da linha laser vista pela câmara até à compensação do efeito de distorção radial da imagem.

Este capítulo debruça-se precisamente sobre essas acções, de cuja eficácia depende a correcta medida de cada um dos valores de altura; e no final, a qualidade das "imagens" de distâncias.

9.2 A Obtenção da Linha Laser

A medida da altura para cada ponto de um objecto atingido pelo feixe laser, é uma função do desvio local da linha laser, tal como é vista pela câmara, em relação à parte dela não obstruída. Este desvio deve-se à projecção desse feixe segundo um determinado ângulo de inclinação e é determinado pela geometria da "gaiola" construída com barras, delimitadora do espaço de acção do robot, numa das quais foi suspenso o dispositivo de rotação do espelho reflector.

Há, portanto, que conseguir destacar essa linha laser vista pela câmara do resto da imagem, de modo a poder ser alvo de medida directa e posterior cálculo. Ainda, há que garantir rapidez na sua obtenção e eficiência na sua filtragem. Para tal foram adoptadas algumas soluções, quer em hardware quer em software, que serão apresentadas nas secções seguintes.

9.2.1 A Binarização em Tempo Real

Dada a óbvia facilidade no tratamento de imagens binárias em oposição às de intensidade (ou de níveis de cinzento), foram aquelas sempre adoptadas. Houve portanto que efectuar sempre a binarização de imagem antes de qualquer análise, como no caso já descrito dos algoritmos a 2D onde era construído o histograma da imagem, escolhido um nível de cinzento intermédio entre os associados aos objectos e ao fundo, e com base nesse valor eram postos a zero todos os pixeis de valor inferior àquele e a 255 os de valor superior ao mesmo. Esta conversão é efectuada pixel a pixel por acesso ao ficheiro de imagem da placa de aquisição e por isso muito lenta (alguns segundos). Ora, envolvendo o processo de aquisição de alturas a binarização de inúmeras imagens, tornou-se claro que aquela solução teria um custo de tempo proibitivo, pelo que houve que recorrer a outras possibilidades. E de facto foi possível usar uma capacidade do hardware da placa de aquisição de imagem que permite converter automaticamente, através do uso de tabelas de conversão programáveis, os valores de intensidade medidos, noutros quaisquer, desde que dentro dos limites de 0 e 255. Trata-se das LUTs ("Look-Up Tables") que funcionam em duas modalidades: (1) as intensidades são registadas em memória tal como foram medidas, mas a sua mostra em monitor ("display") é afectada por aquela tabela (Output LUT); ou (2) as intensidades são imediatamente convertidas sendo registadas já alteradas (Input LUT). Sendo estes processos realizados por hardware, torna-se óbvio que o tempo de execução seja praticamente instantâneo acompanhando o próprio processo de leitura das intensidades.

Foi adoptada a opção da "Input LUT" uma vez que era necessário possuir em memória os valores de intensidade convertidos e não apenas visualizá-los.

Os processos de programação e de activação daquelas tabelas envolvem a manipulação dos registos da própria placa de aquisição de imagem, através dos portos de saída, cujos endereços são ajustáveis por hardware. Sendo o endereço base o valor 340_H , a programação da LUT de entrada envolve os endereços 340_H+2_H e 340_H+3_H [PcVision89,cap.5] correspondentes respectivamente ao "LUT Address Register" e ao "LUT Data Port"; ou seja, aos valores de intensidade a ser alterados e aos seus novos valores. Para poder registar em memória (da placa) os valores de intensidade nos quais se irão transformar todos os valores de entrada entre 0 e 255 há que sucessivamente escrever no endereço 342_H os valores iniciais e no endereço 343_H os valores transformados, produzindo assim uma tabela completa de conversões. A Tabela 1 ilustra as acções necessárias para a selecção da LUT de entrada nº 4 e respectiva programação em degrau para um limiar de binarização de **170**.

Tabela 1

Acção		Escrita de
<p><i>selecção da LUT de entrada 4:</i></p> <p>Seleccção do modo "LUT de entrada" bits 0 e 1 do registo 1 (341_H)</p> <p>Activação de 1 das 8 LUTs de entrada (bits 2 a 4 de 341_H) p.ex. 4 (apenas as LUTS 4 a 7 estão disponíveis para programação)</p> <p><i>programação da LUT de entrada 4:</i></p> <p>registo do valor 0 entre 0 e 169: registo em 342_H (registo do endereço) registo em 343_H (registo do valor)</p> <p>registo de valor 255 de 170 a 255: registo em 342_H (registo do endereço) registo em 343_H (registo do valor)</p>		<p>(19_D em 342_H)</p> <p>3_D (op. lógica OR)</p> <p>16_D (op. lógica OR)</p> <p>i ($\forall i \in [0,170[$) 0</p> <p>i ($\forall i \in [170,256[$) 255</p>

A activação da LUT de entrada é sempre feita no registo 341_H ("LUT Control Register") através da escrita (operação lógica OR) do valor 3_H.

9.2.2 A Importância do Contraste: O Filtro Vermelho

Uma boa binarização pressupõe a salientação dos objectos de interesse contra o fundo, na qual este é posto a zero. Ao contrário do que acontece nos algoritmos 2D, já não tem interesse fazer salientar os objectos presentes na cena uma vez que não são eles o objecto de estudo mas sim a linha laser projectada. Sendo assim, a cor dos objectos relativamente ao fundo torna-se irrelevante e deixam de existir os condicionalismos relativos a ela. O contraste agora deve ser obtido na cor vermelha, entre a linha laser e toda a cena incluindo os objectos. Uma das implicações deste condicionalismo é a utilização de uma fonte laser com potência suficiente, o que elimina os díodos laser mais comuns. Quanto a isso, o laser de He-Ne de 30mW adoptado corresponde inteiramente.

O uso de objectos espelhados foi também evitado. Mas o melhor contributo para a obtenção de um bom contraste foi a aplicação de um filtro vermelho na câmara que fazia salientar a frequência de 632,8nm produzida pelo laser. Deste modo, a escolha de um valor intermédio para a binarização viria muito facilitada e a obtenção de imagens binarizadas, com a linha laser bem nítida, seriam bem conseguidas.

Na Fig.1 é apresentado um caso utilizado para estudo.

Na Fig.2 é mostrado o histograma da imagem quando o filtro vermelho não é utilizado. Aparentemente os pixels dos objectos possuem níveis de cinzento bastante diferentes dos da linha laser, mas uma ampliação do histograma (Fig.3) mostra o contrário.

Para o limiar de binarização utilizado no software (170), já se verifica (Fig.4) a salientação da linha laser projectada com eliminação dos pixels dos objectos (onde os vários traços com diferentes abcissas traduzem as várias alturas a que os objectos ou mesa atingidos se encontram do plano horizontal); mas há parte de um objecto que se mantém na imagem, inviabilizando o processo de aquisição de alturas. A obtenção de uma boa binarização implica o constante ajuste da abertura do diafragma da câmara consoante o tipo de imagem que se possui, o que contraria o pressuposto da não intervenção humana no funcionamento do Agente Laser.

A utilização do filtro vermelho, por seu lado, dispensa essa intervenção, uma vez que alarga a distância entre os níveis de cinzento dos objectos e os da linha laser (Figs. 5 e 6), aumentado assim a gama de valores possíveis para limiar de binarização. A escolha desse limiar deixa de ser alvo de cuidados e pode ser arbitrada, passando a ser utilizado para todas as imagens o mesmo valor. Na Fig.7 mostra-se o resultado da binarização da imagem para um limiar de 170 (com utilização do filtro vermelho), em que aparece já somente a linha laser tal como era pretendido.

Uma primeira análise dos histogramas apresentados poderia sugerir a utilização de um limiar próximo de 255 já que todos os pixels da linha laser possuem este valor de nível de cinzento. Contudo, isto deve-se à saturação dos pixels a ela associados, resultado da abertura utilizada do diafragma da câmara, o que pode nem sempre acontecer. E nessa situação, apenas uma percentagem dos pixels da linha laser apareceriam na imagem binarizada. Perder-se-ia portanto informação valiosa (na realidade seriam os pixels das extremidades da linha laser que desapareceriam uma vez que possuem menor intensidade luminosa).

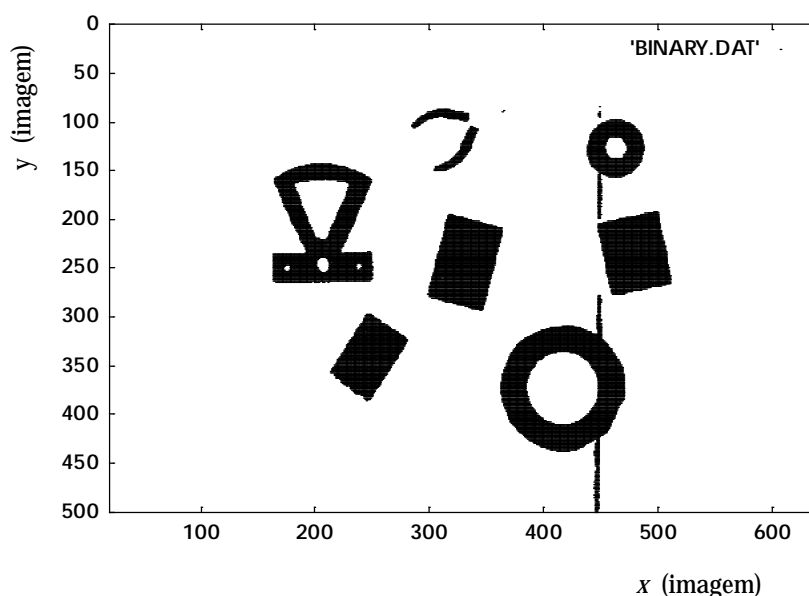


Fig. 1 - Imagem binarizada da cena que inclui a linha laser projectada.

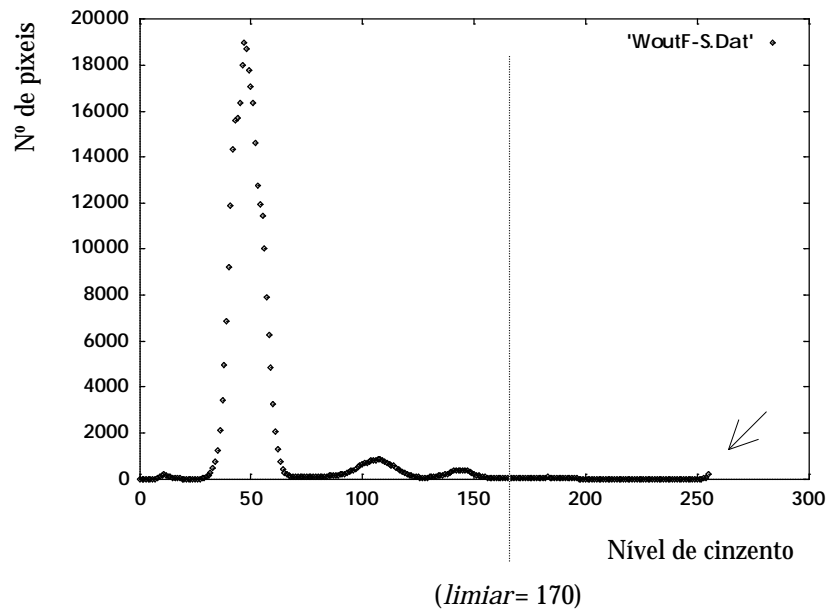


Fig. 2 - Histograma da imagem sem uso do filtro vermelho. (A seta indica os pixels da linha laser).
 Aparentemente, os níveis de cinzento associados aos objectos diferem muito dos associados à linha laser.

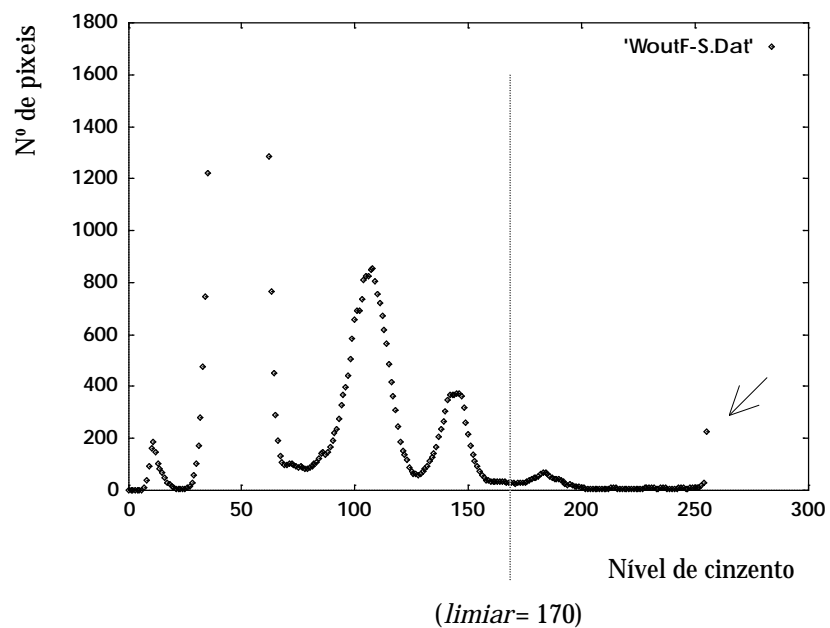


Fig. 3 - Uma ampliação do histograma mostra que há pixels com nível de cinzento próximo dos da linha laser. No caso do limiar utilizado (170), parte dos objectos vai-se manter na imagem após binarização. (A seta indica os pixels da linha laser).

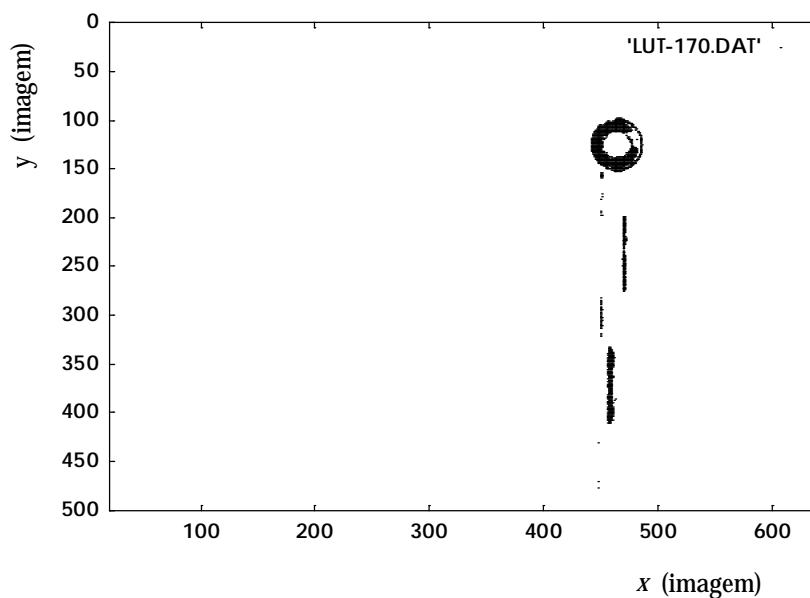


Fig. 4 - Aquisição binarizada da imagem quando o filtro não é usado. A escolha do limiar de binarização é difícil: por vezes aparecem objectos conjuntamente com a linha laser.

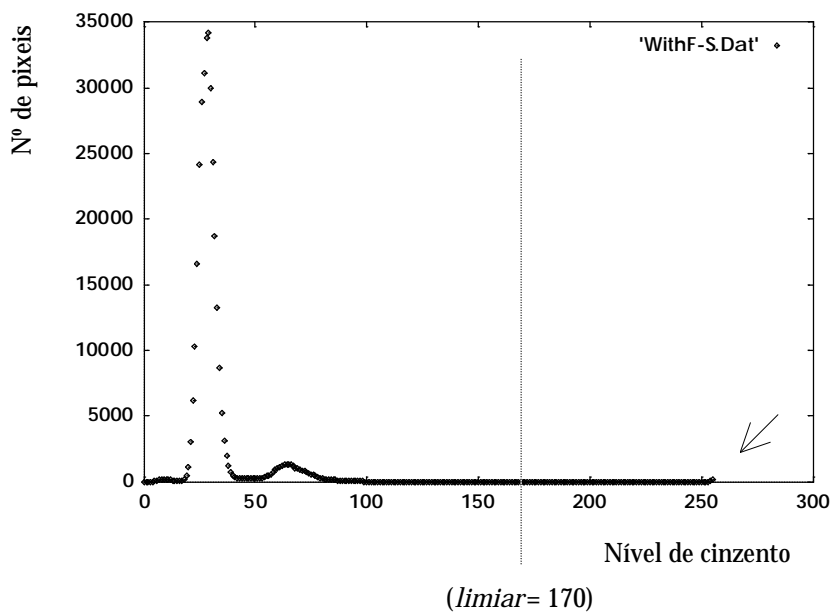


Fig. 5 - Histograma da imagem quando o filtro vermelho é usado.

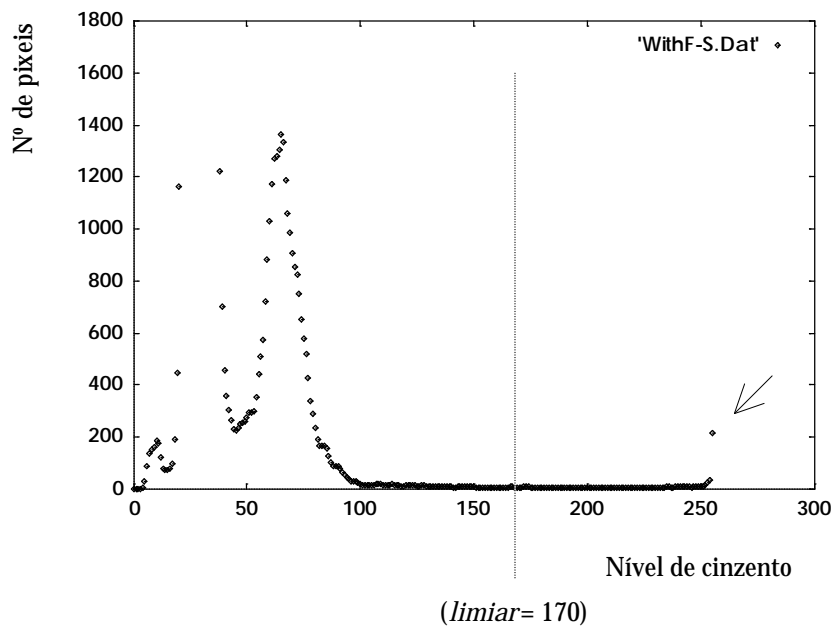


Fig. 6 - Histograma ampliado com uso do filtro vermelho. (A seta indica os pixels da linha laser). Neste caso os pixels dos objectos possuem já níveis de cinzento suficientemente afastados dos da linha laser para permitir alguma arbitrariedade na escolha do limiar de binarização.

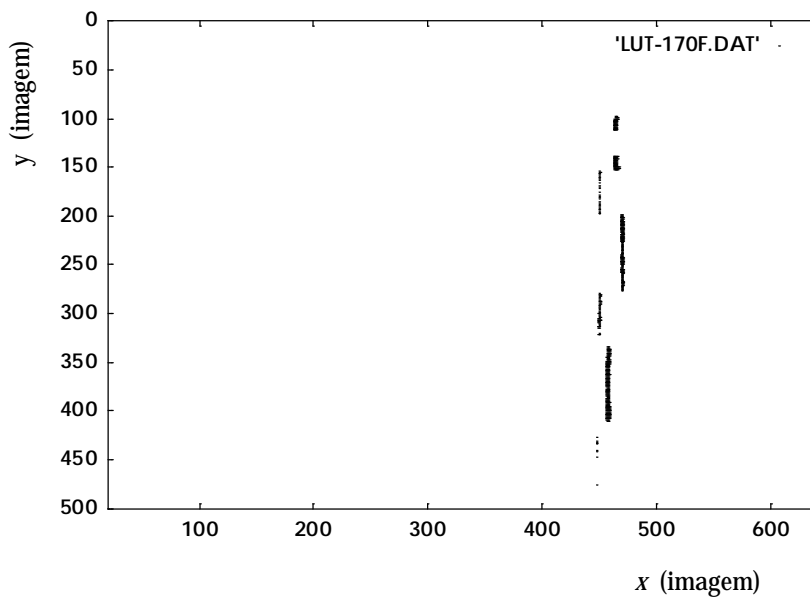


Fig. 7 - Aquisição binarizada da imagem quando o filtro é usado. Uma gama larga de valores para limiar fornece uma boa binarização de que resulta o aparecimento de apenas a linha laser.

9.2.3 A Filtragem por Software da Linha Laser

Embora se tivesse incluído no sistema óptico uma lente convergente de distância focal 1,5m para focar o melhor possível a linha laser na cena, o facto é que ela mantinha uma espessura de ~3mm. E mesmo que se obtivesse um valor inferior àquele para uma dada posição da linha, ele iria certamente aumentar para outras posições da mesma e à medida que o espelho fosse rodando, devido ao aumento da distância lente-cena.

A medida do desvio na linha (visto pela câmara) provocado por obstrução de um objecto, impõe de qualquer modo que a espessura não exceda 1 pixel, passando a focalização efectuada apenas a favorecer leituras para objectos de pequena altura. A redução da espessura da linha para aquele valor terá então que ser feita por software, numa operação que normalmente se denomina esqueletização. É então procurada a posição dos pixels claros (255) para cada linha na imagem e calculado o seu valor médio que é registado num vector. Uma vez atingida a última linha (511), toda a linha laser binarizada é apagada sendo substituída por outra de largura 1 pixel, cujos pixels ficam localizados conforme o conteúdo do vector anteriormente referido.

Esta esqueletização é absolutamente necessária mas não suficiente. Devido a imperfeições na lente cilíndrica e a reflexões múltiplas nas duas interfaces da lente convergente que provocam flutuações de intensidade na linha projectada, ou ainda devido à diminuição desta nas extremidades da linha, esta, após ser binarizada apresenta inúmeras lacunas e imperfeições que requerem tratamento. Foram então implementados em software dois filtros que actuam no vector (a partir do momento em que se dispõe das abcissas de cada pixel para as várias linhas da imagem entre 0 e 511, passa-se a actuar apenas neste vector por uma questão de rapidez de execução). O 1º considera lacunas de pequena extensão, atribuíveis portanto à binarização e não à ausência de linha laser, com valor inferior a 20 pixels; essas lacunas são então preenchidas de acordo com os dois valores que as delimitam (escolheu-se o valor médio das abcissas registadas no vector, desde que a sua diferença não excedesse 2 pixels).

Por último, é efectuada uma suavização por convolução com uma máscara linear de comprimento ajustável $1 \times mask$. O valor actual de *mask* é 7 e a operação associada àquela convolução é a média aritmética das 7 abcissas. Obtém-se assim um outro vector que contém as abcissas finais para a linha laser na imagem binária que é então mostrada no monitor na sua configuração final. É com base nesta linha filtrada que serão efectuadas as medidas de altura.

Apresenta-se de seguida (Fig.8) uma sequência de esquemas que ilustram todo o processo descrito atrás.

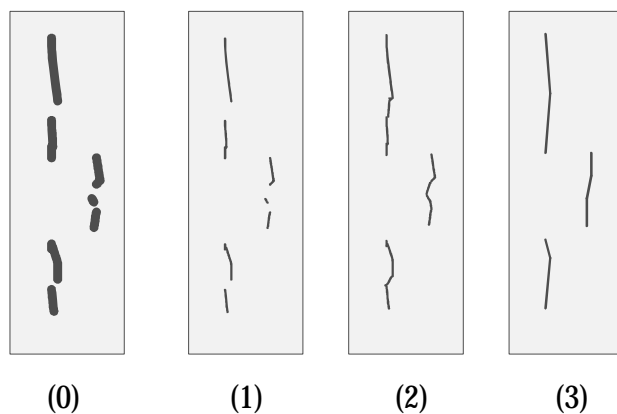


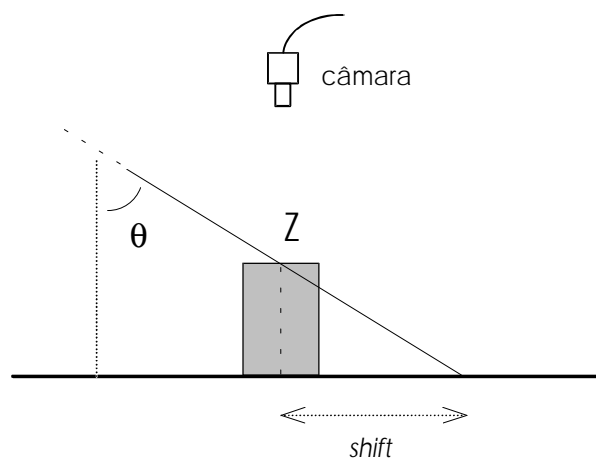
Fig. 8 - Sequência de filtragem da linha laser:

- (0) linha original; (1) linha esqueletizada;
 (2) preenchimento de lacunas; (3) suavização da linha.

9.3 A Medida das Alturas Z

Cada uma das medidas Z a efectuar baseiam-se como já foi dito no desvio da linha laser (em unidades de pixel) observado na imagem (Fig.9). Ou seja, se esse desvio for $shift$ e o feixe tiver uma inclinação de θ (medido da vertical), então Z será dado por,

$$Z = shift \times pixel_to_mm / \operatorname{tg}\theta$$

Fig. 9 - Avaliação da altura Z através da medida de $shift$.

onde $pixel_to_mm$ é o factor de conversão local de unidades de pixel para milímetros. Ora, se bem que aquele valor $shift$ seja de fácil medida (ela traduz-se pela diferença entre a abcissa a ser considerada e a de referência), o mesmo não se poderá dizer do ângulo θ , cujo cálculo não é imediato. Contudo, conhecida a configuração do sistema de

projecção da linha laser e as suas medidas, assim como o valor do contador de passos enviados ao controlador do motor, é possível conhecer aquele θ .

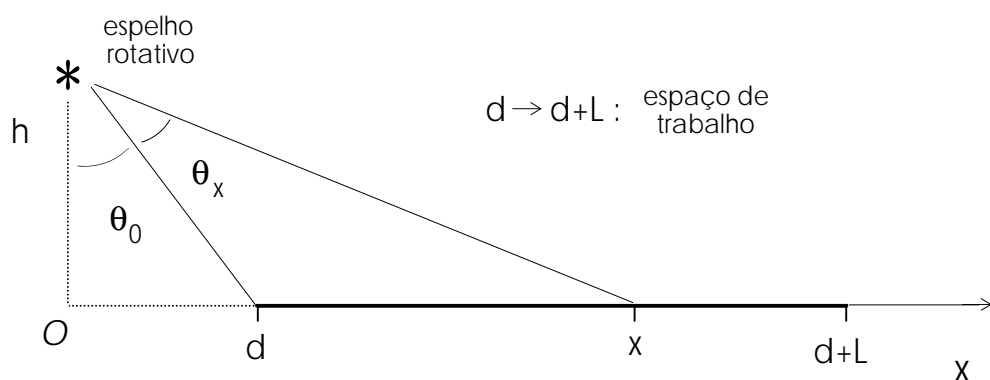


Fig. 10 - Indicação das posições relativas entre o espelho rotativo e o espaço de trabalho.

De acordo com a figura acima, θ_0 é o ângulo inicial de inclinação do feixe (quando este se encontra na posição de referência) dado por $\arctg(d/h)$ e θ_x é o ângulo acrescido por rotação do espelho, ou ainda pelo motor passo-a-passo. Este valor pode ser calculado conhecidas as características deste, assim como da caixa redutora, e ainda do nº de passos já enviados ao controlador do motor mantido sempre em memória com o nome de *counter*. Se *angle_per_step* for o ângulo de rotação mínimo do motor (para um incremento) e *gear* for o factor de desmultiplicação da caixa redutora, então Z será dado por,

$$Z = \frac{\text{shift} \times \text{pixel_to_mm}}{\text{tg} \left(\arctg(d/h) + \text{counter} \times \frac{\text{angle_per_step}}{\text{gear}} \right)}$$

O desvio *shift* é medido relativamente à linha não obstruída, que pode ser encontrada numa das, ou em ambas as extremidades da linha laser na imagem. Para cada uma destas linhas é então definido um valor X_{ref} que será o valor médio das abcissas da linha na parte superior e inferior, desde que não difiram de um valor muito elevado (<6 pixel). Neste último caso, é tomado para X_{ref} o menor dos dois valores encontrados, significando que o outro já se refere à linha obstruída.

Para cada linha projectada na cena, é feita uma amostragem de p_per_line alturas Z que são registadas em memória conjuntamente com as respectivas coordenadas da imagem (x,y) . Este registo de (x,y) deve-se ao facto de não ser possível garantir deslocamentos laterais rigorosamente constantes da linha laser e por consequência não se obter uma densidade rigorosamente igual de pontos de medida para toda a cena.

Os seus vários valores são então registados sequencialmente em memória RAM tal como é ilustrado nas Figs. 11 e 12 :

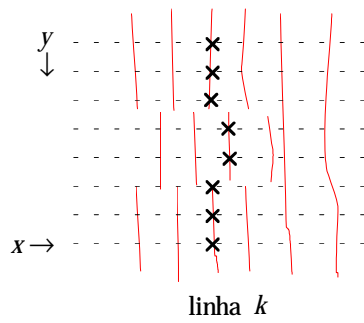


Fig. 11 - A amostragem de alturas Z (as linhas são mostradas em simultâneo embora sejam projectadas em sequência).

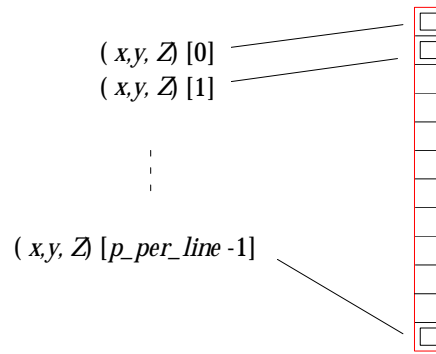


Fig. 12 - Registo das estruturas associadas aos Zs sequencialmente em memória RAM.

Na Fig.13 é mostrada a sequência de registo em memória RAM e em disco da informação associada às várias linhas laser projectadas.

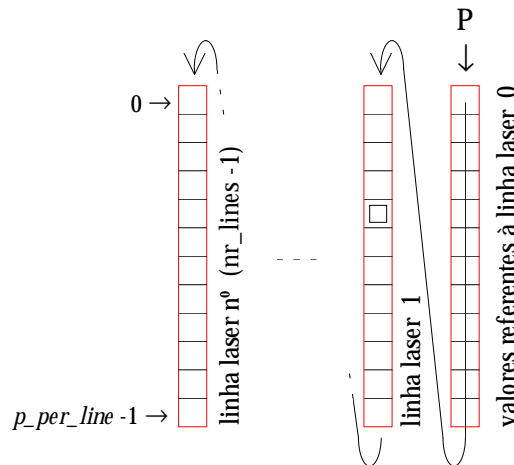


Fig. 13 - Registo em sequência dos conjuntos de estruturas associados a cada linha laser, na memória RAM. P é o apontador para o início do registo.

9.3.1 O Registo em Memória acima dos 64KB

Uma forma de verificar a correcta amostragem das alturas Z é efectuar a sua visualização em monitor. Foi então escrita uma função que a partir do conhecimento do endereço do 1º registo em memória e do nº de registos contidos no vector, efectua uma leitura sequencial das estruturas mencionadas e regista no ponto de coordenadas (x,y) um valor proporcional à altura registada Z da estrutura. Esse valor que deverá pertencer ao intervalo [0,255] será dado por,

$$gray_level = 255 \times \frac{Z}{Z_{m\acute{a}x}}$$

onde $Z_{m\acute{a}x}$ é o valor máximo de Z encontrado no vector.

Ora, no decurso das primeiras experiências de visualização constatou-se a ocorrência de anomalias que posteriormente se verificou dever-se ao grande volume de informação a ser registado em RAM. De facto, a quantidade de memória requerida excedia frequentemente os 64KB, valor máximo permitido pelo modelo de memória utilizado ("Large"). Sendo assim, houve necessidade, sempre que aquele máximo era atingido, de desdobrar o valor do apontador para memória em valor de Segmento e Deslocamento ("Offset") (Fig.15) e forçar o incremento daquele. Da recominação de ambos (Fig.16) resultava um apontador para memória acima daqueles 64KB. Conseguiu-se assim ultrapassar aquela limitação e obter uma maior capacidade de armazenamento (Fig.14).

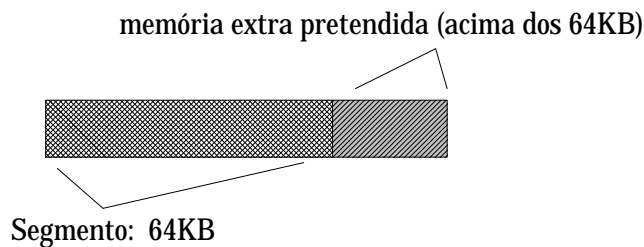


Fig. 14 - Registo sequencial das estruturas com uso de memória acima dos 64KB.

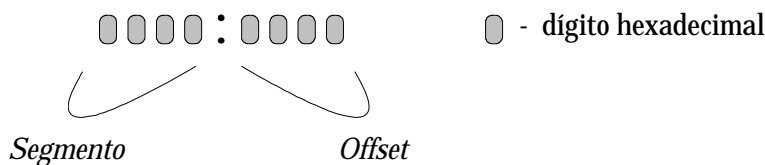


Fig. 15 - Extração do Segmento e Offset de um apontador.

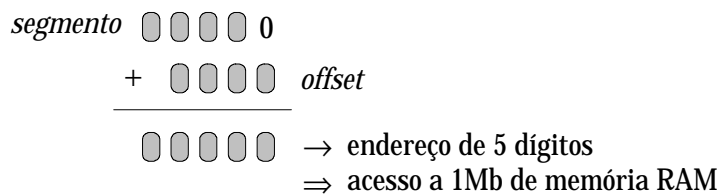


Fig. 16 - Avaliação (interna) de um endereço a partir do Segmento e do Offset (recomposição do apontador).

9.3.2 O Registo em Disco do Vector de Estruturas

A necessidade de reutilização posterior dos valores de Z (por exemplo para representação gráfica bi e tridimensional) levou a que se incluísse no algoritmo de medição de alturas, uma secção responsável pelo registo em disco das estruturas. Para que os valores fossem legíveis por aplicações gráficas, optou-se inicialmente pelo registo em modo texto, contendo cada linha uma sequência Z , y e x . No início deste ficheiro, foi ainda incluído um conjunto de três inteiros: a resolução da amostragem (res), o nº de linhas amostradas (n_lines) e por fim o nº de pontos amostrados por cada linha (p_per_line). O produto destes dois últimos inteiros determina o comprimento do ficheiro a ser registado; eles são absolutamente necessários aquando da leitura, para reservar memória RAM. Foi ainda escrito um programa para leitura e mostra em monitor.

Como alternativa ao processo anterior de registo, foi ainda escrito um outro programa que efectua o registo binário das mesmas estruturas. Este tipo de registo possui a vantagem de ocupar menos espaço em disco e ser substancialmente mais rápido que o anterior, quer na leitura quer na escrita. Ele é seleccionado quando se pretende registar o vector apenas para posterior mostra em monitor (com a finalidade apenas de o visualizar ou para processamentos posteriores a partir do ficheiro de imagem, como por exemplo, na construção de imagens de distância com resolução máxima, como se verá em capítulos posteriores). Não tem portanto utilidade para aplicações gráficas como no caso anterior, mas tão somente para registo. Este ficheiro contém no seu início os três valores inteiros do tipo anterior de ficheiro (em modo texto para facilidade de inspecção). Para maximizar a velocidade de registo, foram considerados blocos de 32KB que eram transferidos para o disco de uma só vez, sendo feita a verificação do "Offset" (para eventual incremento do Segmento) apenas após cada um destes registos.

9.3.3 A Codificação das Alturas no "Display", por Côr

Por uma questão de facilidade de interpretação de resultados achou-se conveniente atribuir côr aos pixels, de acordo com o seu conteúdo que é proporcional à altura Z medida. Tratando-se agora apenas de uma visualização (aliás não há sequer aquisição de imagem através da câmara) selecciona-se a LUT de saída, cuja acção ocorre apenas na mostra em monitor. Tal como acontece com a LUT de entrada, também esta pode ser programada, com a particularidade útil de permitir fazê-lo para cada uma das três cores RGB: vermelho, verde e azul. Cada LUT de saída desdobra-se então em três, uma para cada côr.

Se a cada côr for associado um dado intervalo de níveis de cinzento do registo do pixel, então consegue-se obter a pretendida codificação por côr. Cada uma daquelas LUTs é programada separadamente como a seguir se ilustra.

Por exemplo, para a LUT 6 de saída, a sua activação é feita por selecção no registo de controle das LUTs (escrita de 192_D em 341_H). A programação em separado para cada uma das 3 cores, por exemplo para os gráficos apresentados na Fig.17, é ilustrada nas Tabelas 2, 3 e 4.

Tabela 2

Acção		Escrita de
<i>programação da LUT de saída 6 referente ao Azul:</i>		$192_D + 2$ (em 341_H)
registro de valor crescente entre 0 e 63: registro em 342_H (registro do endereço) registro em 343_H (registro do valor)		i ($i \in [0,64[$) $4 \times i$
registro do valor 255 de 64 a 127: registro em 342_H (registro do endereço) registro em 343_H (registro do valor)		i ($i \in [64,128[$) 255
registro de valor decrescente de 128 a 191: registro em 342_H (registro do endereço) registro em 343_H (registro do valor)		i ($i \in [128,192[$) $766 - 4 \times i$
registro do valor 0 de 192 a 255: registro em 342_H (registro do endereço) registro em 343_H (registro do valor)		i ($i \in [192,256[$) 0

Tabela 3

Acção		Escrita de
<i>programação da LUT de saída 6 referente ao Verde:</i>		$192_D + 1$ (em 341_H)
registro do valor 0 entre 0 e 63: registro em 342_H (registro do endereço) registro em 343_H (registro do valor)		i ($i \in [0,64[$) 0
registro de valor crescente de 64 a 127: registro em 342_H (registro do endereço) registro em 343_H (registro do valor)		i ($i \in [64,128[$) $-254 + 4 \times i$
registro do valor 255 de 128 a 191: registro em 342_H (registro do endereço) registro em 343_H (registro do valor)		i ($i \in [128,192[$) 255
registro de valor decrescente de 192 a 255: registro em 342_H (registro do endereço) registro em 343_H (registro do valor)		i ($i \in [192,256[$) $1022 - 4 \times i$

Tabela 4

Acção	Escrita de
<p><i>programação da LUT de saída 6 referente ao Vermelho:</i></p> <p>registo do valor 0 entre 0 e 127: registo em 342_H (registo do endereço) registo em 343_H (registo do valor)</p> <p>registo de valor crescente de 128 a 191: registo em 342_H (registo do endereço) registo em 343_H (registo do valor)</p> <p>registo do valor 255 de 192 a 255: registo em 342_H (registo do endereço) registo em 343_H (registo do valor)</p>	<p>192_D +0 (em 341_H)</p> <p>i ($i \in [0,128[$) 0</p> <p>i ($i \in [128,192[$) -510+4<i>i</i></p> <p>i ($i \in [192,256[$) 255</p>

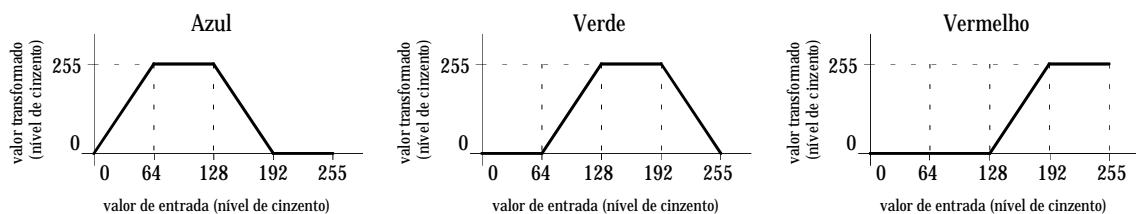


Fig. 17 - Gráficos de conversão da LUT de saída 6 para as 3 cores.

9.4 O Efeito de Perspectiva: A Correção da Fórmula de Z

A fórmula para o cálculo de Z deduzida numa das secções anteriores, embora correcta, não contempla o efeito de perspectiva derivado da localização da câmara. De facto se a câmara for deslocada da sua posição inicial, qualquer que ela seja, a parte desviada da linha laser por obstrução de um objecto, é "vista" (pela câmara) a deslocar-se também, dando origem a medidas diferentes de *shift*. Contudo, à parte a deslocação da câmara, nada se modifica pelo que a medida de Z deve manter-se constante. Ora, tal só é atingido se a quantidade " $Z.tg\alpha$ " for subtraída a " $shift \times pixel_to_mm$ ", conforme se pode observar na Fig.18. No caso da figura referida, se a posição da câmara for a vertical do objecto (situação na qual a correcção a efectuar é aproximadamente nula) e ela se afastar horizontalmente dessa posição, então o valor de *shift* deixa de ser a distância medida (em unidades de pixel) entre os pontos 2 e 3 e passa a ser medida entre os pontos 1 e 3. De referir que todas as linhas são vistas projectadas no plano $Z=0$ e este facto será importante na avaliação, que será feita na secção seguinte, do factor de conversão entre pixeis e milímetros.

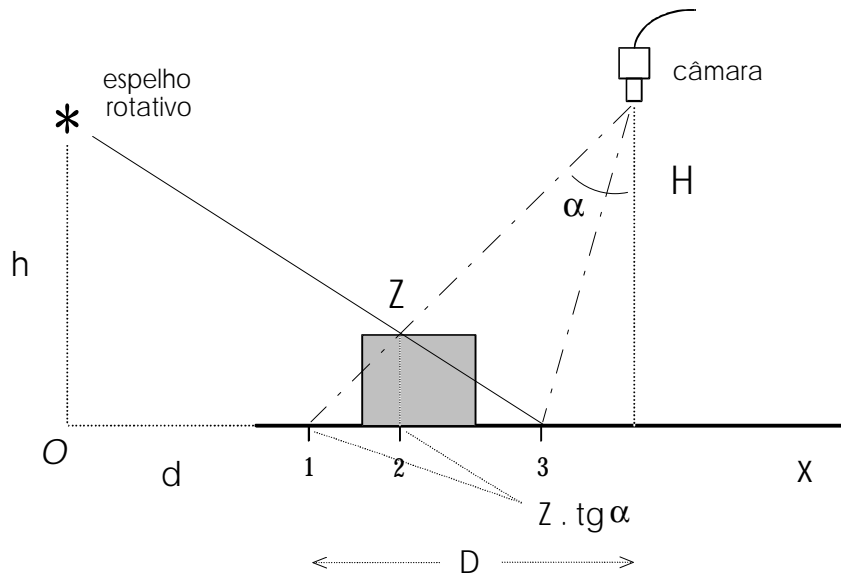


Fig. 18 - A incorrecção na medida de Z devido ao efeito de perspectiva.

A expressão de Z terá então a seguinte forma:

$$Z = \frac{\text{shift} \times \text{pixel_to_mm} - Z \times \text{tg}\alpha}{\text{tg}\left(\arctg(d/h) + \left(\text{counter} \times \frac{\text{angle_per_step}}{\text{gear}}\right)\right)}$$

Ainda, atendendo a que $\text{tg}\alpha = D/H$ e reorganizando a expressão, obtém-se o seguinte resultado final:

$$Z = \frac{\text{shift} \times \text{pixel_to_mm}}{\text{tg}\left(\arctg(d/h) + \left(\text{counter} \times \frac{\text{angle_per_step}}{\text{gear}}\right)\right) + \frac{D}{H}}$$

H é de fácil medição: consiste na distância entre a câmara e o plano $Z=0$ (mais rigorosamente consiste na distância entre o centro da câmara no modelo "pin-hole" e o plano $Z=0$, mas esse valor não difere apreciavelmente do anteriormente referido). De mais problemática avaliação será o factor D já que envolve a conversão entre pixeis e milímetros e em pontos diferentes pelo que o factor de conversão virá inclusivamente modificado dum ponto para o outro. Contudo, se D for correctamente avaliado, então a parcela D/H será essencial para uma correcta medição das alturas Z , aliás confirmada aquando da calibração do medidor laser (e cuja descrição será feita em secção posterior).

9.5 A Distorção da Imagem e a Avaliação do Factor Variável *pixel_to_mm*

Pixel_to_mm é, como se verificou, o factor de conversão entre pixeis e milímetros sendo aqueles medidos ao longo da coordenada x da imagem, qualquer que seja o ponto considerado. Experimentalmente constata-se que aquele factor não é constante havendo portanto a necessidade de o medir de uma forma sistematizada e encontrar uma função em x e y que forneça aquele valor.

Embora a variação de *pixel_to_mm* se verificasse em toda a imagem, iniciou-se a medida no centro do ecrã, ou seja, para $y \approx 256$. De frisar que qualquer que seja o ponto da imagem onde aquele valor seja medido, a conversão refere-se sempre à medida do pixel feita segundo x , pois é nessa direcção que *shift* é medido.

A correspondência entre *pixel* (segundo x) e mm foi feita usando uma escala milimétrica colocada inicialmente, como já foi dito, na linha $y \approx 256$; com a ajuda de um programa que permitia mover um ponto luminoso na imagem, fazia-se coincidi-lo com vários traços da escala. Sempre que uma dessas coincidências era obtida, registava-se a abcissa daquele ponto luminoso. Fazendo um registo para vários desses pontos da escala tais que cobrissem toda a extensão da imagem segundo x , obtinha-se uma lista completa de correspondência entre *pixel* e mm. O intervalo mínimo usado foi de 10mm. A lista é a apresentada na Tabela 5.

Tabela 5

Pixel	mm		Pixel	mm		Pixel	mm
7,0	0		216,0	260		434,0	520
15,0	10		224,5	270		441,5	530
22,5	20		232,5	280		450,0	540
31,0	30		241,0	290		458,0	550
39,0	40		249,5	300		466,5	560
47,0	50		258,0	310		475,0	570
55,0	60		266,0	320		483,0	580
62,0	70	centro	275,0	330	centro	491,5	590
70,0	80		283,0	340		499,5	600
78,0	90		291,5	350		508,0	610
86,0	100		299,5	360		516,0	620
94,0	110		308,0	370		524,0	630
103,0	120		316,0	380		533,0	640
111,0	130		324,0	390		541,5	650
119,0	140		333,0	400		549,5	660
126,5	150		341,5	410		557,0	670
134,5	160		350,0	420		565,0	680
143,0	170		358,0	430		573,0	690
150,0	180		366,5	440		581,0	700
159,0	190		375,0	450		589,0	710
167,0	200		383,5	460		597,0	720
175,0	210		392,0	470		605,0	730
183,0	220		400,0	480		613,5	740

192,0	230
200,0	240
208,0	250

408,5	490
417,0	500
425,0	510

621,0	750
629,0	760

Pela análise do gráfico da Fig.19 pode-se verificar que a escala milimétrica cresce mais rapidamente que a de x , pelo que o factor $pixel_to_mm$, que não é mais do que a derivada de mm em ordem a $x \equiv pixel$, é superior à unidade.

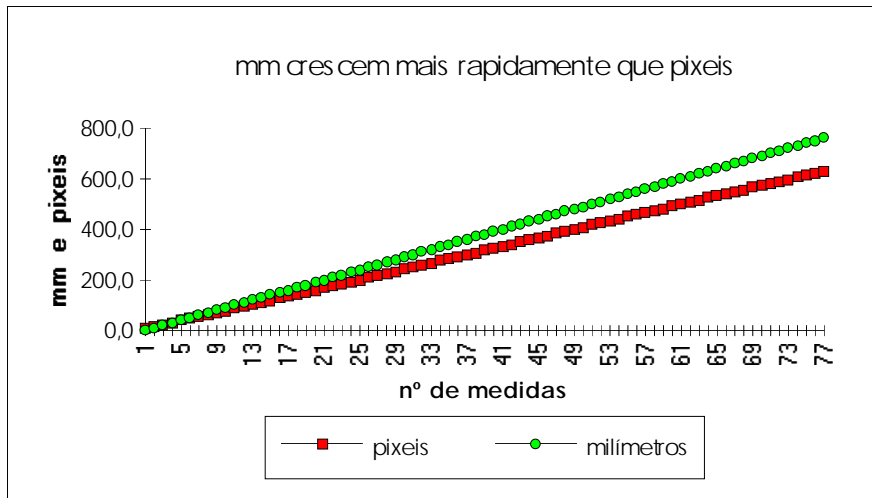


Fig. 19 - Crescimento de mm e $pixel$ com o nº de medidas.

Pela representação gráfica de mm em função de x , pode-se também verificar que a relação é aproximadamente linear com um declive de 1,212 $mm/pixel$, de acordo com a Tabela 6 e a Fig.20.

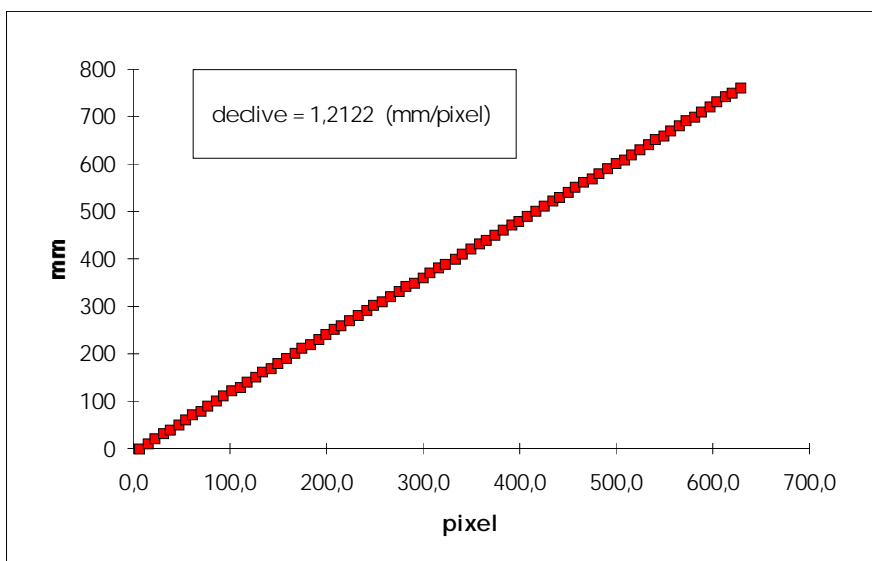


Fig. 20 - Representação de mm em função de $x \equiv pixel$.

Tabela 6

p_to_mm (70 mm)	p_to_mm (150 mm)		simétricos (70 mm)	simétricos (150 mm)	p_to_mm (70 mm)	p_to_mm (150 mm)		simétricos (70 mm)	simétricos (150 mm)
0	0		0	0	1,188665	1,195747	c	1,195025	1,196644
0	0		0	0	1,188665	1,194975		1,198974	1,196644
0	0		0	0	1,183726	1,19321		1,201841	1,195633
1,247074	0		0	0	1,183726	1,192199		1,193905	1,194485
1,260698	0		1,253886	0	1,191425	1,190185		1,195199	1,194249
1,269169	0		1,256745	0	1,188879	1,190939		1,192643	1,194884
1,284079	0		1,262837	0	1,191381	1,188918		1,190109	1,194515
1,284079	1,253986		1,26704	0	1,188879	1,191678		1,187596	1,195769
1,278139	1,252867		1,268235	1,248027	1,191425	1,195514		1,190109	1,198575
1,249004	1,251473		1,256508	1,245536	1,199061	1,198314		1,197777	1,200104
1,222358	1,250063		1,240294	1,242361	1,193862	1,199852		1,196439	1,201259
1,217115	1,250612		1,226229	1,241678	1,201429	1,200109		1,206696	1,20294
1,224873	1,246722		1,222044	1,237296	1,201429	1,202952		1,213339	1,206047
1,235535	1,24288		1,222229	1,233507	1,211826	1,204514		1,215799	1,207759
1,241105	1,239085		1,218712	1,229352	1,214482	1,206593		1,214482	1,210748
1,269482	1,239633		1,235434	1,227123	1,209193	1,207115		1,213154	1,212866
1,257824	1,236856		1,233553	1,224045	1,214482	1,207894		1,215799	1,214987
1,246579	1,235201		1,233268	1,221405	1,20165	1,210765		1,204094	1,218812
1,235727	1,234652		1,22389	1,22113	1,206807	1,209973		1,214396	1,220286
1,235727	1,235201		1,219972	1,221405	1,204218	1,207609		1,219972	1,221405
1,221985	1,230599		1,214396	1,220286	1,212053	1,207609		1,22389	1,22113
1,206538	1,22686		1,204094	1,218812	1,219957	1,207609		1,233268	1,221405
1,217115	1,22208		1,215799	1,214987	1,209283	1,211235		1,233553	1,224045
1,217115	1,218617		1,213154	1,212866	1,201385	1,214613		1,235434	1,227123
1,214482	1,214903		1,214482	1,210748	1,196319	1,21962		1,218712	1,229352
1,219771	1,211004		1,215799	1,207759	1,208922	1,224134		1,222229	1,233507
1,225248	1,209143		1,213339	1,206047	1,219216	1,227871		1,222044	1,237296
1,211962	1,20577		1,206696	1,20294	1,235342	1,232743		1,226229	1,241678
1,199017	1,202666		1,196439	1,201259	1,25823	1,234659		1,240294	1,242361
1,196494	1,201895		1,197777	1,200104	1,264012	1,239599		1,256508	1,245536
1,188794	1,201636		1,190109	1,198575	1,258332	1,243187		1,268235	1,248027
1,186313	1,19986		1,187596	1,195769	1,25	0		1,26704	0
1,188836	1,200113		1,190109	1,194515	1,241594	0		1,262837	0
1,196406	1,19883		1,192643	1,194884	1,244322	0		1,256745	0
1,198973	1,198314		1,195199	1,194249	1,247074	0		1,253886	0
1,204084	1,196771		1,193905	1,194485	0	0		0	0
1,219957	1,198056		1,201841	1,195633	0	0		0	0
1,209283	1,198314		1,198974	1,196644	0	0		0	0
1,201385	1,197541	c	1,195025	1,196644					

O gráfico das derivadas locais (que fornecem os valores locais do factor *pixel_to_mm*) revela contudo que essa linearidade é apenas aparente, apresentando na realidade valores ligeiramente inferiores no centro da imagem como se pode observar nas Figs. 21 e 22.

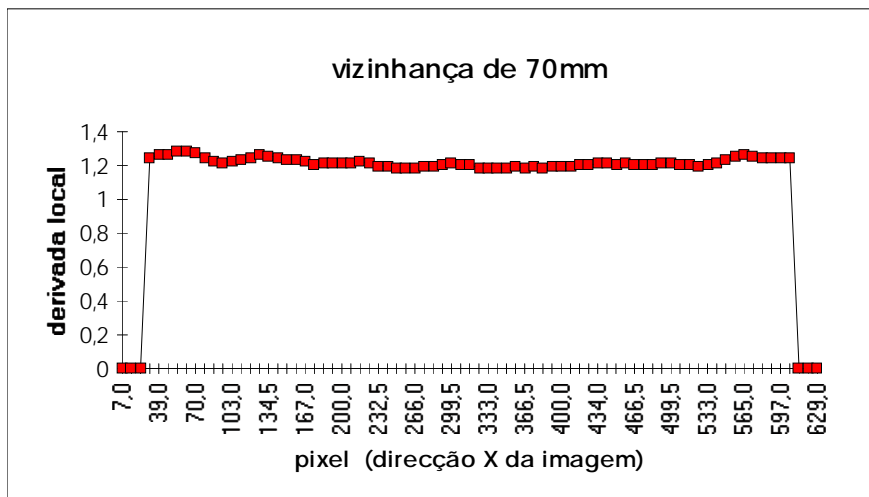


Fig. 21 - Representação das derivadas locais para uma vizinhança de cálculo de 70mm.

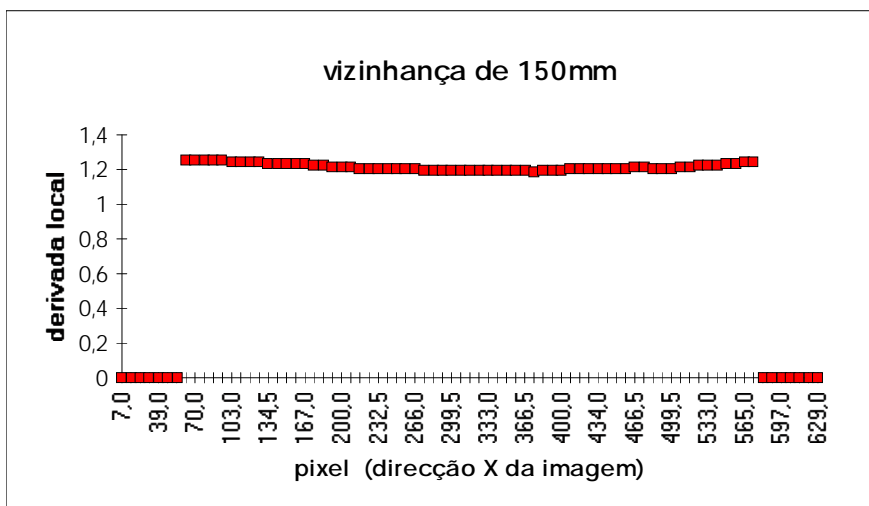


Fig. 22 - Representação das derivadas locais para uma vizinhança de cálculo de 150mm.

Naqueles dois gráficos foram utilizadas vizinhanças de 70mm e 150mm, dentro das quais a variação de mm foi considerada linear com $x \equiv \text{pixel}$ e a melhor recta foi ajustada por regressão linear. Pode-se verificar que para o 2º caso (150mm), a função obtida é mais bem comportada. Decidi contudo manter os cálculos e gráficos para o 1º caso (70mm) para obter resultados de referência. A oscilação observada para este caso deve-se exclusivamente ao carácter descontínuo do pixel e de não haver precisão visual abaixo dos 5mm o que impede de estabelecer correspondências perfeitas (de referir que a escolha do valor de mm correspondente aos vários valores de x foi feita por

observação directa da imagem e do ponto luminoso nela sobreposto, este gerado por um programa auxiliar).

Aquela variação de *pixel_to_mm* deve-se decerto à distorção da imagem provocada pelo sistema óptico da câmara, conhecida por aberração esférica. Sendo essa distorção radial é de supor que aquele gráfico seja simétrico em relação ao centro da imagem. E ele é-o de facto. Contudo, devido a erros de observação aleatórios, verificam-se pequenas diferenças entre os ramos esquerdo e direito da função, que só são rigorosamente equalizados pela adopção dos valores médios entre pontos correspondentes dos dois ramos. Nessa operação foi considerado para centro da escala x o valor 320 (como seria obrigatório dado que a largura da imagem é de 640 pixels) a que corresponde o valor 385 na escala milimétrica. São os seguintes (Figs. 23 e 24) os gráficos já simetrizados em torno de $x=320$.

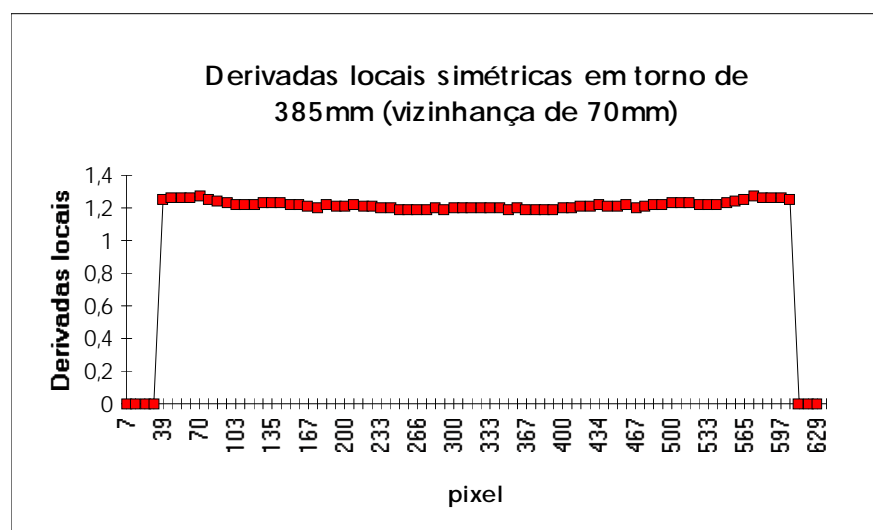


Fig. 23 - Simetrização do gráfico em torno de 385mm (vizinhança de 70mm).

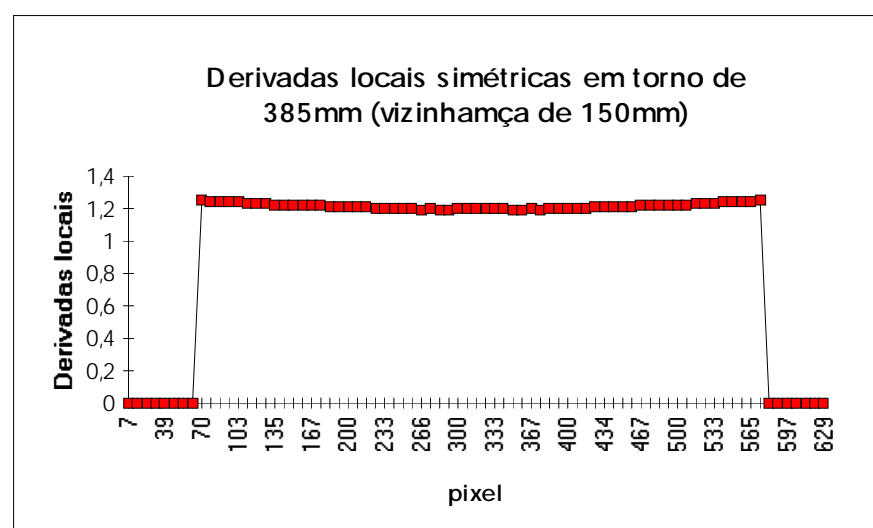
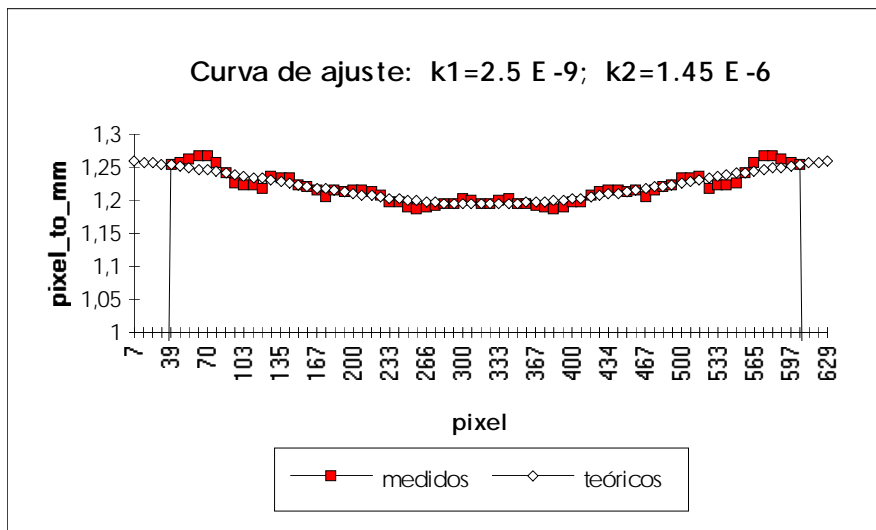


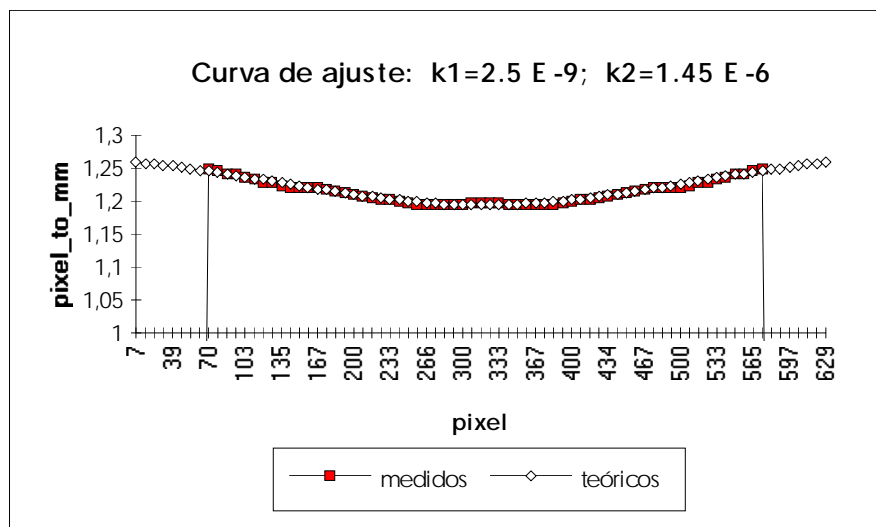
Fig. 24 - Simetrização do gráfico em torno de 385mm (vizinhança de 150mm).

A Fig.25 mostra essa mesma função ampliada de modo a poder observar-se as oscilações já justificadas e que a escolha da vizinhança de 150mm para o cálculo das derivadas locais, embora atenua, não faz desaparecer completamente. Ela só será ignorada após o ajuste da melhor função teórica, ajuste esse que é mostrado na Fig.25c. Esse processo foi bastante dispendioso em termos de tempo já que foi feito por tentativas e aproximações sucessivas; mas apesar disso, conseguiu-se obter um bom ajuste.

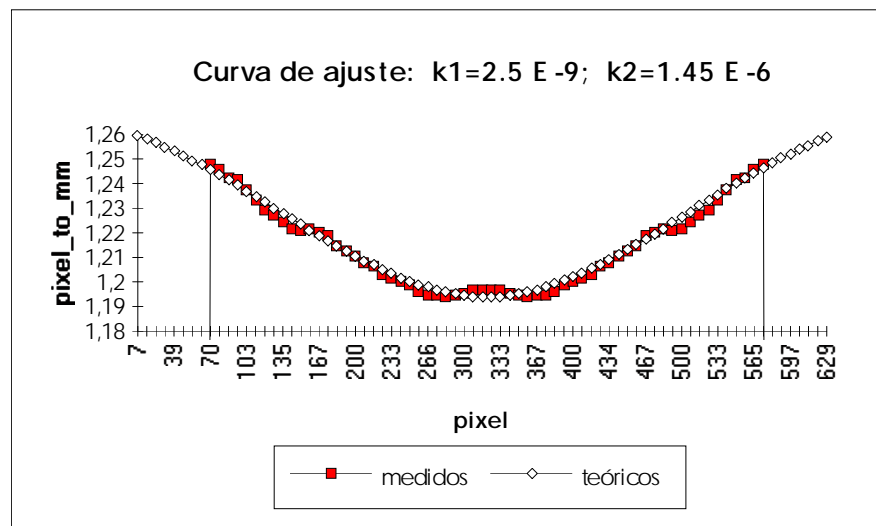
De notar que a melhor função não é uma parábola como seria de esperar, mas antes uma função cúbica.



(a)



(b)



(c)

Fig. 25 - Ajuste da melhor função aos pontos experimentais

- a) vizinhança de 70mm
- b) vizinhança de 150mm
- c) vizinhança de 150mm (gráfico ampliado)

A função que melhor se ajusta aos pontos experimentais é a seguinte:

$$pixel_to_mm = -2,5 \cdot 10^{-9} \cdot |\chi - 319,5|^3 + 1,45 \cdot 10^{-6} \cdot (\chi - 319,5)^2 + 1,194$$

O coeficiente $-2,5 \cdot 10^{-9}$ embora muito pequeno em relação aos restantes, é responsável pelo ligeiro abrandamento no crescimento de $pixel_to_mm$ perto dos extremos da imagem. Se se fizer uma translação do sistema de coordenadas de modo a que $x=0$ corresponda ao centro da imagem (em torno da qual se verifica a simetria na distorção da imagem), ou seja $x = \chi - 319,5$ a expressão fica:

$$pixel_to_mm = \begin{cases} x < 0 \Rightarrow 2,5 \cdot 10^{-9} \cdot x^3 + 1,45 \cdot 10^{-6} \cdot x^2 + 1,194 \\ x \geq 0 \Rightarrow -2,5 \cdot 10^{-9} \cdot x^3 + 1,45 \cdot 10^{-6} \cdot x^2 + 1,194 \end{cases}$$

Para o extremo direito, por exemplo, $pixel_to_mm=1,261$, que corresponde a uma variação de apenas 6% relativamente ao centro. Esta variação, embora pequena, não deve ser desprezada se se pretender obter bons resultados para as medidas de Z .

Uma medida idêntica efectuada também na direcção de x , mas agora ao longo da linha $y=0$, apresenta resultados muito idênticos diferindo apenas no termo constante da expressão como se poderá constatar dos registos e gráficos de ajuste seguintes (efectuou-se agora apenas para uma vizinhança de 150mm).

A tabela de correspondências encontrada foi a apresentada na secção seguinte.

9.5.1 Tabela de Correspondências em $y=0$ e Derivadas Locais

Tabela 7

$pixel = x$	mm	p_to_mm (150 mm)	p_to_mm simétricos	valores teóricos
14,0	0	0	0	1,272048
22,0	10	0	0	1,270508
29,0	20	0	0	1,269077
37,0	30	0	0	1,267356
44,5	40	0	0	1,265664
52,5	50	0	0	1,263784
60,0	60	0	0	1,261956
68,0	70	1,286665	1,260535	1,259946
75,5	80	1,284019	1,255955	1,25801
83,5	90	1,281417	1,253984	1,255899
91,0	100	1,281124	1,253837	1,253882
99,0	110	1,274933	1,251412	1,251697
107,0	120	1,269966	1,248125	1,249487
115,0	130	1,26422	1,243932	1,247259
123,0	140	1,263938	1,244998	1,245019
130,5	150	1,26253	1,243213	1,242917
138,0	160	1,260254	1,241805	1,240819
146,0	170	1,25543	1,23688	1,238591
155,0	180	1,252604	1,234675	1,236109
163,0	190	1,249802	1,233274	1,233931
171,0	200	1,247026	1,232825	1,231789
178,0	210	1,244274	1,231317	1,229949
186,0	220	1,241546	1,228747	1,227894
194,5	230	1,237124	1,224321	1,225773
203,0	240	1,235743	1,22324	1,223727
211,0	250	1,235468	1,221678	1,221877
219,0	260	1,228124	1,218527	1,220108
227,0	270	1,223888	1,215101	1,218428
235,0	280	1,218632	1,211426	1,216845
243,0	290	1,218632	1,211687	1,215367
252,0	300	1,220226	1,211971	1,213838
260,0	310	1,217554	1,211148	1,212607
268,0	320	1,214665	1,208413	1,211504
277,0	330	1,21336	1,207633	1,210427
284,5	340	1,213621	1,208793	1,209669
293,0	350	1,213621	1,209052	1,208972
301,0	360	1,211758	1,21045	1,20848
309,0	370	1,212289	1,210195	1,208157

318,0	380	centro	1,209143	centro	1,209143	1,208003
326,0	390		1,208101		1,210195	1,208061
334,0	400		1,209143		1,21045	1,208297
342,0	410		1,204483		1,209052	1,208706
350,5	420		1,203966		1,208793	1,209319
359,5	430		1,201905		1,207633	1,21016
367,5	440		1,202162		1,208413	1,211064
376,0	450		1,204742		1,211148	1,212178
384,0	460		1,203716		1,211971	1,213362
392,0	470		1,204742		1,211687	1,214669
401,0	480		1,20422		1,211426	1,216278
409,0	490		1,206314		1,215101	1,217823
417,5	500		1,20893		1,218527	1,219573
425,5	510		1,207889		1,221678	1,221315
433,5	520		1,210737		1,22324	1,22314
442,0	530		1,211519		1,224321	1,225163
450,0	540		1,215949		1,228747	1,227138
459,0	550		1,218359		1,231317	1,229431
467,0	560		1,218624		1,232825	1,231524
475,0	570		1,216746		1,233274	1,233661
483,5	580		1,216746		1,234675	1,235972
491,0	590		1,218329		1,23688	1,238037
499,5	600		1,223356		1,241805	1,2404
507,0	610		1,223895		1,243213	1,242497
516,0	620		1,226058		1,244998	1,245019
524,0	630		1,223645		1,243932	1,247259
533,0	640		1,226283		1,248125	1,249765
540,5	650		1,22789		1,251412	1,251835
548,5	660		1,22655		1,253837	1,254017
556,0	670		1,22655		1,253984	1,256032
565,0	680		1,22789		1,255955	1,258401
573,0	690		1,234404		1,260535	1,260454
581,5	700		1,237659		0	1,262572
588,5	710		0		0	1,264261
597,0	720		0		0	1,266236
605,5	730		0		0	1,26812
614,0	740		0		0	1,269904
621,5	750		0		0	1,271387
629,0	760		0		0	1,272778
637,0	770		0		0	1,274154

Tabela 7- Tabela com *mm*, *pixel*, derivadas locais e derivadas simetrizadas em torno de 380mm.

Aquela tabela contém já as derivadas locais para uma vizinhança de 150mm, assim como os valores simetrizados em torno de $x=318,0$. Mais uma vez se verifica que a escala milimétrica cresce mais rapidamente que a dos pixels segundo x . Na Fig.26 apresenta-se o gráfico das derivadas locais simetrizadas com ampliação.

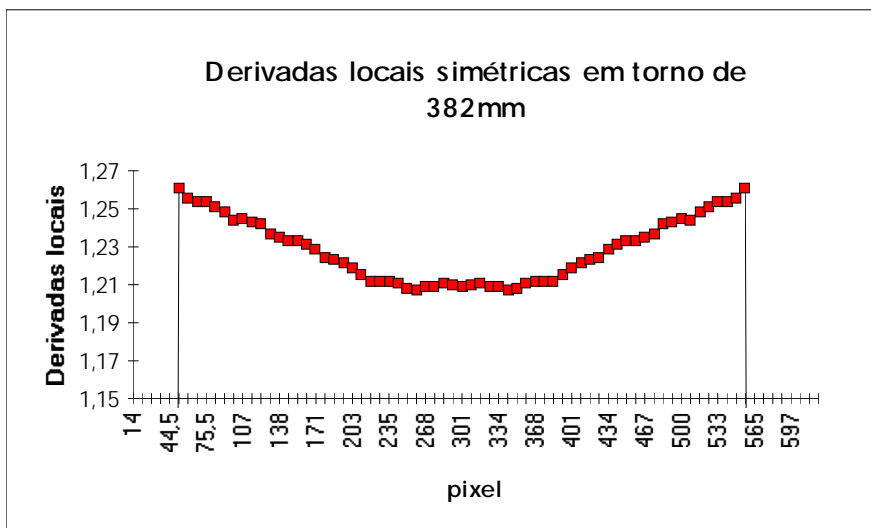


Fig. 26 - Gráfico das derivadas locais simetizadas com ampliação (vizinhança de 150mm), para y=0.

No gráfico seguinte (Fig.27) pode-se constatar que a função de melhor ajuste é semelhante à já obtida atrás:

$$pixel_to_mm = -2,5 \cdot 10^{-9} \cdot |x-319,5|^3 + 1,45 \cdot 10^{-6} \cdot (x-319,5)^2 + 1,208$$

diferindo apenas no termo constante que passa de 1,194 para 1,208.

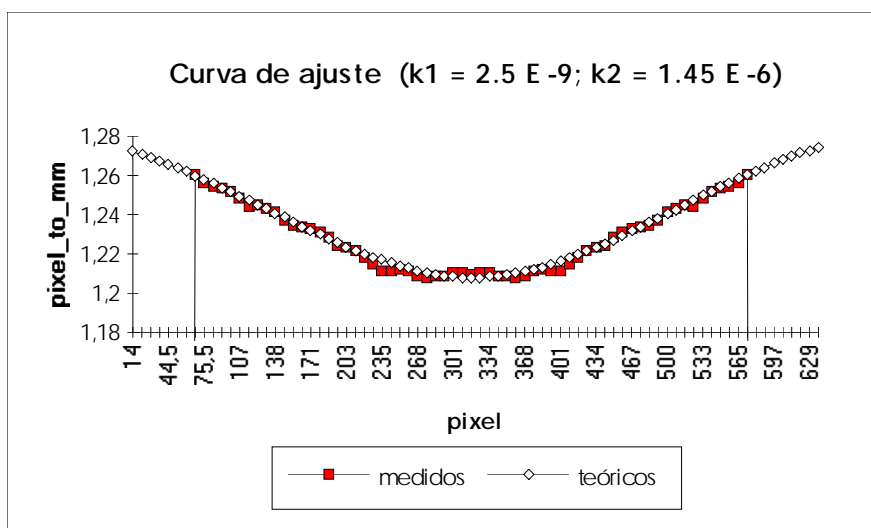


Fig. 27 - Curva de ajuste para pixel_to_mm referente à linha superior da imagem (y=0).

A função "teórica" será então a seguinte:

$$pixel_to_mm = -2,5 \cdot 10^{-9} \cdot |x-319,5|^3 + 1,45 \cdot 10^{-6} \cdot (x-319,5)^2 + y_factor$$

que corresponde ao gráfico final da Fig.28 (com $y_factor=1,194$).

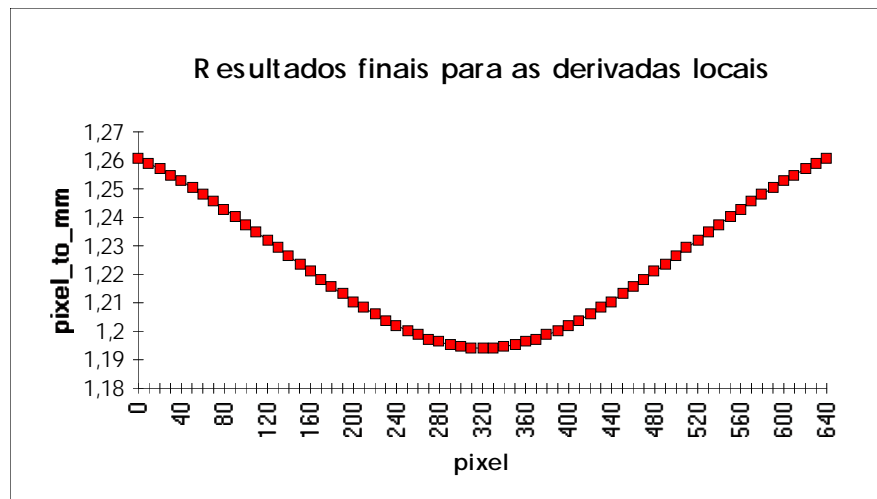


Fig. 28 - Função teórica para $pixel_to_mm$ (com $y_factor=1,194$).

Estes resultados sugerem então que se adopte para melhor função a atrás obtida, exceptuando o termo constante que será obrigado a variar entre aqueles dois valores para $y=255,5$ e $y=0$. Esse termo, constante em x mas que virá em função de y , pode ser dado por:

$$y_factor = 0,014 \times |1 - y/255,5| + 1,194$$

Esta função linear garante então uma variação suave com y , que somada com os termos em x fornece a função global para $pixel_to_mm$, ou seja, para todas as coordenadas da imagem:

$$pixel_to_mm = -2,5 \cdot 10^{-9} \cdot |x-319,5|^3 + 1,45 \cdot 10^{-6} \cdot (x-319,5)^2 + 1,194 + 0,014 \cdot |1 - y/255,5|$$

Se a abcissa nula for atribuída ao centro da imagem, então mais uma vez $x=x-319,5$ e a expressão vem simplificada.

Em face deste resultado pode-se obter uma representação tridimensional da função $pixel_to_mm(x,y)$ (Fig.29).

$$pixel_to_mm(x,y) = -2,5e-9 \cdot |x|^3 + 1,45e-6 \cdot x^2 + 1,194 \cdot x + 0,014 \cdot \left|1 - \frac{y}{255,5}\right|$$

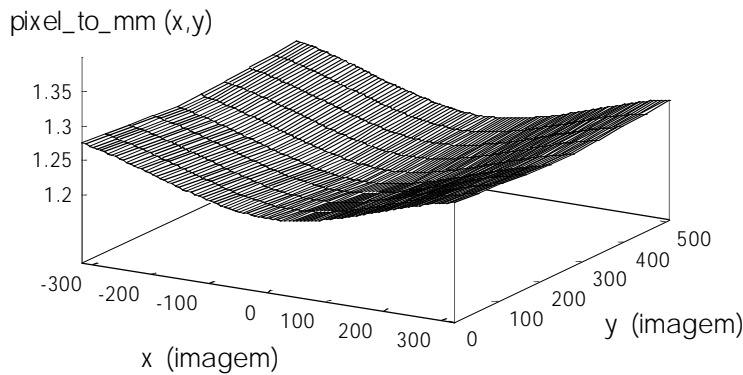


Fig. 29 - Representação de $pixel_to_mm$ como função de x e y .

Para finalizar esta secção far-se-á ainda referência ao modo de calcular a quantidade $D(x,y)$ indicada na Fig.18.

Estando já de posse de $pixel_to_mm$ para todo o (x,y) e sabendo que aquele valor corresponde à derivada da escala milimétrica em ordem a $x \equiv pixel$, ou seja $d(mm)/dx$, pode-se então obter por integração a relação entre mm e $x \equiv pixel$ (mantendo a origem de y na linha superior da imagem mas efectuando a translação da origem de x da coluna esquerda para a coluna central, ou seja, fazendo $x=x-319,5$).

$$mm(x,y) = \begin{cases} x < 0 \Rightarrow & 2,5 \cdot 10^{-9} \cdot \frac{x^4}{4} + 1,45 \cdot 10^{-6} \cdot \frac{x^3}{3} + 1,194 \cdot x + 0,014 \cdot \left|1 - \frac{y}{255,5}\right| \cdot x \\ x \geq 0 \Rightarrow & -2,5 \cdot 10^{-9} \cdot \frac{x^4}{4} + 1,45 \cdot 10^{-6} \cdot \frac{x^3}{3} + 1,194 \cdot x + 0,014 \cdot \left|1 - \frac{y}{255,5}\right| \cdot x \end{cases}$$

Conhecidas as duas abcissas em unidades de pixel (na Fig.18 uma delas será 0 uma vez que corresponde ao centro da imagem) pode-se então obter a sua distância real em mm através da diferença entre duas expressões do tipo apresentado atrás:

$$D(x,y) = mm(x,y) - mm(0,y) = mm(x,y)$$

com $x=0$ no centro da imagem obtido por $x=x-319,5$ e $y=0$ correspondente à linha superior.

A tabela teórica de correspondências entre $x \equiv \text{pixel}$ e D é mantida em ficheiro do qual se extraiu parte para análise do sinal deste:

Tabela 8

x	$mm(x)$	$D(x)$	$\tan(\alpha)$
0	-391,364	-391,364	-0,33738
10	-378,767	-378,767	-0,32652
20	-366,188	-366,188	-0,31568
30	-353,628	-353,628	-0,30485
40	-341,089	-341,089	-0,29404
310	-11,9405	-11,940	-0,01029
320	0	0,000	0,00000
330	11,94048	11,940	0,01029
590	328,5719	328,572	0,28325
600	341,0885	341,089	0,29404
610	353,6275	353,628	0,30485
620	366,1875	366,188	0,31568
630	378,767	378,767	0,32652

Na fórmula de Z , $shift \times pixel_to_mm$ foi subtraído por $D(x)/H = \tan \alpha$. Ora de acordo com a Fig.18 essa quantidade deve ser positiva para $x < 320$ e negativa para $x > 320$ (sendo 0 para $x = 320$). Mas a orientação para x na figura é inversa ao x da imagem dada pela câmara pelo que $x \rightarrow 640 - x$ e os sinais trocam na comparação de x com 320, donde que os resultados mostrados na tabela apresentam todos o respectivo sinal correcto.

9.6 A calibração do Sistema Laser: A Avaliação dos Parâmetros Mecânicos e Geométricos

Na secção anterior foi referido que havia uma certa dificuldade na medição de alguns parâmetros geométricos contidos na fórmula da altura Z . Mais concretamente, trata-se das medidas de h e d , ou seja da distância do espelho rotativo ao plano $Z=0$ e a distância entre a posição de referência da linha laser e a vertical do mesmo espelho medida na direcção de varrimento daquela linha. Ora, conhecida a fórmula para a medição das alturas e perante o conhecimento agora explícito do parâmetro variável $pixel_to_mm$ (obtido na secção anterior) e de $D(x,y)$, é possível calcular os valores desconhecidos, aplicando-a a objectos de altura conhecida. Z é dado pela fórmula:

$$Z = \frac{shift \times pixel_to_mm}{\operatorname{tg}\left(\arctg(d/h) + \left(counter \times \frac{angle_per_step}{gear}\right)\right) + \frac{D}{H}}$$

Mas Z é agora conhecido pois trata-se da altura de um objecto de teste cuja altura é conhecida. A introdução de um par de valores ($counter, shift$) que constituem agora as variáveis, não permite obter valores precisos nem confirmar a validade da introdução de parâmetros de correcção como $D(x,y)/H$ (informação de perspectiva). Sendo assim, deve-se fazer variar $counter$ (ou seja enviar números fixos de incrementos do motor passo-a-passo) e medir os valores de $shift$ correspondentes (desvios observados na imagem da linha laser por incidência do feixe no objecto, em unidades de pixel). Ora, por linearização da função e ajuste da melhor recta por regressão linear é possível conhecer finalmente os parâmetros geométricos desconhecidos. Mais ainda, a disposição linear dos pontos experimentais, após linearização da expressão, vem confirmá-la como válida; e em particular vem confirmar a validade da introdução do factor de correcção $D(x,y)/H$ devido a efeitos de perspectiva (mais adiante se verá que se este factor não for introduzido os pontos experimentais não se alinham ao longo de uma recta como previsto).

A linearização da expressão é obtida da seguinte forma: agora que Z é constante, $pixel_to_mm(x,y)$ e $D(x,y)$ possuem fórmulas explícitas que fornecem os seus valores conhecidas as coordenadas de imagem (x,y) deve-se separar devidamente as novas variáveis $counter$ e $shift$. Então rearranjando a fórmula como se apresenta a seguir e efectuando uma substituição de variáveis, consegue-se obter uma relação linear:

$$\operatorname{tg}\left(\arctg\left(\frac{d}{h}\right) + counter \times \frac{angle_per_step}{gear}\right) + \frac{D}{H} = shift \times pixel_to_mm \Leftrightarrow$$

$$\Leftrightarrow \arctg\left(\frac{d}{h}\right) + counter \times \frac{angle_per_step}{gear} + \frac{D}{H} = \arctg\left(shift \times pixel_to_mm - \frac{D}{H}\right)$$

Ora, se fizer,

$$\begin{cases} Y = \arctg\left(shift \times pixel_to_mm - \frac{D}{H}\right) \\ X = counter \end{cases}$$

obtem-se uma relação linear $Y = m.X + b$, onde o declive m e a ordenada na origem b serão dados por,

$$\begin{cases} m = \frac{angle_per_step}{gear} \\ b = \arctg\left(\frac{d}{h}\right) \end{cases}$$

m fornece então os parâmetros mecânicos e b os geométricos. O valor de $angle_per_step/gear$ é conhecido pois depende das características do motor passo-a-passo e da caixa redutora: $\pi/12.000$. Contudo, a adopção para aquele factor do valor obtido na calibração (desde que não difira muito do valor conhecido, pois tal indicaria a existência de um qualquer erro grave) permite compensar quaisquer erros experimentais que surjam, que, embora pequenos, sejam difíceis de identificar e corrigir.

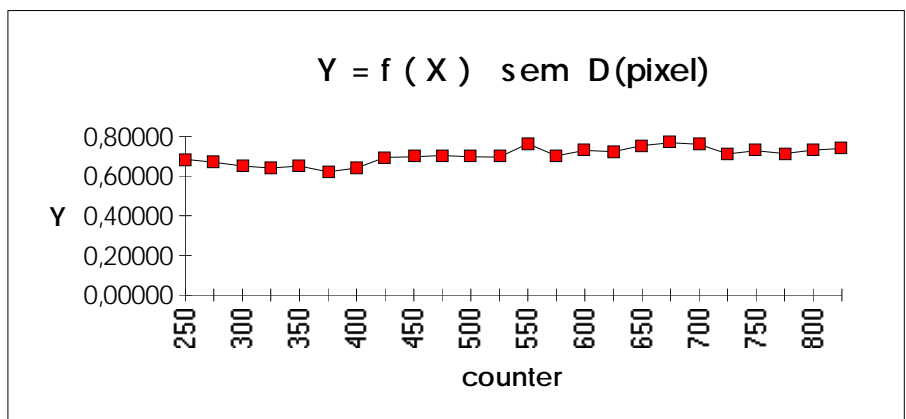
São então os seguintes os resultados experimentais obtidos para 6 objectos cujas alturas são $h_1=10,0\text{mm}$, $h_2=20,2\text{mm}$, $h_3=30,4\text{mm}$, $h_4=40,6\text{mm}$, $h_5=50,8\text{mm}$ e $h_6=61,0\text{mm}$.

9.6.1 Experiências de Calibração

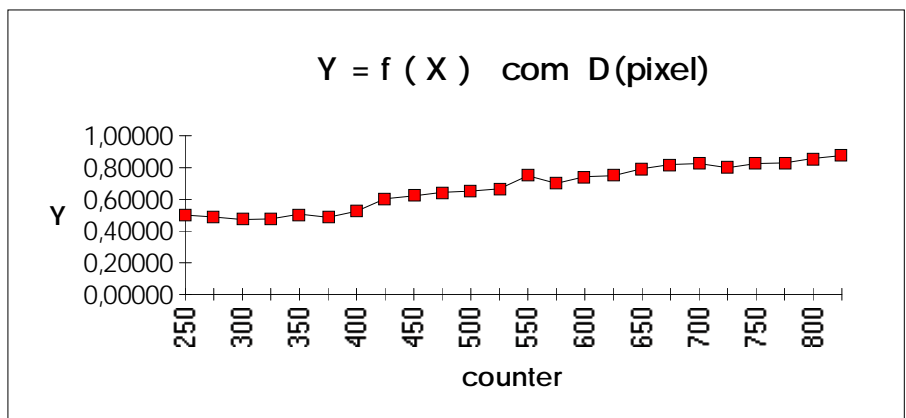
Objecto de calibração 1

Exp1.txt (10,0mm)

valores obtidos de 'calibrat.c'			----- valores calculados -----					--- gráficos ---	
counter	pixel	shift	p_to_m m	s x p_	mm	D(pixel)	tan(α)	sem D Y	com D Y
250	579,553	6,55	1,24797	8,17420	315,52	315,52	0,27200	0,68527	0,49932
275	569,353	6,35	1,24540	7,90827	302,80	302,80	0,26104	0,66912	0,48719
300	558,120	6,12	1,24246	7,60387	288,83	288,83	0,24899	0,65012	0,47272
325	540,033	6,03	1,23757	7,46254	266,40	266,40	0,22966	0,64110	0,47684
350	520,193	6,19	1,23205	7,62641	241,90	241,90	0,20854	0,65154	0,50599
375	499,842	5,84	1,22636	7,16192	216,89	216,89	0,18697	0,62151	0,48675
400	481,118	6,12	1,22118	7,47365	193,98	193,98	0,16722	0,64181	0,52569
425	458,855	6,86	1,21526	8,33671	166,85	166,85	0,14384	0,69494	0,60387
450	437,921	6,92	1,21006	8,37364	141,47	141,47	0,12196	0,69711	0,62099
475	416,038	7,04	1,20516	8,48432	115,04	115,04	0,09918	0,70358	0,64302
500	393,987	6,99	1,20092	8,39446	88,52	88,52	0,07631	0,69834	0,65186
525	370,987	6,99	1,19744	8,37009	60,94	60,94	0,05253	0,69690	0,66520
550	348,000	8,00	1,19508	9,56066	33,44	33,44	0,02883	0,76294	0,74766
575	324,077	7,08	1,19402	8,45369	4,87	4,87	0,00420	0,70180	0,69935
600	299,487	7,49	1,19459	8,94747	-24,50	-24,50	-0,02112	0,72991	0,74151
625	274,372	7,37	1,19678	8,82028	-54,52	-54,52	-0,04700	0,72280	0,74862
650	247,825	7,82	1,20061	9,38880	-86,34	-86,34	-0,07443	0,75389	0,79201
675	220,013	8,01	1,20600	9,66004	-119,81	-119,81	-0,10328	0,76811	0,81887
700	192,846	7,85	1,21230	9,51659	-152,65	-152,65	-0,13160	0,76063	0,82534
725	163,076	7,08	1,22005	8,63792	-188,86	-188,86	-0,16281	0,71245	0,79852
750	135,256	7,26	1,22773	8,91329	-222,90	-222,90	-0,19216	0,72800	0,82545
775	105,000	7,00	1,23618	8,65326	-260,18	-260,18	-0,22429	0,71332	0,82826
800	75,221	7,22	1,24421	8,98322	-297,11	-297,11	-0,25613	0,73189	0,85697
825	45,289	7,29	1,25160	9,12415	-334,47	-334,47	-0,28833	0,73963	0,87636



(a)



(b)

Fig. 30 - Representação de Y em função de X.

(a) sem informação de perspectiva.

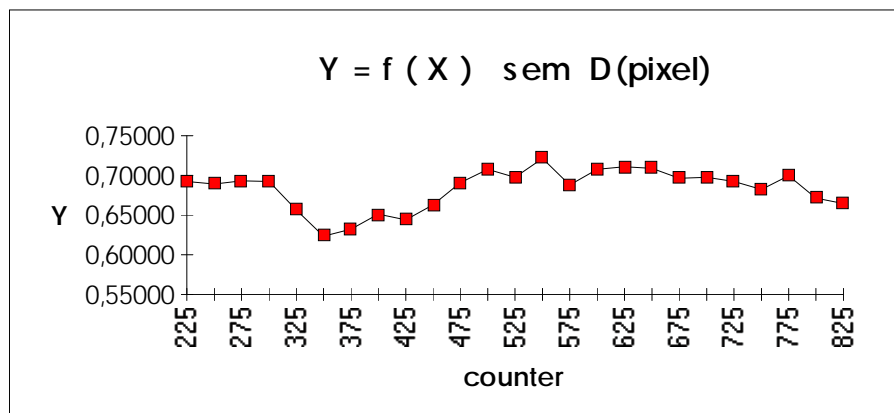
(b) com informação de perspectiva.

Objecto de calibração 2

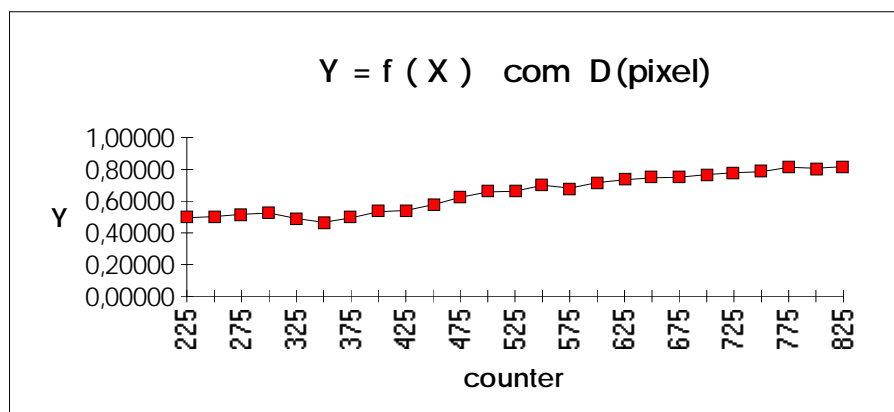
Exp2.txt (20,2mm)

valores obtidos de 'calibrat.c'			----- valores calculados -----					--- gráficos ---	
counter	pixel	shift	p_to_m m	s x p_	mm	D(pixel)	tan(α)	sem D Y	com D Y
225	596,357	13,36	1,25392	16,7524	336,53	336,53	0,29011	0,69237	0,49453
250	582,308	13,31	1,25061	16,6457	318,96	318,96	0,27497	0,68923	0,50213
275	570,447	13,45	1,24783	16,7833	304,17	304,17	0,26221	0,69327	0,51704
300	559,461	13,46	1,24512	16,7593	290,50	290,50	0,25043	0,69257	0,52502
325	546,539	12,54	1,24150	15,5685	274,46	274,46	0,23660	0,65663	0,49056
350	526,763	11,76	1,23601	14,5355	250,01	250,01	0,21552	0,62374	0,46689
375	507,026	12,03	1,23015	14,7987	225,71	225,71	0,19457	0,63228	0,49361
400	487,571	12,57	1,22473	15,3949	201,86	201,86	0,17402	0,65122	0,53163

425	465,448	12,45	1,21892	15,1755	174,87	174,87	0,15075	0,64431	0,54079
450	445,000	13,00	1,21342	15,7745	150,04	150,04	0,12935	0,66300	0,57748
475	423,821	13,82	1,20817	16,6970	124,43	124,43	0,10727	0,69074	0,62357
500	401,367	14,37	1,20339	17,2927	97,39	97,39	0,08395	0,70801	0,65751
525	378,115	14,12	1,19940	16,9355	69,48	69,48	0,05989	0,69772	0,66149
550	354,857	14,86	1,19625	17,7762	41,64	41,64	0,03590	0,72166	0,70107
575	330,897	13,90	1,19440	16,6022	13,01	13,01	0,01122	0,68795	0,68122
600	306,474	14,47	1,19408	17,2784	-16,15	-16,15	-0,01392	0,70760	0,71559
625	281,557	14,56	1,19553	17,4069	-45,93	-45,93	-0,03959	0,71126	0,73354
650	254,461	14,46	1,19868	17,3330	-78,38	-78,38	-0,06757	0,70916	0,74680
675	226,039	14,04	1,20346	16,8966	-112,54	-112,54	-0,09702	0,69658	0,75101
700	199,000	14,00	1,20938	16,9313	-145,20	-145,20	-0,12517	0,69759	0,76673
725	169,769	13,77	1,21661	16,7527	-180,70	-180,70	-0,15577	0,69238	0,77790
750	142,409	13,41	1,22428	16,4175	-214,13	-214,13	-0,18459	0,68247	0,78407
775	111,799	13,80	1,23275	17,0120	-251,78	-251,78	-0,21705	0,69993	0,81415
800	81,942	12,94	1,24080	16,0560	-288,75	-288,75	-0,24893	0,67159	0,80681
825	51,684	12,68	1,24879	15,8347	-326,47	-326,47	-0,28144	0,66484	0,81702



(a)



(b)

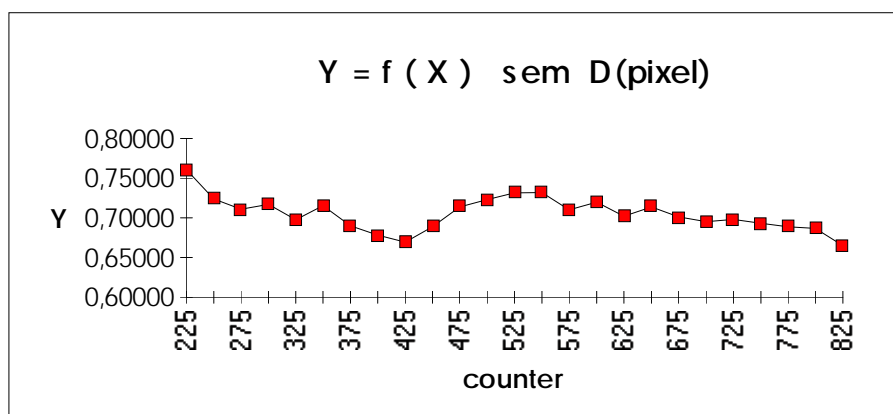
Fig. 31 - Representação de Y em função de X.

- (a) sem informação de perspectiva.
- (b) com informação de perspectiva.

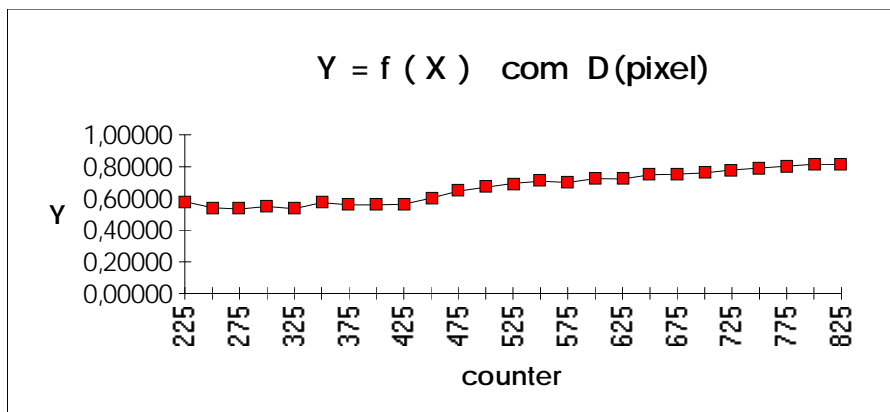
Objecto de calibração 3

Exp3.txt (30,4mm)

valores obtidos de 'calibrat.c'			----- valores calculados -----					--- gráficos ---	
counter	pixel	shift	p_to_m m	s x p_	mm	D(pixel)	tan(α)	sem D	com D
225	605,089	23,09	1,25392	28,9531	347,47	347,47	0,29954	0,76102	0,57839
250	590,487	21,49	1,25061	26,8757	329,18	329,18	0,28378	0,72394	0,54063
275	578,987	20,99	1,24783	26,1919	314,81	314,81	0,27139	0,71118	0,53317
300	568,287	21,29	1,24512	26,5087	301,48	301,48	0,25989	0,71713	0,54927
325	554,525	20,52	1,24150	25,4757	284,37	284,37	0,24514	0,69750	0,53516
350	534,377	21,38	1,23601	26,4258	259,41	259,41	0,22363	0,71558	0,57331
375	513,401	20,40	1,23015	25,0951	233,54	233,54	0,20133	0,69010	0,55800
400	494,012	20,01	1,22473	24,5069	209,74	209,74	0,18081	0,67848	0,55884
425	472,728	19,73	1,21892	24,0492	183,74	183,74	0,15840	0,66929	0,56412
450	451,634	20,63	1,21342	25,0329	158,09	158,09	0,13628	0,68888	0,60206
475	429,805	21,80	1,20817	26,3382	131,66	131,66	0,11350	0,71393	0,64535
500	407,293	22,29	1,20339	26,8235	104,51	104,51	0,09010	0,72298	0,67000
525	384,756	22,76	1,19940	27,2984	77,44	77,44	0,06676	0,73169	0,69349
550	360,829	22,83	1,19625	27,3103	48,78	48,78	0,04205	0,73191	0,70815
575	336,890	21,89	1,19440	26,1455	20,17	20,17	0,01739	0,71030	0,70022
600	312,329	22,33	1,19408	26,6639	-9,16	-9,16	-0,00790	0,72002	0,72446
625	286,525	21,52	1,19553	25,7278	-39,99	-39,99	-0,03447	0,70235	0,72209
650	259,975	21,98	1,19868	26,3471	-71,77	-71,77	-0,06187	0,71410	0,74837
675	232,333	21,33	1,20346	25,6698	-104,96	-104,96	-0,09049	0,70123	0,75176
700	204,975	20,98	1,20938	25,3728	-137,97	-137,97	-0,11894	0,69550	0,76163
725	175,962	20,96	1,21661	25,5002	-173,16	-173,16	-0,14927	0,69797	0,77941
750	147,637	20,64	1,22428	25,2691	-207,72	-207,72	-0,17907	0,69349	0,79052
775	117,312	20,31	1,23275	25,0372	-244,98	-244,98	-0,21119	0,68896	0,80249
800	88,089	20,09	1,24080	24,9277	-281,12	-281,12	-0,24235	0,68681	0,81562
825	57,090	19,09	1,24879	23,8395	-319,71	-319,71	-0,27561	0,66503	0,81443



(a)



(b)

Fig. 32 - Representação de Y em função de X.

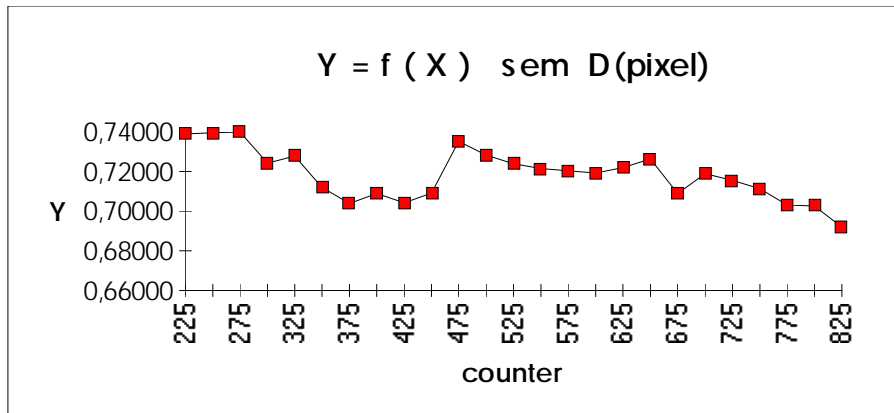
- (a) sem informação de perspectiva.
- (b) com informação de perspectiva.

Objecto de calibração 4

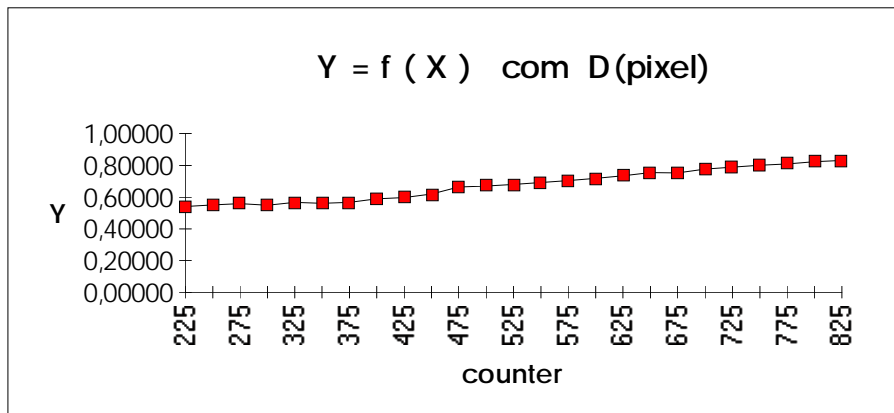
Exp4.txt (40,6mm)

valores obtidos de 'calibrat.c'			----- valores calculados -----					--- gráficos ---	
counter	pixel	shift	p_to_m m	s x p_	mm	D(pixel)	tan(α)	sem D Y	com D Y
225	615,450	29,45	1,25610	36,9920	360,47	360,47	0,31075	0,73893	0,54070
250	603,550	29,55	1,25359	37,0435	345,54	345,54	0,29788	0,73962	0,55103
275	593,623	29,62	1,25135	37,0649	333,10	333,10	0,28716	0,73991	0,55915
300	580,772	28,77	1,24827	35,9127	317,04	317,04	0,27331	0,72421	0,54864
325	563,080	29,08	1,24377	36,1688	295,00	295,00	0,25431	0,72774	0,56686
350	543,290	28,29	1,23846	35,0361	270,44	270,44	0,23313	0,71197	0,56206
375	522,963	27,96	1,23283	34,4699	245,32	245,32	0,21148	0,70392	0,56756
400	503,366	28,37	1,22734	34,8196	221,21	221,21	0,19070	0,70891	0,58818
425	482,244	28,24	1,22149	34,4949	195,35	195,35	0,16841	0,70428	0,59801
450	460,651	28,65	1,21573	34,8306	169,04	169,04	0,14572	0,70906	0,61885
475	439,341	30,34	1,21040	36,7236	143,19	143,19	0,12344	0,73531	0,66310
500	417,037	30,04	1,20537	36,2093	116,25	116,25	0,10021	0,72830	0,66962
525	393,890	29,89	1,20091	35,8951	88,40	88,40	0,07621	0,72397	0,67954
550	369,825	29,83	1,19729	35,7152	59,55	59,55	0,05133	0,72148	0,69179
575	346,819	29,82	1,19499	35,6347	32,03	32,03	0,02761	0,72036	0,70455
600	321,777	29,78	1,19400	35,5575	2,12	2,12	0,00183	0,71928	0,71825
625	296,927	29,93	1,19474	35,7586	-27,55	-27,55	-0,02375	0,72208	0,73530
650	270,122	30,12	1,19730	36,0626	-59,61	-59,61	-0,05139	0,72628	0,75428
675	241,976	28,98	1,20164	34,8235	-93,37	-93,37	-0,08049	0,70896	0,75353
700	215,440	29,44	1,20699	35,5339	-125,32	-125,32	-0,10804	0,71895	0,77696
725	186,073	29,07	1,21400	35,2910	-160,87	-160,87	-0,13868	0,71556	0,78934
750	157,646	28,65	1,22152	34,9966	-195,48	-195,48	-0,16852	0,71141	0,80042
775	126,981	27,98	1,23004	34,4166	-233,07	-233,07	-0,20092	0,70316	0,80913

800	96,762	27,76	1,23845	34,3793	-270,37	-270,37	-0,23308	0,70262	0,82378
825	65,987	26,99	1,24658	33,6453	-308,61	-308,61	-0,26604	0,69200	0,83060



(a)



(b)

Fig. 33 - Representação de Y em função de X.

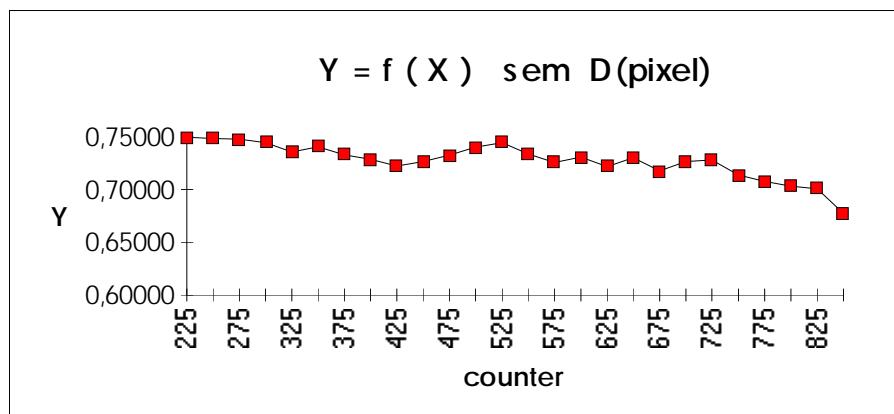
- (a) sem informação de perspectiva.
- (b) com informação de perspectiva.

Objecto de calibração 5

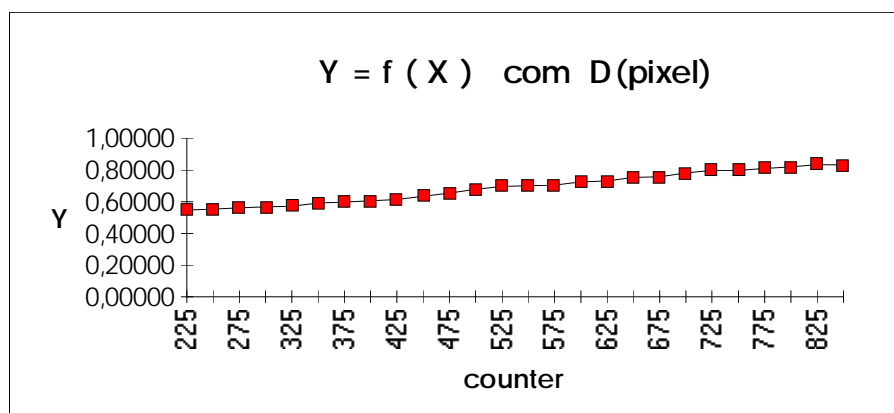
Exp5.txt (50,8mm)

valores obtidos de 'calibrat.c'			----- valores calculados -----					--- gráficos ---	
counter	pixel	shift	p_to_m m	s x p_ m	mm	D(pixel)	tan(α)	sem D Y	com D Y
225	624,573	37,57	1,25787	47,2584	371,94	371,94	0,32064	0,74930	0,54748
250	614,537	37,54	1,25591	47,1469	359,32	359,32	0,30976	0,74812	0,55379
275	604,549	37,55	1,25381	47,0804	346,79	346,79	0,29896	0,74741	0,56063
300	590,402	37,40	1,25059	46,7722	329,07	329,07	0,28369	0,74414	0,56720
325	570,890	36,89	1,24579	45,9572	304,72	304,72	0,26269	0,73539	0,57072
350	551,390	37,39	1,24066	46,3884	280,48	280,48	0,24179	0,74004	0,59125
375	531,000	37,00	1,23507	45,6976	255,24	255,24	0,22003	0,73257	0,59685

400	511,837	36,84	1,22971	45,3026	231,62	231,62	0,19967	0,72826	0,60541
425	490,548	36,55	1,22377	44,7289	205,50	205,50	0,17716	0,72193	0,61296
450	469,036	37,04	1,21793	45,1122	179,24	179,24	0,15452	0,72617	0,63287
475	447,667	37,67	1,21243	45,6723	153,27	153,27	0,13213	0,73230	0,65425
500	424,416	38,42	1,20696	46,3715	125,15	125,15	0,10789	0,73986	0,67774
525	401,905	38,90	1,20235	46,7716	98,03	98,03	0,08451	0,74413	0,69642
550	378,214	38,21	1,19842	45,7917	69,60	69,60	0,06000	0,73359	0,69949
575	354,667	37,67	1,19564	45,0397	41,41	41,41	0,03570	0,72537	0,70502
600	330,114	38,11	1,19415	45,5089	12,08	12,08	0,01041	0,73051	0,72471
625	304,417	37,42	1,19434	44,6923	-18,61	-18,61	-0,01604	0,72153	0,73050
650	277,994	37,99	1,19637	45,4502	-50,19	-50,19	-0,04327	0,72987	0,75339
675	249,916	36,92	1,20026	44,3137	-83,83	-83,83	-0,07227	0,71731	0,75691
700	222,427	37,43	1,20548	45,1212	-116,89	-116,89	-0,10077	0,72626	0,77986
725	193,337	37,34	1,21218	45,2629	-152,06	-152,06	-0,13108	0,72782	0,79632
750	166,073	36,07	1,21924	43,9779	-185,20	-185,20	-0,15966	0,71354	0,79792
775	134,390	35,39	1,22797	43,4578	-223,97	-223,97	-0,19308	0,70766	0,80909
800	104,815	34,81	1,23623	43,0332	-260,41	-260,41	-0,22449	0,70281	0,81995
825	73,412	34,41	1,24468	42,8296	-299,36	-299,36	-0,25807	0,70048	0,83351
850	41,612	32,61	1,25244	40,8420	-339,07	-339,07	-0,29230	0,67716	0,83129



(a)



(b)

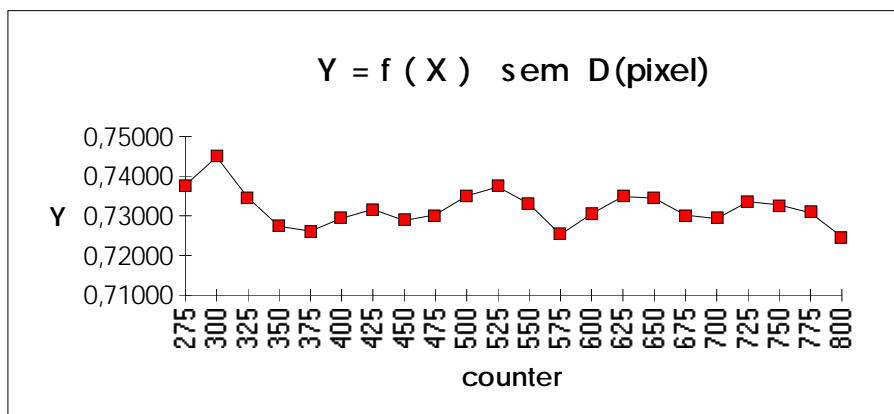
Fig. 34 - Representação de Y em função de X.

- (a) sem informação de perspectiva.
- (b) com informação de perspectiva.

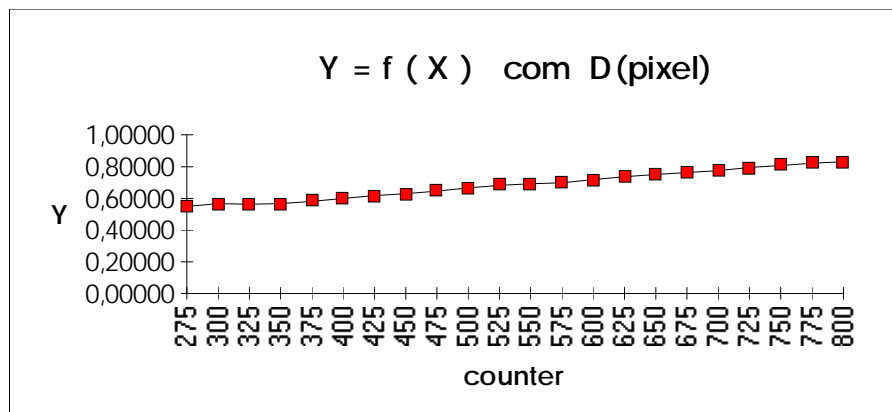
Objecto de calibração 6

Exp6.txt (61,0mm)

valores obtidos de 'calibrat.c'			----- valores calculados -----					--- gráficos ---	
counter	pixel	shift	p_to_m m	s x p_	mm	D(pixel)	tan(α)	sem D	com D
275	603,2	44,23	1,25352	55,4431	345,14	345,14	0,29754	0,73771	0,54874
300	592,0	44,96	1,25096	56,2431	331,02	331,02	0,28536	0,74485	0,56694
325	579,2	44,15	1,24787	55,0935	315,02	315,02	0,27157	0,73456	0,56333
350	560,7	43,68	1,24314	54,3004	292,02	292,02	0,25174	0,72736	0,56820
375	540,8	43,76	1,23777	54,1648	267,31	267,31	0,23044	0,72612	0,58164
400	521,2	44,24	1,23235	54,5190	243,19	243,19	0,20965	0,72935	0,59998
425	500,7	44,65	1,22658	54,7669	217,88	217,88	0,18783	0,73161	0,61740
450	478,6	44,62	1,22050	54,4589	190,92	190,92	0,16459	0,72881	0,62939
475	457,0	44,96	1,21478	54,6164	164,56	164,56	0,14186	0,73024	0,64573
500	434,6	45,60	1,20928	55,1432	137,45	137,45	0,11849	0,73501	0,66583
525	412,0	46,00	1,20433	55,3990	110,18	110,18	0,09498	0,73732	0,68274
550	388,8	45,77	1,20005	54,9261	82,26	82,26	0,07091	0,73305	0,69248
575	366,2	45,18	1,19685	54,0735	55,19	55,19	0,04757	0,72528	0,69800
600	341,7	45,74	1,19466	54,6437	25,96	25,96	0,02238	0,73049	0,71794
625	316,2	46,17	1,19402	55,1280	-4,58	-4,58	-0,00395	0,73488	0,73704
650	289,1	46,08	1,19531	55,0800	-36,93	-36,93	-0,03184	0,73444	0,75171
675	261,6	45,57	1,19845	54,6134	-69,85	-69,85	-0,06022	0,73021	0,76266
700	235,3	45,32	1,20288	54,5145	-101,37	-101,37	-0,08739	0,72931	0,77584
725	206,5	45,48	1,20903	54,9867	-136,15	-136,15	-0,11737	0,73360	0,79471
750	179,2	45,15	1,21578	54,8924	-169,27	-169,27	-0,14593	0,73275	0,80778
775	148,7	44,67	1,22399	54,6757	-206,46	-206,46	-0,17799	0,73078	0,82121
800	117,8	43,82	1,23261	54,0130	-244,35	-244,35	-0,21065	0,72472	0,83122



(a)



(b)

Fig. 35 - Representação de Y em função de X.

- (a) sem informação de perspectiva.
- (b) com informação de perspectiva.

9.6.2 Tabela Final de Resultados

A Tabela Final de Resultados para $H=(1120+30)$, $H=(1120+40)$ e $H=(1120+50)$ mm, considerando a informação de perspectiva $D()$, é a seguinte:

declive		
+30	+40	+50
0,000778	0,000773	0,000768
0,000648	0,000643	0,000637
0,000536	0,000531	0,000526
0,000542	0,000536	0,000531
0,000510	0,000505	0,000500
0,000571	0,000566	0,000561

ord. na origem		
+30	+40	+50
0,256400	0,259537	0,262651
0,306400	0,309737	0,312919
0,386993	0,390164	0,393277
0,391294	0,394527	0,397702
0,413892	0,417156	0,420360
0,376976	0,380336	0,383633

<declive>
0,000592

<ord. na origem>
0,358576

9.6.3 Obtenção dos Parâmetros Geométricos e Mecânicos

O valor de escolhido para H foi de $(1120+40)$ mm.

Naquelas tabelas fez-se variar o valor de H para 3 valores possíveis: $(1120+30)$, $(1120+40)$ e $(1120+50)$ mm. 1120 mm é a distância medida entre o vértice da lente da câmara e o plano $Z=0$. Mas H deve corresponder à distância medida a partir do orifício do modelo "pin-hole" para a câmara, cujo valor exacto é difícil de definir. Se ele se

encontrar aproximadamente a meio do sistema óptico então o valor a somar a 1120mm poderá estar compreendido entre 30 e 50mm. Sendo assim foram experimentados aqueles 3 valores, dos quais se escolherá o que der melhores resultados.

Ora, por análise da tabela de resultados pode-se verificar que na realidade o valor de H tem pouca influência nos resultados. O mesmo não se poderá dizer dos objectos de prova para os quais os resultados demonstram haver alguma variação. Esta poderá, pelo menos em parte, ser explicada pelo carácter descontínuo do pixel e de haver algum erro na avaliação correcta de *shift* para objectos de pequena altura, para os quais aquele valor é também pequeno. Para se poder ter uma ideia do erro introduzido na medida de *shift* dever-se-á analisar as tabelas acima apresentadas. No caso do objecto com 10,0mm de altura, por exemplo, o valor de *shift* ronda os 7 pixeis; para uma tolerância introduzida pela leitura de 0,5 pixeis teremos um erro de cerca de 7% logo à partida. Por esta ordem de ideias seriam então mais correctos os parâmetros obtidos para os objectos mais altos. Contudo, a flutuação obtida para eles não permite adoptar esses valores por falta de um grau elevado de certeza. Poder-se-ia também adoptar os parâmetros associados a uma altura Z que ocorresse mais frequentemente. Mas, mais uma vez, a flutuação dos valores é demasiado elevada (~13%) para se poder adoptá-los com segurança pelo que optou-se pelo valor médio de todos os parâmetros obtidos para todos os objectos de prova. São eles,

$$\left\{ \begin{array}{l} \text{declive} = \frac{\text{angle_per_step}}{\text{gear}} \approx 0,000592 \pm 13\% \\ \text{ord. na origem} = \arctg\left(\frac{d}{h}\right) = \theta_0 \approx 0,358576\text{rad} \pm 14\% \approx 20,5^\circ \pm 3^\circ \end{array} \right.$$

Para a obtenção destes valores foram considerados apenas os resultados para a nova variável Y , com a informação de perspectiva, ou seja, com a inclusão de D/H . Aliás, as representações gráficas efectuadas sem e com D/H para todos os 6 objectos de calibração confirmam precisamente a validade da introdução daquela informação: só para os segundos gráficos se obtém uma recta, como pretendido, com declive positivo (*angle_per_step/gear* é positivo) e ordenada na origem também positiva (como seria de esperar atendendo a que é igual a $\arctg(d/h)$).

Aqueles valores de declive e ordenada na origem, são então introduzidos na fórmula de Z como parâmetros mecânicos e geométricos fixos.

9.6.4 Avaliação Aproximada dos Erros em Z

As tolerâncias que acompanham aqueles parâmetros, vão obviamente repercutir-se no erro de medida de Z , mas qual será a sua influência em termos numéricos? A resposta pode ser-nos dada pela Teoria de Propagação de Erros. Agora, os parâmetros fixos serão as variáveis com os erros associados atrás mostrados:

$$Z = \frac{a}{\text{tg}(x + b \times y) + c}$$

$$\begin{cases} x = \text{arctg}\left(\frac{d}{h}\right) = \theta_0 \rightarrow \Delta x = 14\% \\ y = \frac{\text{angle_per_step}}{\text{gear}} \rightarrow \Delta y = 13\% \end{cases}$$

sendo constantes os termos,

$$\begin{cases} a = \text{shift} \times \text{pixel_to_mm} \\ b = \text{counter} \\ c = \frac{D}{H} \end{cases}$$

Então,

$$\Delta Z \leq \left| \frac{\partial Z}{\partial x} \cdot \Delta x \right| + \left| \frac{\partial Z}{\partial y} \cdot \Delta y \right| = \frac{a / \cos^2(x + b \cdot y)}{[\text{tg}(x + b \cdot y) + c]^2} \cdot \Delta x + \frac{a / \cos^2(x + b \cdot y)}{[\text{tg}(x + b \cdot y) + c]^2} \cdot b \cdot \Delta y$$

Ora, c pode ser desprezado relativamente a $\text{tg}(x+b \cdot y)$:

por exemplo na exp.6 e na situação mais desfavorável (*counter* pequeno), $D \approx 345\text{mm}$ e $\text{counter} = 275$ pelo que,

$$\text{tg}(x+b \cdot y) \approx \text{tg}(0,36 + 0,0006 \cdot 275) = \text{tg}(0,525) \approx 0,6$$

e, $c = D/H = 345/1160 \approx 0,3 < 0,6$

Numa outra situação (intermédia), $D = 138\text{mm}$ e $\text{counter} = 500$:

$$\text{tg}(x+b \cdot y) \approx \text{tg}(0,36 + 0,0006 \cdot 500) = \text{tg}(0,66) \approx 0,78$$

e, $c = D/H = 138/1160 \approx 0,12 \ll 0,78$

Na situação mais favorável da exp.6, $D = -4,6\text{mm}$ e $\text{counter} = 625$:

$$\text{tg}(x+b \cdot y) \approx \text{tg}(0,36 + 0,0006 \cdot 625) = \text{tg}(0,735) \approx 0,9$$

e, $c = -4,6/1160 \approx -0,0004 \ll 0,9$

Pode-se então simplificar a expressão de ΔZ :

$$\frac{a / \cos^2(x + b \cdot y)}{[\operatorname{tg}(x + b \cdot y) + c]^2} \approx \frac{a / \cos^2(x + b \cdot y)}{\operatorname{tg}^2(x + b \cdot y)} = \frac{a}{\operatorname{sen}^2(x + b \cdot y)}$$

Pelo que,

$$\Delta Z \leq \frac{a}{\operatorname{sen}^2(x + b \cdot y)} \cdot \Delta x + \frac{a \cdot b}{\operatorname{sen}^2(x + b \cdot y)} \cdot \Delta y$$

a e b são na realidade variáveis; mas para uma estimativa do erro de Z , basta substituir por um par de valores retirados das experiências (de preferência correspondentes a uma situação desfavorável para majorar aquele erro). Por exemplo, ainda na exp.6 para $(counter; D) = (275; 345)$ teremos $shift \approx 44$ e $pixel_to_mm \approx 1,25$, pelo que,

$$\Delta Z \leq \frac{55}{\operatorname{sen}^2(0,36 + 0,0006 \cdot 275)} \cdot 0,36 \cdot 0,14 + \frac{55 \cdot 275}{\operatorname{sen}^2(0,36 + 0,0006 \cdot 275)} \cdot 0,0006 \cdot 0,13 \approx$$

$$\approx 11 + 5 = 16\text{mm}$$

O que para $Z = 61,0\text{mm}$ dá um erro relativo de **26%**.

Numa outra situação da exp.6, $\left\{ \begin{array}{l} counter = 600 \\ D = 26\text{mm} \\ shift = 46 \\ pixel_to_mm \approx 1,2 \end{array} \right.$

$$\text{logo, } \Delta Z \leq \frac{55 \cdot 2 \cdot 0,36 \cdot 0,14}{\operatorname{sen}^2(0,36 + 0,0006 \cdot 600)} + \frac{55 \cdot 2 \cdot 600 \cdot 0,0006 \cdot 0,13}{\operatorname{sen}^2(0,36 + 0,0006 \cdot 600)} \approx 6,4 + 6,4 = 12,8\text{mm}$$

o que mais uma vez para $Z = 61,0\text{mm}$ dá um erro relativo de **21%**.

Estas estimativas para erro de Z foram calculadas por excesso. Na realidade os erros experimentais na medida de Z vieram a mostrar-se bem menores que aqueles valores e por conseguinte a confirmar este sistema como bom medidor de alturas.

9.6.5 O Programa de Calibração

Para terminar, resta fazer referência ao programa desenvolvido para a medição e registo de $shift$ em função de $counter$. Neste programa, são enviados conjuntos de 25

incrementos ao controlador do motor passo-a-passo até que a linha laser saia fora do campo de visão da câmara (ou o utilizador opte por interromper a execução), após o que a sua execução pára; para cada um destes conjuntos a variável *counter*, que é mantida sempre em memória e constantemente actualizada, é registada e o desvio *shift* medido e também registado. Este par de valores é acompanhado pelo valor $x \equiv pixel$ (abcissa da imagem em unidades de pixel) da linha laser não desviada e usada como referência para o cálculo das variáveis *pixel_to_mm*, $D(x,y)$ e $mm(x,y)$.

Mostra-se de seguida um exemplo de registo daquele programa, para avaliação dos parâmetros fixos.

```
Computing shifts... (type 'n' to end)
Z = 61,0 mm
```

medida nº	<i>pixel</i>	<i>counter</i>	<i>shift</i>
0	603.235	275	44.23
1	591.958	300	44.96
2	579.155	325	44.15
3	560.683	350	43.68
4	540.762	375	43.76
...
17	235.321	700	45.32
18	206.476	725	45.48
19	179.155	750	45.15
20	148.667	775	44.67
21	117.821	800	43.82

```
internal counter = 825
No shifts have been detected... execution has stopped.
another height ? type a control char ... ('n' to exit)

Execution stopped by user...
```

Como se pode verificar, estes resultados correspondem aos dados utilizados na tabela para o caso de $Z=61,0\text{mm}$.

9.7 Apresentação de Amostragens de Altura com Várias Resoluções

Uma vez de posse de todos os parâmetros geométricos assim como da função de escolha do melhor factor de conversão de *pixel* para *mm*, é possível obter boas amostragens de alturas que são automaticamente mostradas em monitor com codificação de côr por altura e registadas em disco quer em modo texto quer em modo binário.

Nesta secção apresenta-se o resultado obtido em monitor do varrimento de uma mesma cena e consequente amostragem de alturas, para várias resoluções. As medidas de altura efectuadas serão comparadas com os valores reais e os tempos de execução serão também comparados.

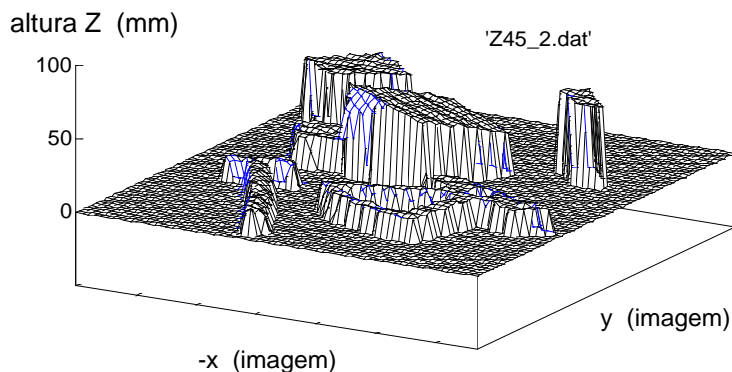


Fig. 36 - Perspectiva tridimensional dos objectos presentes na cena (foi utilizado $inc_x=2$ e $inc_y=2$ para registo dos valores de Z num ficheiro e inibição do registo dos valores (x,y)).

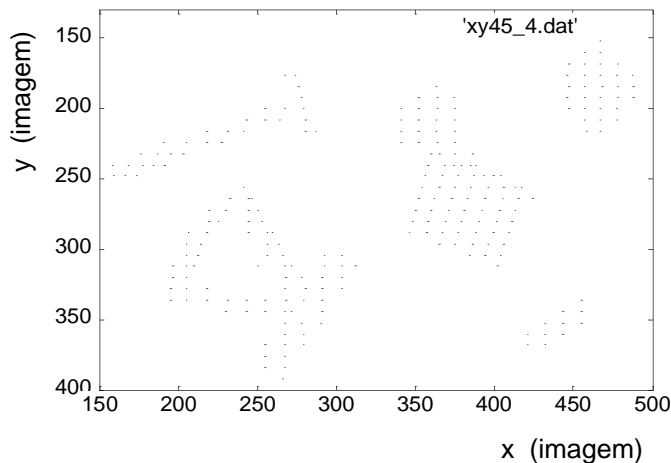


Fig. 37 - Representação do vector de estruturas a 2D (varrimento laser e amostragem das alturas com $res=4$). Foi inibida a extracção dos valores de Z e registado num ficheiro os valores de (x,y) obtidos com $inc_x=inc_y=4$ para representação bidimensional.

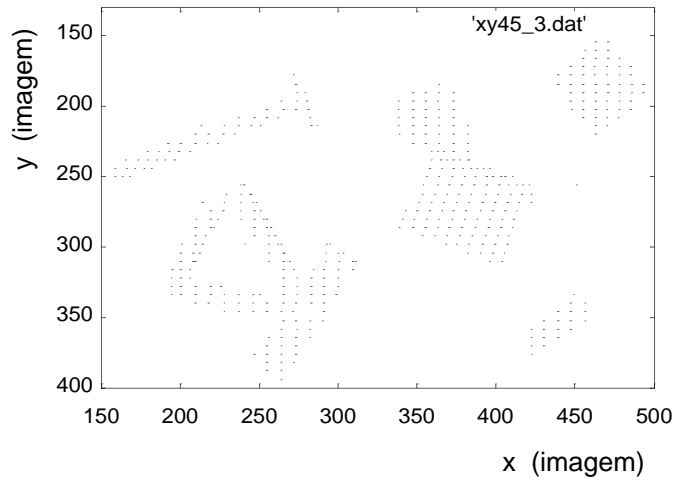


Fig. 38 - Amostragem das alturas com $res=3$.

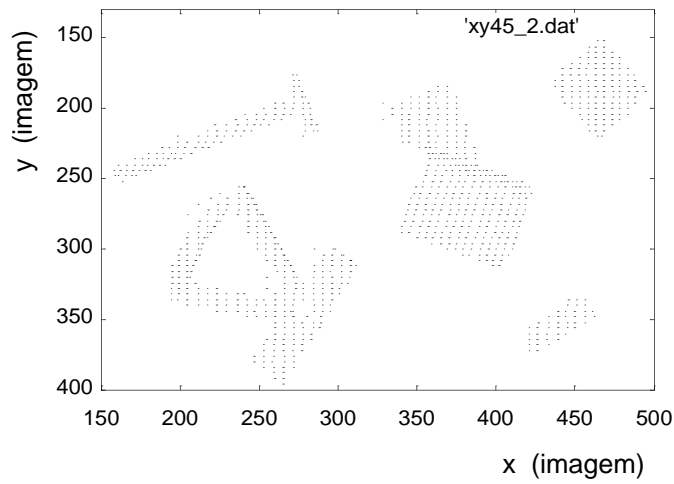


Fig. 39 - Amostragem das alturas com $res=2$.

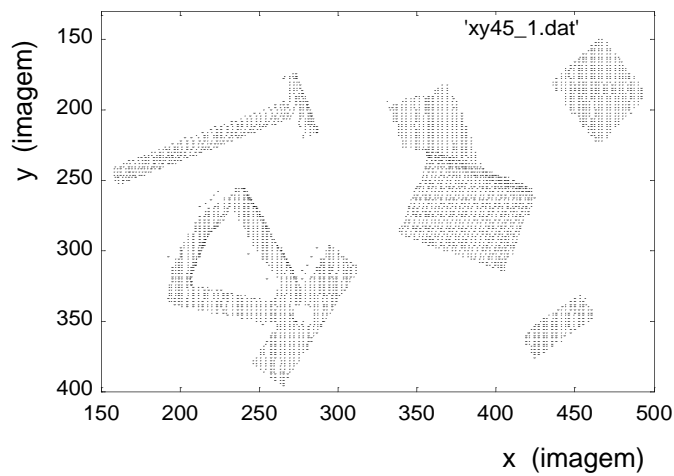


Fig. 40 - Amostragem das alturas com $res=1$.

9.8 A Recepção de Pedidos para Varrimento da Cena e os Tempos de Execução

Os pedidos efectuados por outros agentes são previamente registados num ficheiro numa área de acesso comum em disco. Este ficheiro é procurado pelo agente Laser naquela mesma área. O ficheiro de pedidos deve conter mensagens compreendidas pelo agente Laser, de um conjunto que contempla várias possibilidades, cada uma das quais se refere a um modo distinto de funcionamento (já referenciados na apresentação de agente).

O ficheiro de pedidos para o conjunto de objectos apresentado na secção anterior, por exemplo, é o seguinte:

'requests.Las'

1
<i>n</i>
132
500
136
408

(tendo em conta o rectângulo que o delimita)

"1" é a acção escolhida para o Agente que corresponde a um varrimento local, *n* a traduz a resolução pretendida ($n = 1, 2, 3, 4$), e os últimos 4 parâmetros determinam a coordenadas do rectângulo que define a área a analisar:

$$\begin{cases} x_{esq} = 132 \\ x_{dir} = 500 \end{cases} \quad \begin{cases} y_{esq} = 136 \\ y_{dir} = 408 \end{cases}$$

A execução mais vulgar deste agente é de facto o varrimento laser da cena, dado um conjunto de coordenadas delimitadoras e uma resolução pretendida. A tabela abaixo mostra os vários tempos de execução obtidos experimentalmente:

Resolução	Tempo (s)
4	31
3	41
2	60
1	124

Estes tempos foram obtidos com um PC 486DX a 33MHz. Na altura da escrita desta dissertação havia já disponível no mercado a preço muito reduzido, o Pentium a 133 Mhz, cuja velocidade de execução estimo que aumente em pelo menos 4x, e outros ainda mais rápidos.

Sempre que o varrimento é efectuado na sua resolução máxima ($res=1$), é automaticamente chamado o programa de filtragem do ficheiro de amostragens, de cuja execução resulta a imagem de alturas normalmente registado em disco. O seu tempo de execução é de cerca de 5s que contempla a visualização no monitor dos resultados parciais de cada fase de filtragem. Este valor, que já é pequeno comparativamente com os tempos de varrimento em qualquer das resoluções, pode ainda ser substancialmente reduzido se se optar pela visualização apenas do resultado final.

9.9 A Utilização do Vector de Estruturas

Como foi já visto, o conjunto de algoritmos de cálculo, decisão e intercomunicação a que se chamou Agente Laser funciona de um modo totalmente autónomo, necessitando apenas de um pedido de execução acompanhado dos respectivos parâmetros associados. Das várias acções possíveis destaca-se a construção de imagens de distância quer para uso próprio quer para uso de outros Agentes e a identificação a 3D de objectos na cena. Os resultados obtidos nesta opção concorrem directamente com os obtidos pelo algoritmo de identificação a 2D do Agente Visão.

Este agente pode contudo prestar serviços úteis aos demais agentes em vez de efectuar uma análise completa da cena. Mais concretamente, ele pode executar acções que se destinem a fornecer informações complementares aos algoritmos 2D do Agente Visão de forma a corrigir e levantar ambiguidades derivadas do desconhecimento da terceira dimensão. Deste modo podem-se tornar os resultados deste agente mais correctos. Por exemplo, os dois algoritmos de identificação do Agente Visão podem necessitar de alguns valores de altura em pontos bem determinados para corrigir ou complementar a informação nas suas estruturas de dados. Isto requer protocolos de comunicação e áreas comuns de armazenamento provisório de ficheiro de pedidos e de dados e também de resultados. Mas para além de tudo isso, requer que o Agente Laser possua ou possa construir informação estruturada útil a essas acções.

No caso deste agente possuir uma imagem de distâncias já construída, o fornecimento dessa informação fica extremamente facilitada como se verá adiante. No caso contrário, este agente pode usar (ou construir rapidamente) a estrutura de dados descrita em secções anteriores: o vector de estruturas armazenadas em memória, ou em disco.

É desta particularidade do Agente Laser e dos algoritmos construídos para esse fim que se referirá já a seguir, para cuja acção é essencial (caso não haja uma imagem de distâncias) esse vector de estruturas com informação de alturas e descrito na secção anterior.

9.9.1 O Algoritmo de Atribuição de Alturas a Pontos Aleatórios

Esta acção, que corresponde à opção 3 do agente, consiste no preenchimento do campo da altura das estruturas construídas pelo Agente Visão para descrição dos objectos presentes na cena (Fig.41). Para tal necessita que este ficheiro lhe seja disponibilizado (ou seja, enviado para a área comum de acesso) e recebe o pedido adequado através da transferência do ficheiro que contém os pedidos e que deverá conter duas linhas, uma referente ao modo de operação que neste caso é o 3 e representado pelo caractere "r" e outra contendo um algarismo em texto que se refere à resolução do varrimento laser a efectuar caso não estejam disponíveis nem a imagem de distâncias nem o ficheiro com a amostragem dos valores de altura.

O nome do algoritmo foi escolhido devido ao carácter aleatório dos pontos aos quais o Agente Laser deve atribuir valores de altura, pelo menos do ponto de vista deste agente que *a priori* desconhece a localização dos objectos analisados pelo Agente Visão.

No Anexo G.1 é apresentada uma descrição mais pormenorizada da estrutura de dados utilizada, do conteúdo do ficheiro de pedidos e dos dois ficheiros de respostas e ainda uma descrição pormenorizada das acções empreendidas pelo algoritmo.

9.9.2 O Algoritmo de Atribuição de Alturas a Pontos Pertencentes a Segmentos

Este algoritmo é em tudo semelhante ao anterior excepto que lida com um ficheiro de dados com outra estrutura e proveniência. O nome deriva do facto de apenas alguns poucos valores de Z serem pedidos e destina-se a auxiliar o Agente Visão a distinguir cadeias de segmentos de fronteira pertencentes a objectos diferentes sobrepostos (Fig.41). Trata-se de uma tentativa por parte daquele agente de poder detectar sobreposições e eventualmente identificar o objecto sobreposto pela análise de apenas uma parte da sequência de segmentos do bordo desse objecto.

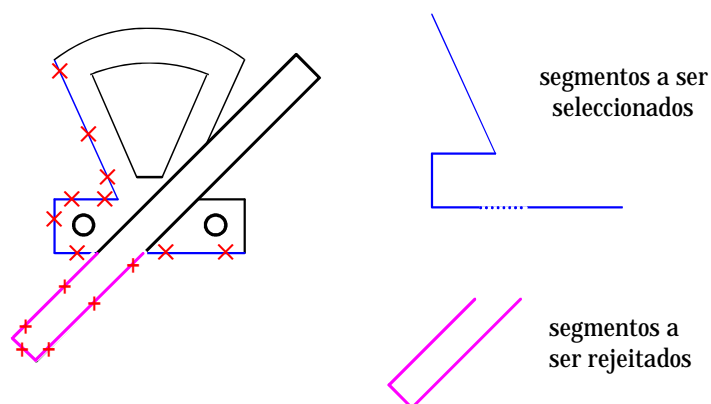


Fig. 41 - Situação real que requer a intervenção do algoritmo de atribuição de alturas: a análise a 2D das sequências de segmentos pertencentes a dois objectos distintos pelo Agente Visão.

Uma vez na posse da altura de alguns pontos chave, o programa é capaz de separar os dois conjuntos de segmentos cada um pertencente apenas a um dos objectos. A eventual identificação da sequência do objecto sobreposto poderá posteriormente dispensar a chamada ao Agente Laser com o pedido de identificar a 3D os objectos presentes na cena o que reduzirá imenso o tempo de execução dos programas de análise da cena que antecedem a acção do planeador de acções de alto nível.

A estrutura de dados utilizada é idêntica à da amostragem de alturas. A acção deste algoritmo é também idêntica à do algoritmo anterior excepto que transfere um ficheiro diferente e com estrutura interna diferente, de e para a área comum de acesso.

No Anexo G.2 é apresentado o conteúdo dos ficheiros de comandos de transferência FTP, que quando executados efectuem as transferências do ficheiro de dados e apresentada a estrutura interna deste.

9.9.3 A Função Comum *get_Z()*

Em ambos os algoritmos é feita uma procura inicial da imagem de distâncias uma vez que é o registo mais completo: ele possui uma altura atribuída a todos os pixels da imagem inicial; na realidade os valores "grey-level" que traduzem a intensidade luminosa são substituídos por outros que após descodificação fornecem a altura em *mm* em cada ponto como já foi visto. A resolução espacial desta imagem é portanto igual à da imagem de intensidades (Fig.42).

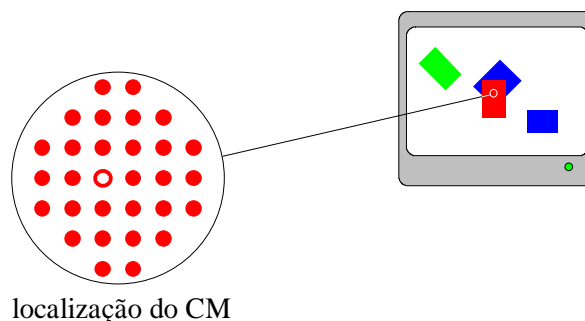


Fig. 42 - Numa imagem de alturas todos os pixels possuem uma altura associada que pode ser obtida por descodificação do valor de nível de cinzento nele registado.

Na falta dela, ambos os algoritmos recorrem ao ficheiro de amostragem de alturas (cuja exibição no monitor é ilustrada na Fig.43) através de uma função comum '*get_Z()*'. Para ser acessível a ambos esta função deve ignorar a estrutura de dados utilizada em qualquer deles o que implica que nenhum deles passe o apontador da sua estrutura de dados mas antes a coordenada (*x,y*) do ponto da imagem em unidades de pixel (para além do apontador para o ficheiro de amostragens, em memória). Ela retorna obviamente o valor *Z* associado. Isto implica uma chamada para cada valor *Z* pretendido o que retarda a execução, mas pelo menos torna-a compatível com ambas as estruturas de dados utilizadas, tornando-a genérica e dispensando assim a duplicação de código.

De qualquer modo o tempo de execução é imperceptível pelo que não causa qualquer atraso na acção do agente.

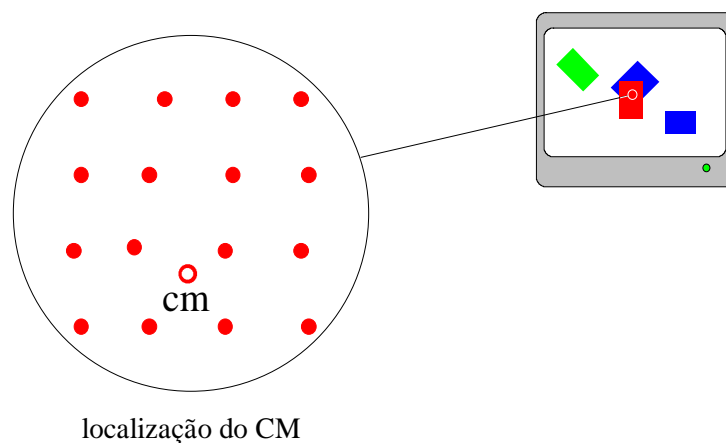


Fig. 43 - Exibição em monitor do ficheiro de amostragem de alturas; a sua resolução é inferior à da imagem de distâncias, ou seja, apenas alguns pixels possuem valores codificados de altura.

A actuação desta função tem em conta o modo como a amostragem de alturas foi feita:

1. A aquisição de alturas não é feita para todos os pontos da imagem, caso contrário não seria uma amostragem mas sim uma imagem de alturas completa, mas para vários pontos de cada linha laser projectada na cena.
2. Como consequência de (1) o valor (x,y) muito provavelmente não coincide com nenhum dos pontos que possuem Z s associados pelo que o valor de Z a retornar deverá ser uma qualquer média de valores de Z da vizinhança desse ponto (x,y) .
3. A amostragem é efectuada para coordenadas de y fixas que dependem apenas dos valores impostos mínimo, y_{min} e máximo, y_{max} , e da resolução contida no ficheiro de pedidos. Vários ficheiros de amostragens foram construídos para os valores limite (160,470) mas os valores mais actuais são (129,447) que têm em conta os limites locais onde os objectos se poderão encontrar. É de relembrar que a linha 128 é utilizada na imagem de distâncias para registar o valor máximo de Z em mm encontrado aquando do varrimento laser da cena para construção do ficheiro de amostragens e necessário para o factor de conversão de valores "grey-level" nela registados para alturas. Por esse facto não se devem registar alturas codificadas nessa linha (as imagens de altura são obtidas por filtragem de um ficheiro de amostragem com a resolução máxima) e daí o valor mínimo $y_{min} = 129$. Ficam então excluídos os 2 primeiros blocos de 64 linhas da imagem original que está de acordo com o raio de acção do robot.

O valor máximo contempla a impossibilidade de os objectos se encontrarem demasiado perto do robot para poderem ser manipulados (os eixos y dos dois sistemas de coordenadas são simétricos). O valor adoptado (447) exclui o último bloco de 64 linhas da imagem de intensidade original. Sendo assim apenas os blocos de 2 a 6 são utilizados (ficando excluídos os dois primeiros e o último) o que reduz imenso a quantidade de informação a utilizar. Por exemplo, no caso da imagem de alturas, o ficheiro reduz de 327.689 bytes para 204.800 (redução em 37,5%).

Aqueles limites aliados à periodicidade na escolha do y dado pela resolução pedida (para a resolução máxima, $res = 1$, a amostragem é feita de 2 em 2 pixels; ou seja a periodicidade é de $2 \times resolu\c{c}o$) faz com que os valores de y para os quais são efectuadas medidas sejam sempre os mesmos e conhecidos, ao contrário do que acontece com as coordenadas x .

Tendo em conta que o ficheiro de amostragens de altura regista as estruturas de dados para y_s crescentes (desde y_{min} até y_{max} num máximo de p_per_line valores que dependem da resolução escolhida res) e x_s decrescentes (começa pelos valores mais altos de x) com limites também pré-definidos mas estes escolhidos pelo agente que envia o pedido, o processo de procura é então o seguinte:

o valor do índice de procura é incrementado até que ultrapasse o valor y do ponto (x,y) como o sugere a Fig.44. Seguidamente "salta-se" de coluna em coluna (incrementos de p_per_line estruturas) até o índice de procura i descer abaixo da coordenada x do ponto (x,y) . Obtém-se assim um ponto (i,j) correspondente ao pixel inferior esquerdo (do ponto de vista da imagem) do "quadrado" delimitador de (x,y) . Se P for o apontador para o vector daquelas estruturas armazenadas em memória, então o apontador para o ponto (i,j) será:

$$p = P + p_per_line \times i$$

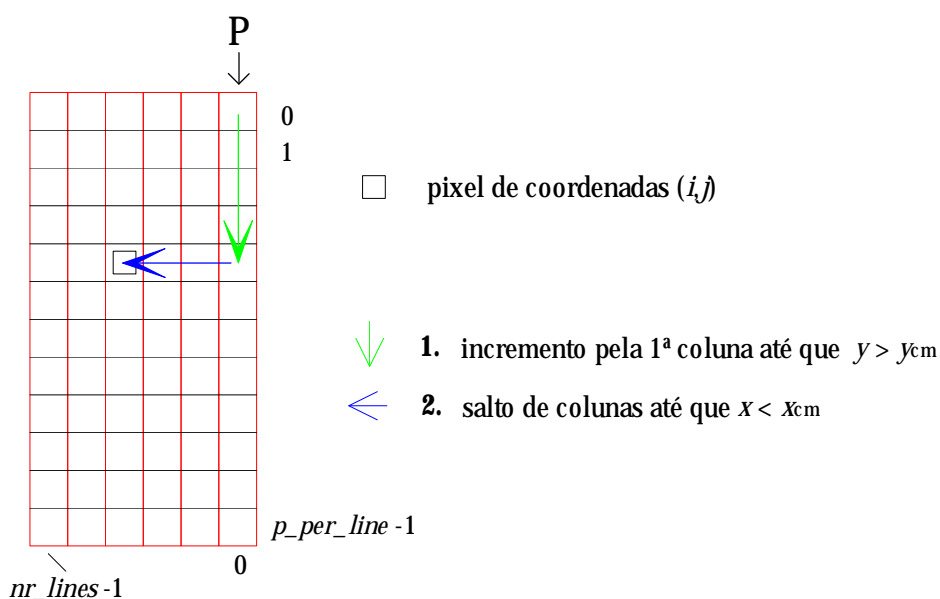


Fig. 44 - Procura do canto inferior esquerdo do "quadrado" delimitador do CM.

Os 3 restantes vértices do "quadrado" delimitador serão então:

vértice superior esquerdo:	$p - 1$
vértice inferior direito:	$p - p_per_line$
vértice superior direito:	$p - 1 - p_per_line$

O valor Z atribuído a (x,y) será a média aritmética dos 4 valores encontrados (nenhum deles é zero já que os pontos (x,y) são frequentemente centros de massa encontrando-se portanto localizados nos centros geométricos das figuras pelo que todos os pontos possuem $Z \neq 0$ para vizinhanças bastante alargadas; tal não será o caso se 'get_Z()' for chamado pelo algoritmo referido em 9.9.2, pelo que aí há que avaliar quantos e quais os pontos para os quais $Z \neq 0$, após o que a média deverá ser encontrada apenas com aqueles pontos).

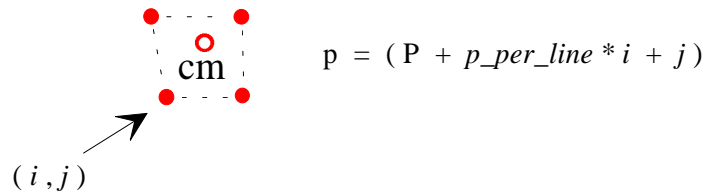


Fig. 45 - Avaliação de Z a associar ao CM dado o "quadrado" delimitador.

Pode ainda dar-se casos em que o ponto (x,y) , supostamente um CM, se encontra em linha com dois pontos de Z definido ou mesmo coincidente com um deles. Esses casos recebem tratamento diferenciado sendo Z encontrado no primeiro caso pela média aritmética de apenas dois valores de Z e no segundo, coincidente, como seria de esperar, com o próprio valor de Z associado a (i,j) .

9.9.4 O Efeito da Altura dos Objectos na Procura do Pixel (i,j)

A aplicação do algoritmo acabado de descrever nunca apresentou problemas para objectos de teste baixos.

Contudo, quando se introduziram objectos de maior altura verificaram-se sistematicamente erros na medida de Z sempre que se utilizava o ficheiro de amostragem de alturas. Isto tem a ver com o tipo de registo que nele é efectuado quando a linha laser não é visível o que corresponde a duas situações: a incidência do feixe laser na parte lateral do objecto ou em zonas de visão bloqueadas pelo próprio objecto.

Quando tal acontece o (x,y) registado é o de um ponto pertencente à linha de referência ao qual é associada a altura $Z=0$. Ora este pormenor de registo dava origem a erros na obtenção do Z para o centro de massa de objectos altos. As duas figuras apresentadas a seguir ilustram estes factos.

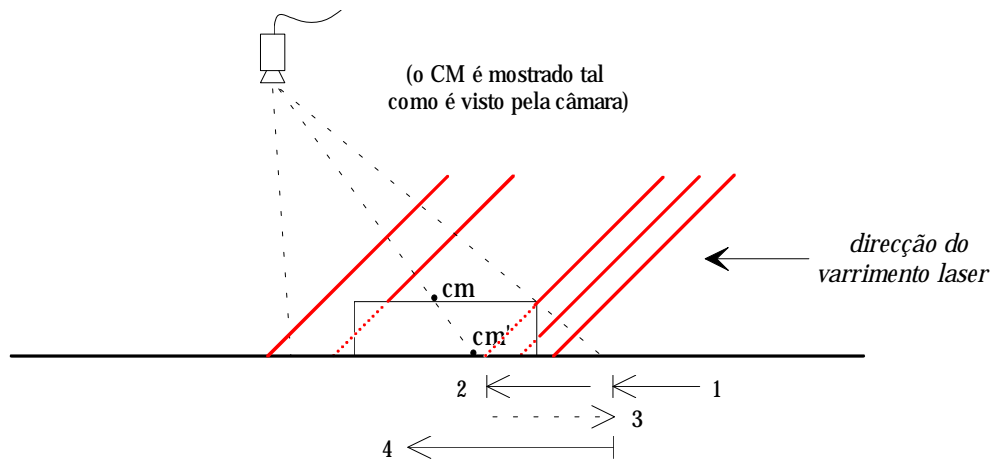


Fig. 46 - Varrimento laser para um objecto baixo:

- (1) a linha laser é vista mover-se suavemente para a esquerda (x decresce).
- (2) a linha laser atinge zonas não visíveis pela câmara: x adquire o valor de X_{ref} (x da linha de referência) que embora decresça, nunca atinge o valor x_{cm} .
- (3) a linha torna-se de novo visível com x a sofrer um aumento brusco para os valores correctos.
- (4) a linha retoma na imagem o seu movimento suave com x a decrescer.

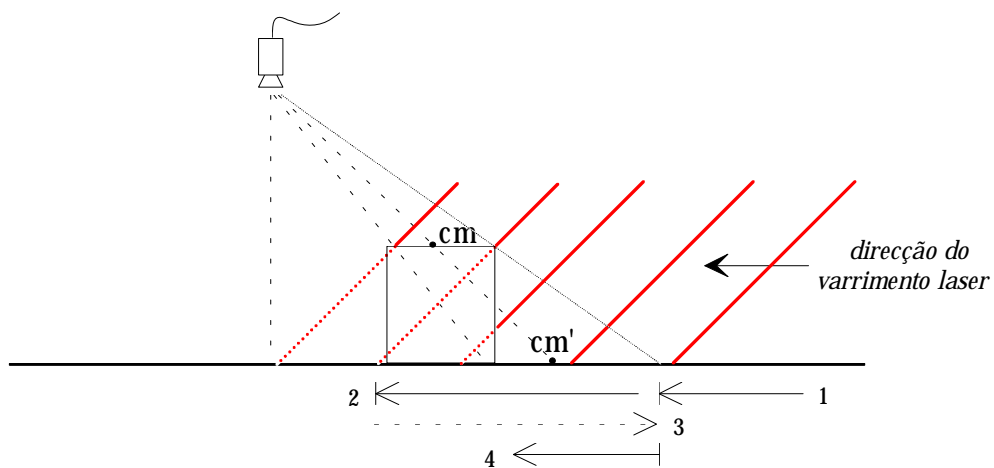


Fig. 47 - Varrimento laser para um objecto alto:

- (1) a linha laser é vista mover-se suavemente para a esquerda (x decresce).
- (2) a linha laser atinge zonas não visíveis pela câmara: x adquire o valor de X_{ref} (x da linha de referência) que decresce e vai abaixo do valor x_{cm} . a procura não deve cessar quando ele é atingido.
- (3) a linha torna-se de novo visível com x a sofrer um aumento brusco, de valores inferiores a x_{cm} para valores superiores a este.
- (4) a linha retoma na imagem o seu movimento suave com x a decrescer, passando de novo por x_{cm} : a procura deve então cessar e Z lido.

Perante esta deficiência encontrada no algoritmo de procura 'get_Z()' houve que contemplar a possibilidade de, aquando do salto de colunas, o x poder descer abaixo de x_{cm} , sem tomar aquele valor para o índice i do canto inferior esquerdo do "quadrado" delimitador do CM. A solução encontrada (e implementada no algoritmo 'get_Z()') foi o de obrigar ao salto de colunas enquanto $x > x_{cm}$ mas sujeito à condição de $Z \neq 0$. Assim, quando o feixe laser incide na parte lateral do objecto e x vai abaixo de x_{cm} , o algoritmo continua a sua procura de x saltando colunas até que elas correspondam de novo ao feixe incidente na parte superior do objecto (situação para a qual Z retoma valores maiores que zero e x adquire valores superiores a x_{cm}). x vai então tomando valores decrescentes sempre com $Z \neq 0$ até que finalmente atinge o CM do objecto parando aí a procura de x e atribuindo este valor ao índice i pretendido.

9.10 A Imagem de Distâncias

Já foi feita referência à imagem de alturas cujo ficheiro associado e registado em disco possui um comprimento total de 204.800 bytes.

Veremos agora como é que é possível obtê-lo tendo como base apenas o ficheiro de amostragem de alturas quer em memória (da placa de aquisição de imagens) quer em disco (em texto ou binário).

Embora haja um pedido expresso no sentido do Agente Laser efectuar a construção de uma imagem de alturas com delimitação espacial especificada no ficheiro de pedidos, ele é mais vulgarmente construído logo após a efectuação de um varrimento laser com obtenção de amostragem de alturas. Obviamente que ela não pode ser efectuada para qualquer resolução devendo esta ser a máxima possível ($res=1$). Assim, sempre que aquela resolução é detectada, é ordenada de imediato a filtragem da imagem do monitor de que resulta a obtenção da imagem de alturas, automaticamente registada em disco.

Essa acção é conseguida mediante a acção de um conjunto de filtros aplicados em série cada um destinado a colmatar lacunas de um determinado comprimento, que se apresentará a seguir:

a) Função de preenchimento na vertical para lacunas de 1 pixel:

Como já foi dito, quando é seleccionada a resolução máxima, a amostragem em y é feita com a periodicidade $res \times 2 = 2$ de forma a ser semelhante à obtida na direcção x ; ou seja, as linhas com informação alternam com outras "em branco". O hiato a preencher é então de 1 *pixel*.

Como se trata de uma filtragem que envolve valores de 3 linhas incluindo aquela a ser alterada e que os desta devem manter-se inalterados até que toda a filtragem dessa linha seja completada (como o impõe qualquer convolução com uma qualquer máscara), há que utilizar um "buffer" temporário onde os resultados são armazenados, aliás, tal como é utilizado nos filtros de imagem bidimensionais

já descritos. Essas 3 linhas não são acedidas directamente ao ficheiro de imagem registado na placa devido à lentidão no acesso individual ao pixel. Em vez disso são lidas as 3 linhas para 3 "buffers" diferentes cada uma lida duma só vez.

Para cada linha a filtrar são então lidas para os respectivos "buffers" as 3 linhas correspondentes. Ora, e uma vez que a linha a ser alterada já forneceu os seus valores originais que a convolução impõe que se mantenham inalterados até que já não sejam necessários para mais cálculos, ela pode ser actualizada na placa de aquisição logo que a filtragem a ela associada termine e também duma só vez.

Ainda para poupar tempo de execução, para se considerar as próximas 3 linhas (com uma linha de incremento) apenas a nova linha (nº 3) é lida mantendo-se os valores das anteriores 2ª e 3ª linhas que agora passam a ser a 1ª e a 2ª. Para tal não se efectua qualquer cópia dos "buffers" 2 e 3 para os 1 e 2! Em vez disso trocam-se os valores dos respectivos apontadores! Ou seja, *ptr1*, o apontador para o 1º "buffer" cujos valores serão ignorados passará a ter o valor de *ptr2*: *ptr1 = ptr2*; *ptr2* passará a ter o valor de *ptr3* e este tomará o valor de *ptr1* ou seja, do "buffer" cujos valores já não interessam por se referirem à linha a ser ignorada, e que receberá os valores da nova linha a ser considerada (abaixo). Ou seja, os valores de *ptr1*, *ptr2* e *ptr3* vão rodando entre si por forma a evitar perda de tempo na cópia dos conteúdos dos "buffers" (para além de ser uma solução mais elegante do ponto de vista de programação!).

Como a linha a ser registada em definitivo na placa de aquisição é a nº 2 cujos valores calculados se encontram no "buffer" temporário e cujos valores antigos ainda a ser utilizados na convolução seguinte se encontram no "buffer" nº 2 que passa a ser o nº 1, a imposição de que os valores dos pixels só sejam alterados quando não são mais utilizados para cálculo numa convolução, é devidamente respeitada!

O resultado desta filtragem é visível na Fig.53.

b) Função de preenchimento na horizontal para lacunas de 1 pixel:

Nesta função a filtragem já é efectuada apenas com os valores da própria linha a ser modificada pelo que se dispensa o uso daqueles 3 "buffers". Contudo deveria em princípio manter, tal como anteriormente, o valor original de um dado pixel até que ele não seja necessário para mais nenhuma convolução. Assim, poderia utilizar 3 variáveis que fariam o papel dos 3 "buffers" anteriores, rodando-se do mesmo modo que anteriormente os respectivos apontadores. Mais simples que isso poderia ser o registo de cada convolução num "buffer" (o que implica um total de 2 "buffers": o da linha original e o da linha a ser registada com as respectivas alterações) que seria posteriormente registado por inteiro na linha considerada.

Tal uso de 2 "buffers" não é contudo necessário podendo neste caso registar-se cada resultado de uma convolução no próprio "buffer". A razão disto reside no facto da convolução só ser efectuada quando uma determinada condição é verificada: neste caso quando se encontra uma combinação de 1-0-1 para 3 pixels consecutivos (existência de uma lacuna com 1 pixel de largura). Ora, quando o

pixel sem valor adquire o da média aritmética do conteúdo dos pixels limitadores, fica a combinação 1-1-1, ou seja, ficam os 3 pixels definidos pelo que no incremento seguinte dos apontadores o 1 central que passa a ser o limite esquerdo doutra combinação, não será mais utilizado uma vez que essa nova combinação só poderá ser 1-1-0 ou 1-1-1; e nenhuma destas combinações obedece à condição imposta para a alteração do pixel central!

Se, por outro lado, se mantivesse o valor original do pixel só teríamos as combinações 0-1-0 ou 0-1-1 que também não obedecem àquela condição.

O resultado desta filtragem é visível na Fig.52.

c) Função de preenchimento na vertical para lacunas de 2 pixels:

Este filtro destina-se a colmatar lacunas com 2 pixels de largura correspondendo à combinação 1-0-0-1. Do mesmo modo que anteriormente é possível registrar o resultado no mesmo "buffer" que contém os valores originais da linha uma vez que só o último pixel da direita poderá encontrar a mesma condição (1-0-0-1) e nesse caso nenhum dos pixels alterados entra no cálculo da média para inserir nos pixels seguintes não preenchidos.

O resultado desta filtragem é visível na Fig.51.

d) Função de preenchimento na vertical para lacunas de 3 pixels:

Neste filtro são consideradas lacunas com 3 pixels de largura, ou seja, combinações do tipo 1-0-0-0-1. Também neste caso o resultado da média aritmética dos 2 pixels extremos pode ser registado directamente no "buffer" preenchendo assim os 3 pixels 0. Nenhum deles é considerado noutra combinação igual que quanto muito poderá envolver o último pixel 1 (deixando portanto de fora os pixels acabados de definir).

O resultado desta filtragem é visível na Fig.50.

e) Função de preenchimento na vertical para lacunas de 4 pixels:

Neste outro filtro são consideradas lacunas com 4 pixels de largura, correspondendo à combinação 1-0-0-0-0-1.

Esta função não é normalmente utilizada pois raramente se encontram lacunas desta largura em amostragens com a resolução máxima.

Por experimentação verificou-se que a sequência mais eficaz de aplicação daqueles filtros é precisamente a inversa da sequência da sua apresentação neste texto; e

daí as figuras na ordem em que se encontram. Desta sequência não faz parte o filtro apresentado em e) por se destinar a lacunas de 4 *pixels* que normalmente não ocorrem em amostragens com resolução máxima (apenas esporadicamente quando o sistema óptico não se encontra bem ajustado, com particular destaque para a incorrecção no posicionamento da lente cilíndrica).

9.10.1 Sequência de Aplicação dos Filtros num Caso Real

Como resultado do varrimento laser efectuado na resolução máxima ($res=1$) para a cena mostrada na Fig.48, obteve-se uma imagem com amostragens como mostra a Fig.49. A representação gráfica que a constitui foi obtida por inspecção do ficheiro de imagem registado na placa de aquisição de imagens e visualizado no monitor. Da execução deste programa resulta uma lista textual das coordenadas x e y dos pixels de altura, que lida por uma aplicação gráfica dá origem à representação bidimensional da imagem de amostragens mostrada naquela figura.

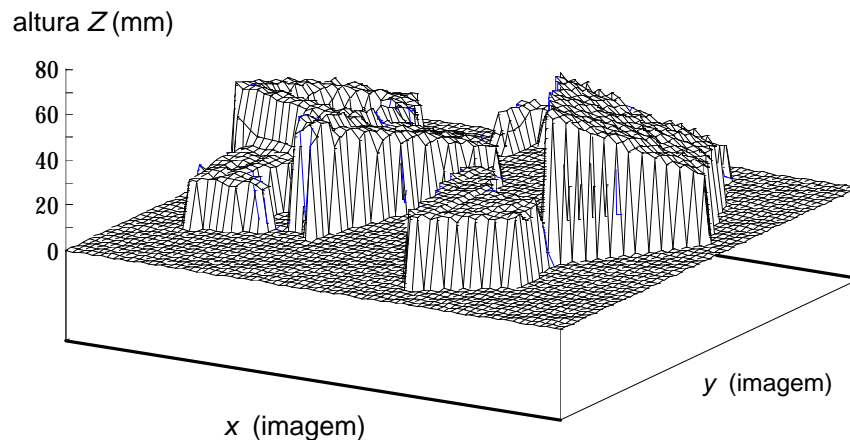


Fig. 48 - Representação tridimensional dos objectos, com base no ficheiro de amostragens obtido após varrimento laser.

Por aplicação sucessiva dos filtros descritos na ordem inversa da sua apresentação, obtém-se como resultado a sequência das Figs. 50 , 51 , 52 e 53. No final obtém-se como se esperava a imagem de alturas, na qual todos os pixels dentro da fronteira dos objectos possuem valores de altura associados (ou seja, não há lacunas).

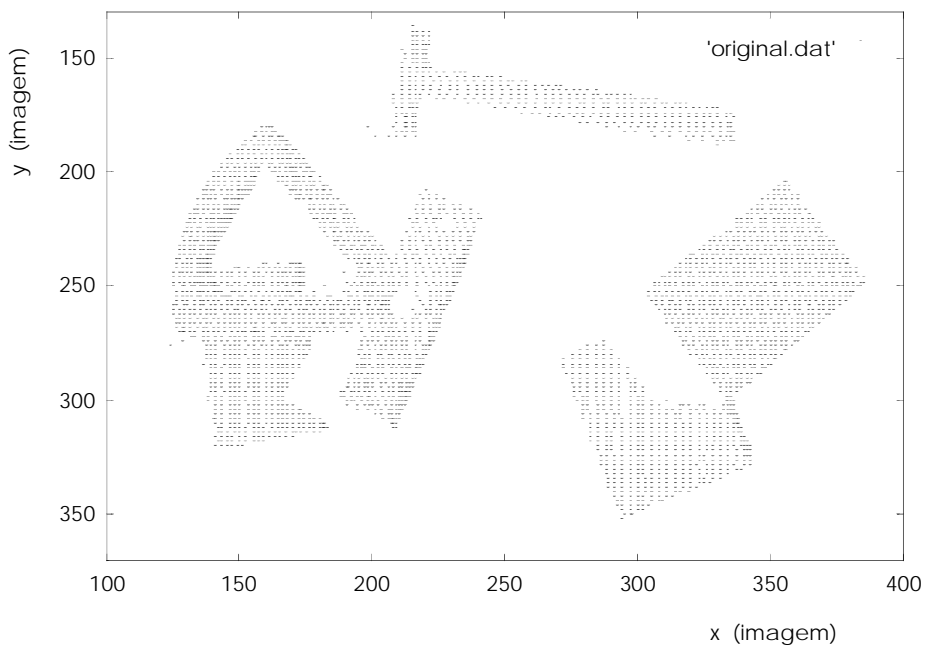


Fig. 49 - Figura original resultante de um varrimento laser com $res=1$.

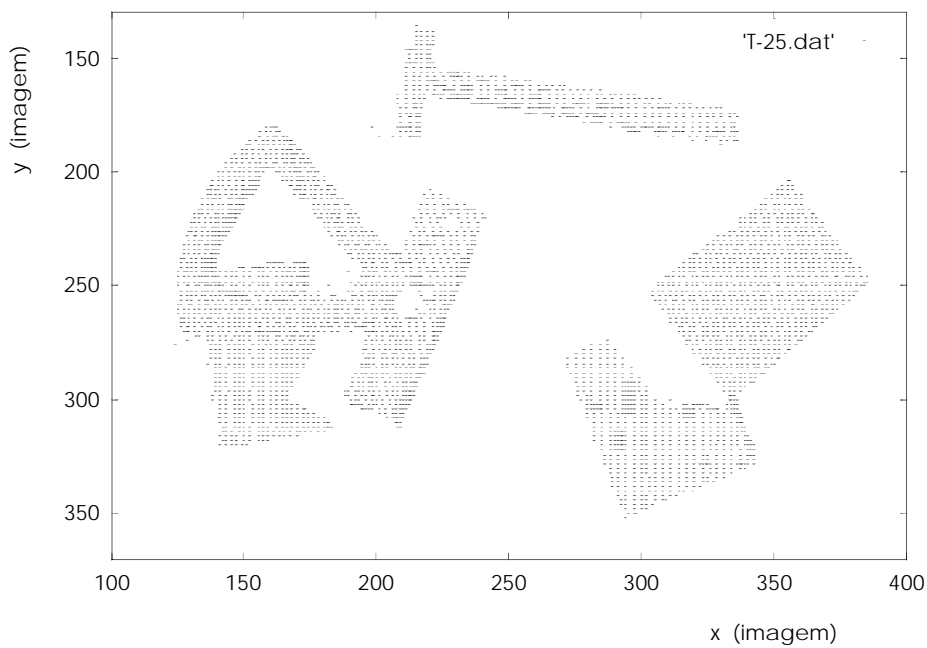


Fig. 50 - Figura obtida após a aplicação do filtro referido em **d**).

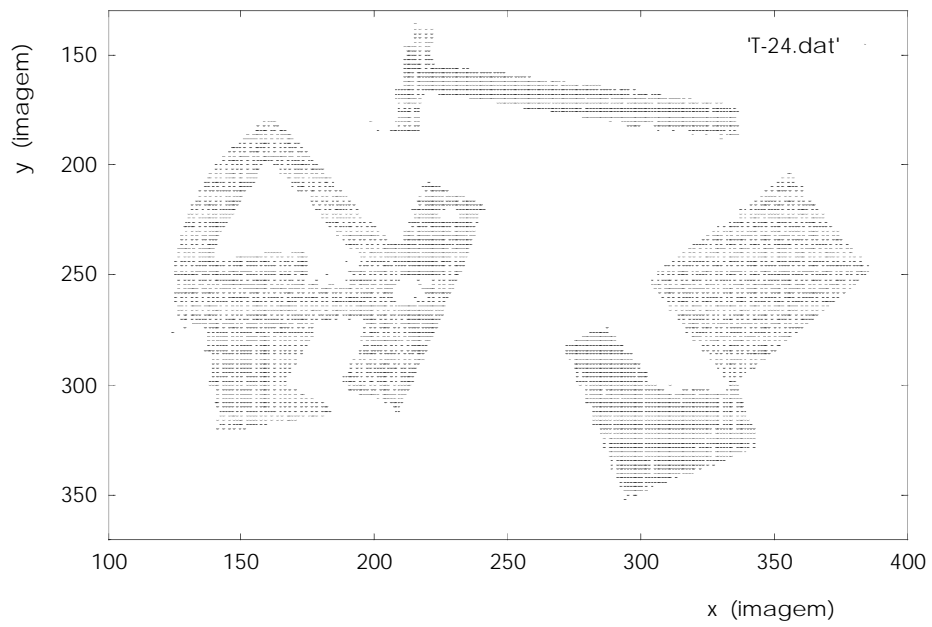


Fig. 51 - Figura obtida após a aplicação do filtro referido em c).

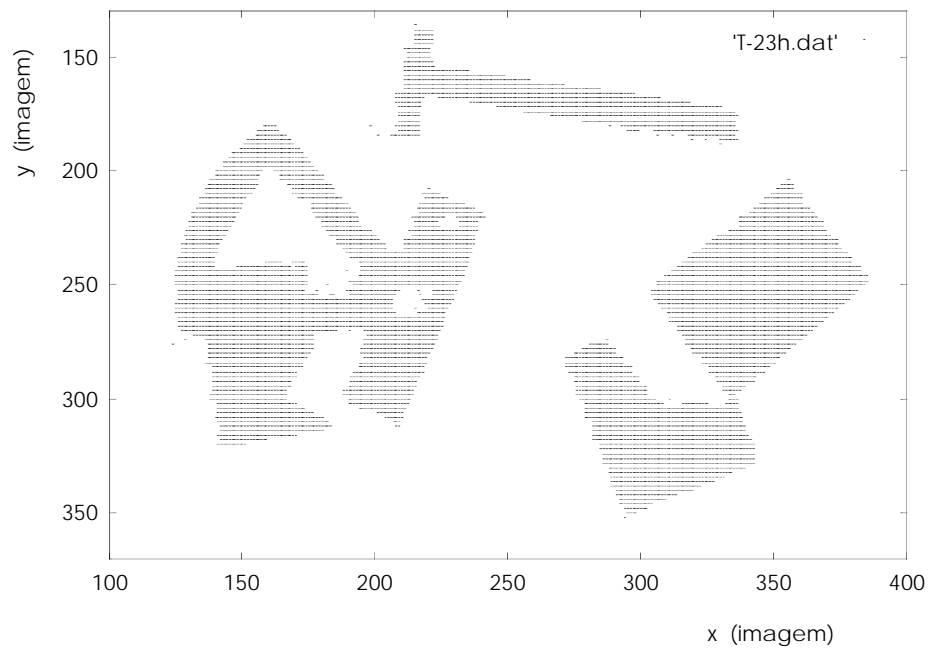


Fig. 52 - Figura obtida após a aplicação do filtro referido em b).

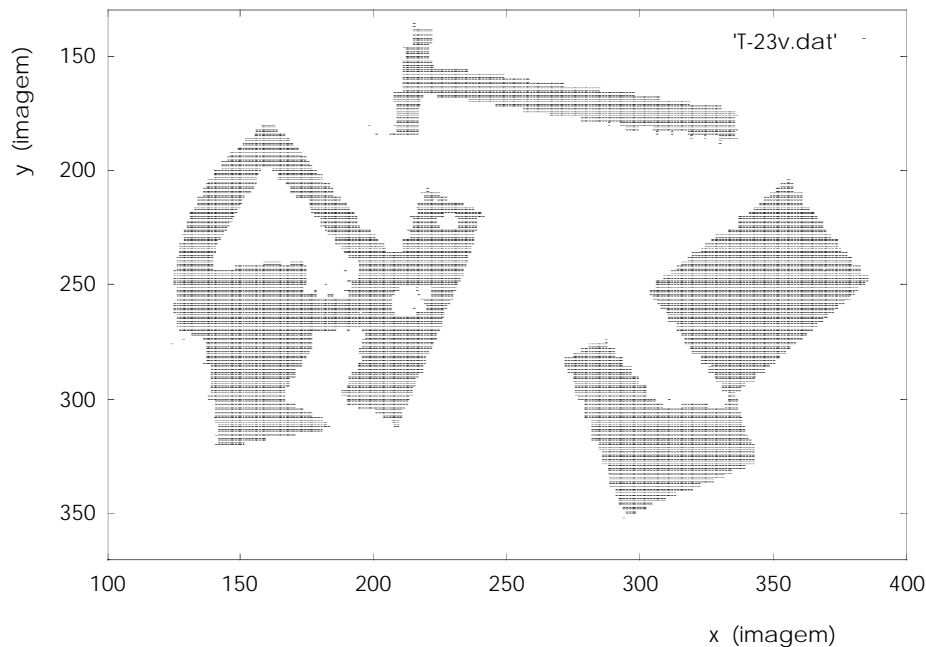


Fig. 53 - Figura final obtida após a aplicação do filtro referido em a).

Para a obtenção destas representações foi necessário seleccionar a resolução máxima para o varrimento laser da cena (condição necessária para desencadear todo o processo de filtragem da imagem de alturas), ou seja, usando $inc_x=1$ e $inc_y=1$.

No Anexo G.3 é apresentado um esquema de chamadas da função principal de filtragem.

9.10.2 O Registo em Ficheiro da Imagem de Alturas

Uma vez conseguida a imagem de alturas com resolução igual à da imagem de intensidades, esta é gravada num ficheiro próprio com 204.800 bytes de extensão. Neste ficheiro são apenas gravados os blocos 2 a 7 com 40KB cada (64×640 bytes), correspondendo assim aos limites de acção do robot detectados na imagem. As linhas limite são então a nº 128 e a nº 447 (Fig.54).

Uma vez que, ao contrário do que acontece nos ficheiros de amostragem de alturas, este ficheiro não contém valores reais de altura em *mm*, mas tão somente valores codificados em nível de cinzento de acordo com a relação:

$$\text{nível_de_cinzento} = Z \times \left(\frac{255}{Z_{\text{máx}}} \right)$$

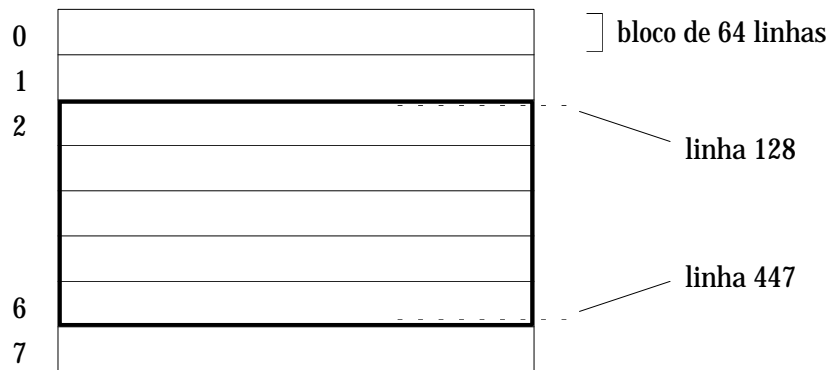


Fig. 54 - Registo em disco de apenas 5 dos 8 blocos do ficheiro de imagem.

então, para que seja possível reconverter estes valores nos respectivos valores de altura em *mm* após lido aquele ficheiro, houve que registar o valor $Z_{máx}$ algures. Foi então decidido registá-lo na própria imagem nas coordenadas (0,128) e (1,128), ou seja, nos primeiros dois pixels da 1ª linha, ficando o 1º pixel reservado para o valor inteiro de $Z_{máx}$ e o 2º para a parte decimal (esta é obtida multiplicando o registo no pixel por 256). Sendo assim tanto o varrimento laser como todo o processo de filtragem excluiu aquela linha para registo de valores, passando então os limites a ser as linhas nº 129 e nº 447.

9.11 Conclusão

Neste capítulo foram apresentadas formulações matemáticas que permitem obter medidas correctas de altura para vários pontos de uma cena, cujo último resultado após aplicação de filtros especiais, quando o varrimento tiver sido efectuado na sua resolução máxima, é a imagem de distâncias. Nesta, todos os pontos dos objectos possuem uma altura definida cujo valor é codificado em nível de cinzento (portanto entre 0 e 255) consoante o valor de altura máxima detectada. Este valor é registado na própria imagem de alturas para uma adequada conversão *nível-de-cinzento* → *altura-em-mm*.

Para que tal fosse possível foram considerados ajustes na fórmula inicial de acordo com o efeito quer da perspectiva quer da distorção da imagem. Foram também medidos parâmetros geométricos e mecânicos associados à montagem do sistema de varrimento laser, através da sua calibração cuidada.

Este sistema de medida pertence ao Agente autónomo Laser, e é chamado numa das suas várias modalidades de funcionamento, após e consoante o pedido efectuado por algum dos restantes agentes.

10.1 Introdução

Após o desenvolvimento do sistema de varrimento laser e da construção das imagens de altura ficou finalmente aberto o caminho para a medição de parâmetros para se poder efectuar identificações de objectos numa cena.

Neste capítulo far-se-á a apresentação desses parâmetros assim como a descrição do modo de os obter e que irão constituir aquilo a que se convencionou chamar "Assinaturas" dos objectos. Estas "Assinaturas" deverão ser utilizadas posteriormente para a sua identificação.

Para que esses parâmetros possam ser utilizados com sucesso na identificação, têm que possuir um elevado grau de fiabilidade. Sendo assim decidiu-se efectuar um estudo da variação dos parâmetros que constituem a "Assinatura" em função da posição na imagem de um mesmo objecto, e que será apresentado neste capítulo.

Finalmente, será apresentado o ficheiro de modelos 3D a ser registado em disco e discutida a sua estrutura. Será ainda descrito o programa que foi desenvolvido para a sua construção e gestão.

10.2 A Construção das "Assinaturas"

Como já foi visto atrás, o método de distinção entre dois objectos sobrepostos, baseia-se na análise das superfícies planas paralelas a XOY, pois só desse modo, aparecerão dois picos bem distintos no histograma de alturas e portanto, separáveis. Uma vez na posse de parâmetros associados a esses picos, bastará correr de novo o programa de segmentação 2 vezes, com a informação do intervalo de valores de altura (ou mais concretamente de nível de cinzento, uma vez que as alturas se encontram codificadas desse modo) assim como do rectângulo que confina o conjunto para mais rápida execução, para que se possa obter informação separada (completa ou não) de todos os objectos que compõem o conjunto. Esta restrição pode não permitir a separação de objectos em sobreposição se estes forem complexos, em particular se estes não

possuírem faces planas paralelas a X0Y, mesmo quando se encontram num dos seus estados estáveis. Para esses casos, não se possui por enquanto, nenhum algoritmo que permita a sua separação, mas ele poderá de futuro ser desenvolvido. Apesar disso, a restrição contempla um número elevado de casos, já que muitas peças de montagem, quando se encontram num dos seus estados estáveis, possuem de facto essa superfície.

Pode-se portanto concluir que esta abordagem é correcta e permitirá resolver um grande número de situações de oclusão.

Foi então desenvolvido um programa que para cada mancha encontrada na imagem, analisa automaticamente o histograma de alturas e separa informação relativa a cada superfície plana encontrada após isolamento dos respectivos picos no histograma.

Essa informação é subsequentemente inserida nos respectivos campos de uma estrutura de dados especialmente construída para este tipo de análise e a que se convencionou chamar "Assinatura" do objecto.

Essa estrutura possui um campo associado ao seu estado; ele apenas pode tomar algumas denominações: " ", "unknown", "visible", "occluded" ou "rotated" consoante se trate respectivamente de um objecto isolado, sobreposição impossível de resolver, em sobreposição, sobreposto ou rodado no espaço.

Os parâmetros *sumZ2* e *sumNr2* que se decidiu incluir na informação da "Assinatura", embora não impliquem a necessidade daquelas faces planas existirem, podem vir a complementar a informação já existente, tornando o processo mais eficiente. O 1º é calculado ainda no programa de segmentação quando do varrimento da imagem de alturas para a medição de parâmetros 2D das manchas e consiste no momento de inércia geométrico relativamente ao plano XY, da figura representada na imagem (em mm²). Este valor é uma característica do objecto e portanto potencialmente útil na sua identificação. O parâmetro *sumNr2* é também um momento de 2ª ordem mas calculado a partir do histograma filtrado associado ao objecto, ou conjunto de objectos em sobreposição: ele consiste, como o próprio nome sugere, no somatório do quadrado do número de pixels associado a cada altura (em codificação "grey-level") e, se não houver rotação, constitui um invariante do objecto perante desvios do seu posicionamento (modificação da posição Z devido a sobreposição) uma vez que esse desvio implica apenas um deslocamento no eixo dos ZZ (ou *pp* na codificação em "grey-level") de todo o histograma.

Finalmente, há 2 campos que registam respectivamente os valores de altura das superfícies planas num máximo de três e a sua área (*Zs* máximos dos picos no histograma, e os somatórios dos pixels relativos a cada pico) (Fig.1). Para facilidade na compreensão do seu futuro uso refira-se que se trata dos vectores *Z[]* e *nr[]*.

Para efectuar todas aquelas medidas, é associado um histograma a cada mancha que é construído aquando da segmentação inicial da imagem de alturas ("range image") e à medida que o varrimento da imagem de alturas vai sendo feito. Esse histograma vai sendo preenchido numa área de memória reservada, com 256 inteiros, correspondentes a cada um dos níveis de cinzento ("grey-level") cujo apontador, aliás, faz parte da informação associada a cada uma dessas manchas, ou seja, está incluído na respectiva estrutura de dados.

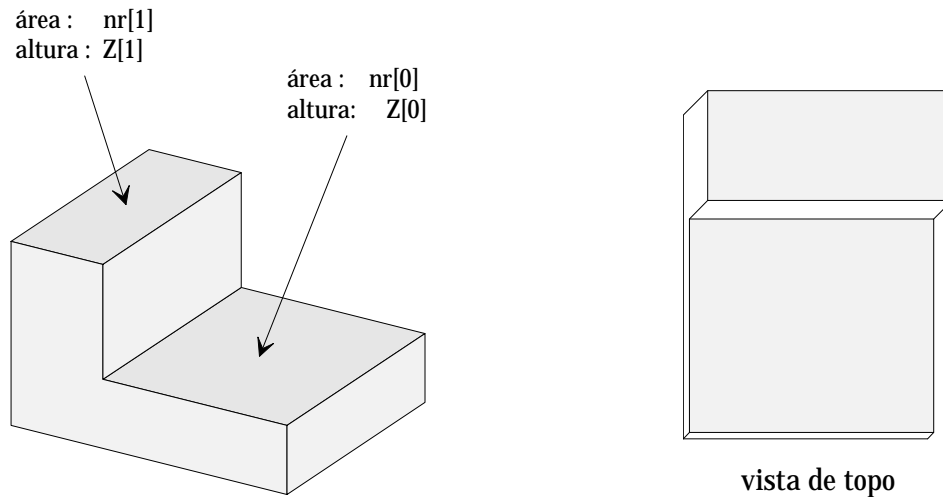


Fig. 1 - Medição dos parâmetros relativos às faces planas paralelas ao plano horizontal; esses valores são extraídos do respectivo histograma 3D.

Essa informação mantém-se em memória enquanto a mancha a ela associada for válida, podendo ser alterada por filtragem do histograma ou sempre que houver qualquer actualização de dados, como será o caso de uma 2ª segmentação da mancha se ela se referir a mais do que um objecto (sub-segmentação).

Apresenta-se de seguida várias situações possíveis com a respectiva construção das "Assinaturas".

10.2.1 Poliedros Sobrepostos e nos seus Estados Estáveis

A situação mais simples de oclusão que poderá ocorrer é a de objectos simples do tipo poliédrico em situação de sobreposição. Neste caso a construção do histograma de alturas (n° de pixels em função da altura, ou mais concretamente, do nível "grey-level" que corresponde à altura codificada) vai revelar a ocorrência de dois picos bem definidos, cada um correspondente a uma das alturas encontradas.

Para além dos dois parâmetros extra acrescentados à estrutura das "Assinaturas", $sumZ2$ e $sumNr2$, são preenchidos dois dos três elementos de entrada de cada um dos vectores $Z[]$ e $nr[]$. $Z[0]$ e $Z[1]$ serão as alturas máximas associadas respectivamente ao 1º e 2º pico, ou seja, a altura aproximada a que se encontram as duas superfícies planas detectadas. Se as duas superfícies não pertencerem ao mesmo objecto, então $Z[1]$ é subtraído por $Z[0]$ por forma a corresponder à altura do 2º objecto e não à distância a que se encontra da superfície XY (esta situação ocorre quando a "Assinatura" contendo os dois picos não é reconhecida como correspondendo a nenhum objecto conhecido, ou seja, contido no ficheiro de modelos 3D).

Em $nr[0]$ e $nr[1]$ são registados os números de pixels encontrados em cada pico do histograma (integral da função desenhada por cada pico em ordem à codificação "grey-level" da altura dentro dos limites que os definem) e que correspondem ao valor aproximado das áreas de cada uma das duas superfícies.

A Fig.2 apresenta uma representação tridimensional de dois objectos numa situação como a descrita acima, e a Fig.3 mostra o histograma 3D já filtrado.

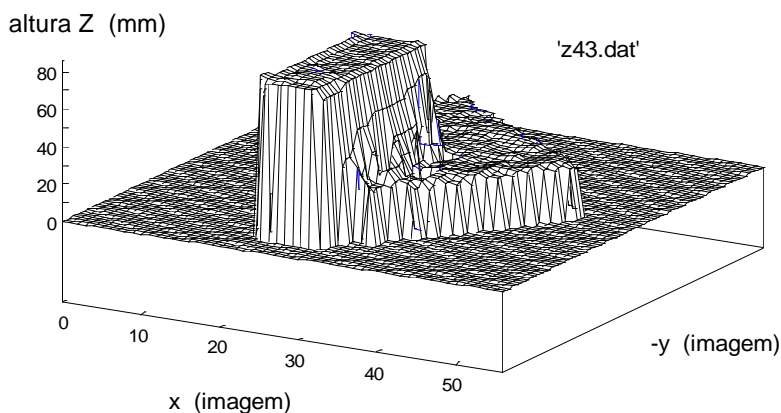


Fig. 2 - Representação tridimensional de dois objectos em situação de sobreposição parcial.

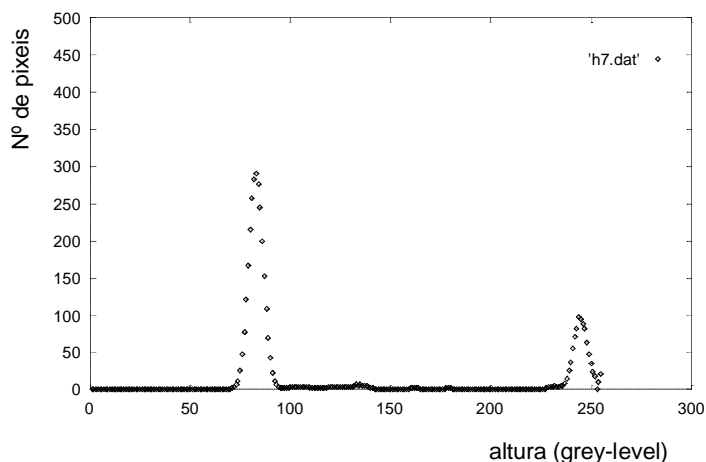


Fig. 3 - Histograma 3D respeitante à situação da figura anterior.

Um possível resultado da execução de segmentação e medida de "Assinaturas" naquela situação, seria o seguinte:

1 Range Blob
----- Blob 0 -----
jmin=216
imin=275 imax=353


```

jmax=324
area=3923
x=315.2, y=268.7
----- signature -----
sumz2 = 9488312 (mm2)
sumNr2 = 15894 (pixel2)
nr_peaks = 2
----- peak 0 -----
Z[0] = 27; nr[0] = 2667
----- peak 1 -----
Z[1] = 83; nr[1] = 914
-----

```

Situações um pouco mais complexas podem ocorrer para um máximo permitido de três picos, como sendo (1) um objecto contendo duas superfícies planas parcialmente ocluído por um outro contendo apenas uma superfície plana ou (2) um objecto simples com apenas uma superfície plana sendo encoberto por um outro contendo duas superfícies planas. Contudo, dada a pouca probabilidade de tais ocorrências o algoritmo de sub-segmentação interpreta a existência dos vários picos como resultado da sobreposição de dois ou três objectos contendo cada um apenas uma superfície plana paralela a XY, desde que aquele conjunto não seja identificado de imediato como pertencendo a apenas um objecto descrito no ficheiro de modelos 3D. Aquelas situações, embora não contempladas na actual versão do programa poderão sempre e em qualquer altura ser consideradas, desde que tal se justifique.

10.2.2 1º Poliedro em Sobreposição num Estado "Não Estável" e o 2º Sobreposto e Parcialmente Visível

A contemplação neste sistema de reconhecimento de apenas objectos com superfícies planas paralelas ao plano XY pode parecer limitativa. No entanto, ele constitui um processo de identificação rápido, utilizando as imagens de alturas, para aqueles objectos que de facto ocorrem com mais frequência em linhas de montagem. Objectos doutro tipo não são reconhecidos por este sistema. Mas ele constitui apenas uma alternativa ao processo de identificação a 2D que já os contempla. Os dois sistemas, aparentemente concorrenciais uma vez que são executados em máquinas ou processos separados e são supostos fornecer resultados do mesmo tipo e formato, provam ser na realidade complementares: o sistema a 2D contempla objectos de qualquer complexidade na condição de não ocorrer qualquer sobreposição enquanto que o a 3D já prevê esta ocorrência mas impondo uma relativa simplicidade nos objectos em estado de sobreposição.

Mantendo então a condição dos objectos serem simples (neste caso poliédricos) a situação que poderá dificultar quer a identificação quer a manipulação robótica posterior, é a rotação no espaço do objecto em sobreposição: caso em que este último se apoia simultaneamente no plano XY e no 2º objecto (Fig.4).

Nesta situação o 2º pico do histograma desaparece dando lugar a uma "faixa" ou "banda" de altura constante (cuja pequena altura corresponde ao também pequeno nº de pixeis que se encontram à mesma altura) entre um valor Z_0 e outro Z_1 (Fig.5). Estes valores correspondem aos valores extremos de altura encontrados para o objecto, ou seja, dos valores de Z associados às suas duas extremidades. Uma vez que o objecto é poliédrico e portanto possui apenas uma superfície plana (condição necessária para que este sistema possa processar a informação) o nº de pixeis associados à mesma altura é constante o que implica que a faixa tenha também altura constante,

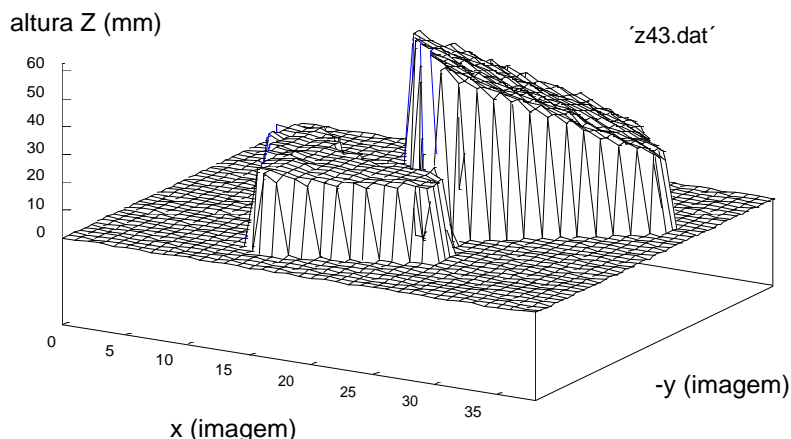


Fig. 4 - Representação tridimensional de dois objectos poliédricos em que o 2º se encontra rodado no espaço.

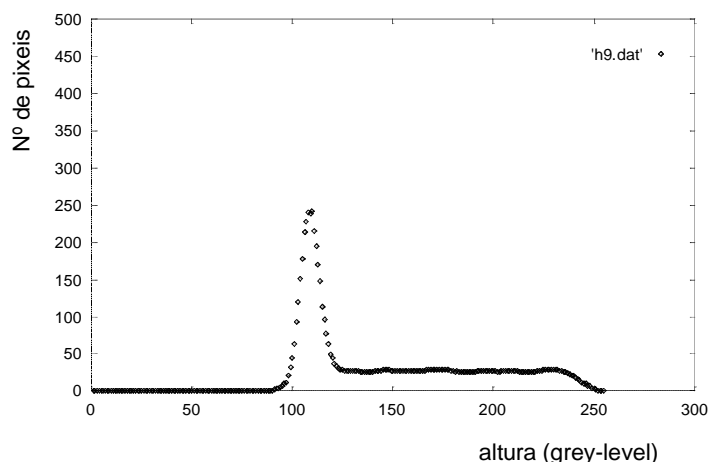


Fig. 5 - Histograma 3D referente ao caso da figura anterior: a rotação no espaço deu origem a uma "faixa".

Nesta situação não é possível efectuar qualquer reconhecimento: o objecto sobreposto não é visto completo e o 2º objecto não apresenta a sua superfície plana paralela a XY impossibilitando a parametrização do pico associado. A "Assinatura" é no entanto preenchida na mesma, não com parâmetros identificadores das superfícies mas com valores numéricos que permitam o seu manuseamento robótico posterior. Por exemplo,

o ângulo de inclinação do objecto, que deve corresponder também à rotação da garra aquando do manuseamento, é obtido através do arco da tangente da razão entre a diferença de alturas Z encontradas (os Z s associados às extremidades do objecto) e o seu comprimento.

10.2.3 A Rotação no Espaço do Objecto em Sobreposição

Nas situações em que apesar da oclusão, os objectos mantinham as respectivas superfícies planas, paralelas ao plano XOY , ou seja o objecto superior sofria apenas uma translação no espaço, o algoritmo inicialmente desenvolvido que esperava encontrar picos no histograma de alturas, correspondia perfeitamente. Porém, verificou-se que a situação de sobreposição implicava frequentemente a rotação no espaço da peça superior, ficando portanto apoiada quer no objecto sobreposto quer no plano XY . O histograma correspondente poderia então quanto muito apresentar apenas um dos picos. Em vez do segundo pico, aparece uma faixa que traduz o aumento gradual das alturas dos pontos do objecto (Fig.5).

Ora, mesmo que não se pretenda uma identificação rigorosa desse objecto (ou ela não seja possível) é pelo menos imperioso que o sistema seja capaz de reconhecer essa situação e de conseguir lidar com ela. Ou seja, é necessário que o algoritmo consiga reconhecer as várias situações possíveis de oclusão, que implicará a detecção da faixa da figura anterior e a extracção de parâmetros úteis à manipulação futura do respectivo objecto. Com esse fim em vista, o 1º algoritmo desenvolvido, foi remetido para uma função: *case_peaks_only* () que mantém a detecção apenas de picos (num máximo de três por mancha) no histograma de alturas. A nova abordagem do programa na sua versão mais actual contempla então as várias combinações possíveis passando por uma fase inicial de detecção das duas figuras possíveis, pico e banda:

1. detecção da existência de pelo menos **um pico**
2. detecção da existência de **uma faixa** independentemente da existência simultânea de picos (correspondente à situação em que o objecto superior está rodado no espaço)
3. chamada de uma a quatro funções de processamento de acordo com a combinação de picos e faixa encontrada:
 - 3 a. Parametrização e registo de **um ou mais picos** (num máximo de 3)
 - 3 b. Parametrização e registo de **uma faixa** (ou banda)
 - 3 c. **Pico e banda** (occlusão parcial e rotação no espaço do objecto superior)
 - 3 d. **Nenhum** dos casos anteriores (situação complexa, impossível de reconhecer)

A divergência para cada uma destas funções depende como foi visto, da detecção prévia de picos e bandas através do uso das primitivas.

10.2.3.1 As Primitivas de Detecção Primária de Picos e Bandas

1. Ocorrência de 1 pico

O objectivo desta primitiva é detectar a ocorrência de pelo menos 1 pico. Ele é facilmente atingido pela procura no histograma de alturas de um nível de cinzento ao qual é associado um número de pixels superior a 50. Se aquele histograma tiver sido previamente filtrado, então aquele número não corresponderá a nenhum pico secundário derivado do ruído na imagem de alturas e então ficará detectada a ocorrência de pelo menos 1 pico. É ainda procurado o valor máximo de Nr , ou seja, do número de pixels associados a um nível de cinzento que é devolvido à função chamante como valor de retorno.

2. Ocorrência de 1 faixa (banda)

Uma banda é caracterizada por possuir poucos pixels associados a cada altura (tipicamente entre 15 e 45, como foi verificado experimentalmente) e abranger uma largura apreciável do histograma 3D. Para evitar a detecção errónea de uma banda no caso de 1 pico, é exigido que a variação no número de pixels para níveis de cinzento consecutivos não seja superior a 6 (valor também observado experimentalmente).

Sempre que se detecta a ocorrência de $15 < Nr < 45$ com uma tal variação inferior a 6, a função entra num ciclo de medida em que é actualizada uma variável que regista a extensão da banda em níveis de cinzento assim como o valor médio da sua altura em termos de nº de pixels. Se a largura da banda detectada for superior a um dado valor pré-definido (nos casos estudados o valor 80 pareceu ser um mínimo exigível) então o ciclo de procura é abandonado e o 1º nível de cinzento associado à banda é retornado.

Quando alguma das condições não é observada, a procura de banda é retomada a partir do último nível de cinzento considerado.

Temos então a informação sobre a existência ou não de pico e banda no histograma 3D e parâmetros a eles associados como sendo o valor máximo do nº de pixels no caso de 1 pico, e o valor médio da altura da banda também em nº de pixels. A combinação de existência de pico e banda determinará a divergência na escolha da função para tratamento posterior da informação no qual os parâmetros retornados são usados.

10.2.3.2 As 4 Funções de Maior Nível

1. Ocorrência de picos apenas

Na situação em que não são detectadas faixas mas apenas picos (um ou mais), então ou picos pelo que é chamada uma função adequada que se encarrega de determinar o nº de picos existentes no histograma 3D e de extrair os parâmetros correspondentes que são registados na "Assinatura" da mancha.

Nesta função é efectuado um varrimento aos elementos de entrada de todos os níveis de cinzento até ao seu valor máximo, 255. Sempre que é detectado no histograma filtrado um valor de Nr superior a 15, é iniciado o somatório dos valores de Nr correspondentes aos mesmos níveis de cinzento mas lidos no histograma original não filtrado e a verificação do nível de cinzento correspondente ao máximo; esta operação cessa logo que Nr (do histograma filtrado) retoma valores inferiores a 3 após o que é verificado o valor daquele somatório. Se ele for superior a 500 então o pico detectado corresponde de facto a uma superfície real, paralela ao plano horizontal e os valores medidos são considerados para registo na respectiva "Assinatura": $nr[]$ vai conter aquele valor de somatório enquanto que $Z[]$ recebe o valor de altura máxima detectada em mm, ou seja, após o nível de cinzento correspondente ter sido multiplicado pelo factor de conversão ($Z_{máx}/255$).

2. Ocorrência de 1 faixa sem picos

Na situação em que as funções de detecção de picos e faixa retornam respectivamente 0 e um inteiro *first*, então apenas foi detectada uma banda com início no nível de cinzento indicado por aquele inteiro. Nesta função repetem-se as medidas efectuadas na função detetora de faixa uma vez que só aquele valor foi retornado: trata-se da extensão da banda, do somatório de todos os pixeis associados a ela e retirados do histograma original e finalmente o total de pixeis da banda, mas estes agora retirados do histograma filtrado. Aquele valor vai fornecer de imediato a projecção da área da superfície visível no plano XY que é registada na "Assinatura" enquanto que este fornece o valor médio da altura da banda por divisão da sua extensão. Embora não se tenha detectado nenhum pico mantém-se a verificação de que não existe nenhuma variação brusca de Nr ($dnr \leq 6$) e que este também não excede 45.

Como já foi visto, a "Assinatura" neste caso não irá conter parâmetros característicos do objecto mas antes valores futuramente úteis à sua manipulação robótica de forma a eliminar a sobreposição. Os dois valores de Z a ser registados em $Z[0]$ e $Z[1]$ são as alturas mínima e máxima da superfícies (que posteriormente fornecerá o ângulo de rotação no espaço) calculados a partir dos dois níveis de cinzento delimitadores da banda: *first*, já obtido antes e $last = first + \text{extensão}$; após multiplicados pelo factor de conversão ($Z_{máx}/255$). $nr[0]$ e $nr[1]$ conterão respectivamente o valor médio da banda retirado do histograma filtrado, e a soma dos pixeis pertencentes à banda no histograma original, ou seja, a projecção da área da superfície que lhe corresponde no plano XY. $Z[2]$ e $nr[2]$ são mantidos a zero.

3. Ocorrência simultânea de picos e faixa

Esta função é chamada quando ambas as figuras são detectadas no histograma filtrado. Logo à partida dispõe-se dos valores de retorno das duas funções de teste: o

valor máximo do 1º pico detectado, e *first*, o 1º nível de cinzento associado à banda. Sendo assim, fica facilitada a tarefa de recolher os parâmetros a registar na "Assinatura": inicia-se a execução pela detecção do início do pico que aparece no histograma para valores mais pequenos de nível de cinzento seguindo-se a medida do somatório de *Nr* até que se atinja *first*, o 1º nível de cinzento associado à banda; ela prossegue então nas medidas da extensão, do valor médio da banda filtrada e do somatório dos *Nr* do histograma original.

$Z[0]$ e $nr[0]$ vão conter os dados relativos ao pico, tal como em na função de processamento de apenas picos e os restantes valores $Z[1]$, $Z[2]$, $nr[1]$ e $nr[2]$, vão conter os da banda tal como na função de processamento de apenas 1 faixa.

Mais uma vez estes parâmetros não serão utilizados para identificação: o pico não corresponde à totalidade da superfície do objecto sobreposto mas apenas à parte visível e a banda refere-se a um objecto rodado no espaço.

Eles serão contudo utilizados, como já foi referido atrás, no cálculo de valores úteis à manipulação do objecto em sobreposição para a sua remoção.

4. Não detecção de picos ou faixa

Esta função é chamada quando não são detectados picos ou banda, ou seja, quando a forma do objecto não se coaduna com o pressuposto da existência de superfícies planas paralelas ao plano horizontal ou rodadas no espaço. Pode ainda corresponder à situação de existir um número excessivo de formas no histograma: p.ex. dois picos com uma banda que a limitação de 3 elementos de entrada para $Z[]$ e $nr[]$ não pode contemplar.

Neste caso, aqueles vectores são postos a zero e a variável de estado toma o valor "unknown".

No Anexo H.1 é mostrado esquema de chamadas de todas estas funções.

Ainda, no Anexo H.2 é apresentado um fluxograma detalhado do programa de medição de "Assinaturas".

10.2.4 Resumo da Codificação da Informação nos 3 casos: Ilustração Gráfica e Resultados

Situação 1: Sobreposição Simples de Poliedros

Se o histograma contiver apenas picos, então a codificação é óbvia: $Z[i]$ conterà a altura associada ao pico i , enquanto que $nr[i]$ será o número de pixels a ele associado, ou seja, a área. Se, pelo contrário, houver rotação no espaço e aparecer uma faixa, então como ficou visto, aqueles dois vectores são utilizados para registar parâmetros que serão úteis para a manipulação robótica do objecto visível.

Mas nesta 1ª situação o registo a efectuar (Fig.6) é o já indicado acima.

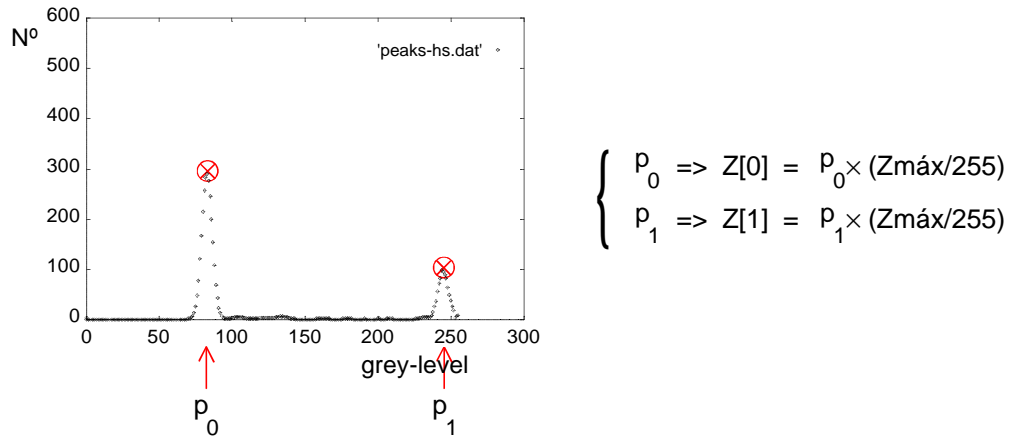


Fig. 6 - Parametrização do histograma 3D e registo na "Assinatura" dos valores referentes a 2 picos.

$Z[0]$ = valor de z correspondente ao máximo do pico, em mm: $p_0 \times (Zmáx/255)$
 $Z[1] = p_1 \times (Zmáx/255)$
 $Z[2] = 0$

$nr[0]$ = área do 1º pico = $\sum_{Z[0]-\Delta Z}^{Z[0]+\Delta Z} nr(z)$

$nr[1]$ = área do 2º pico
 $nr[2] = 0$
 $nr_peaks = 2$
 $name =$ (não é preenchido!)

A execução do programa num caso concreto deu o seguinte resultado:

```

1 Range Blob

----- Blob 0 -----
  jmin=216
  imin=275  imax=353
  jmax=324
  area=3923
  x=315.2, y=268.7
----- Signature -----
  sumz2 = 9488312 (mm2)
  sumNr2 = 15894 (pixel2)
  nr_peaks = 2
  ----- peak 0 -----
  Z[0]= 27, nr[0]=2667
  ----- peak 1 -----
  Z[1]= 83, nr[1]=914
  -----
    
```

Situação 2: Oclusão Total

No caso de ocorrer uma oclusão total, apenas uma faixa aparecerá no histograma 3D e a informação a ser registada será a apresentada na Fig.7.

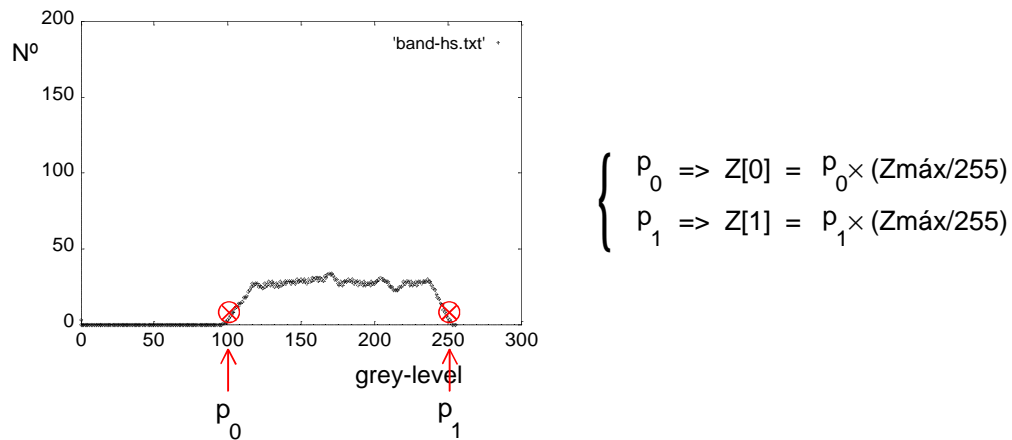


Fig. 7 - Parametrização do histograma 3D e registo na "Assinatura" dos valores referentes a 1 faixa.

$Z[0] = 1^{\circ}$ valor de z em mm

$Z[1] = 2^{\circ}$ valor de z em mm

$Z[2] = 0$

$nr[0] = nr$ médio entre p_1 e p_2

$nr[1] =$ valor da projecção da área na horizontal (integral da faixa entre p_1 e p_2)

$nr[2] = 0$

$nr_peaks = 0$

$name = "rotated"$

Os valores de Z são obtidos por conversão dos valores "grey-level" p_1 e p_2 :

$$Z[0] = p_1 \times (Z_{máx}/255)$$

$$Z[1] = p_2 \times (Z_{máx}/255)$$

A execução do programa num outro caso concreto deu o seguinte resultado:

1 Range Blob


```

----- Blob 0 -----
      jmin=206
imin=304   imax=385
      jmax=302
      area=3976
      x=344.6, y=253.3
----- signature -----
sumz2 = 8701417 (mm2)
sumNr2 = 37150 (pixel2)
nr_peaks = 0
----- band -----
Z[0] = 28; nr[0] = 28
Z[1] = 63; nr[1] = 3802
-----
    
```

Situação 3: Oclusão Parcial

Se a oclusão for apenas parcial, então aparecerão no histograma 3D tanto o pico associado à parte visível da superfície plana do objecto parcialmente sobreposto como a faixa (ou banda) associada à superfície rodada no espaço. Nenhuma daquelas figuras permitirá o reconhecimento dos objectos envolvidos: no 2º caso porque a faixa não é transformável no pico que apareceria se o objecto não estivesse rodado no espaço; no 1º, porque o pico não traduz a superfície completa do objecto sobreposto e por conseguinte os parâmetros extraídos dele não descreveriam correctamente essa superfície. O registo de alguns parâmetros na estrutura de valores da "Assinatura" permitirá contudo obter uma sub-segmentação para posterior separação física dos dois objectos envolvidos na oclusão parcial.

A informação a registar será então a do pico seguida da codificação apresentada na secção anterior (Fig.8).

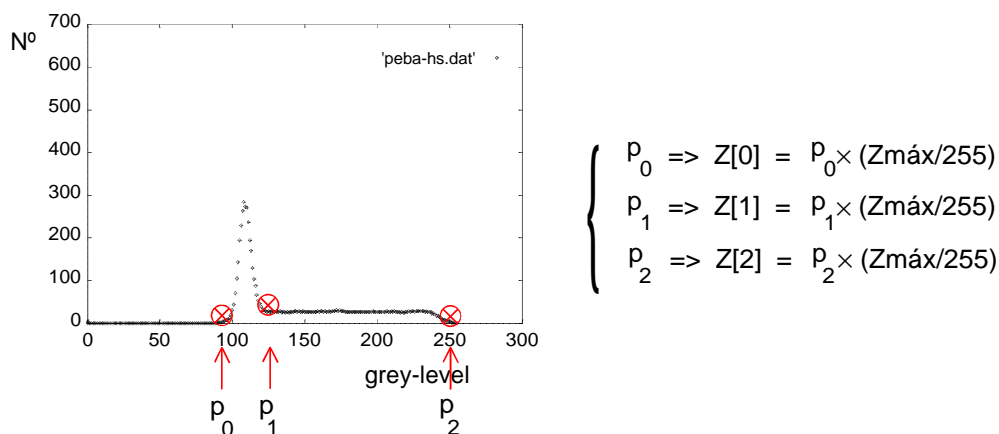


Fig. 8 - Parametrização do histograma 3D e registo na "Assinatura" dos valores referentes a 1 pico + 1 faixa.

$$Z[0] = p_0 \times (Z_{\text{máx}}/255)$$

$$Z[1] = p_1 \times (Z_{\text{máx}}/255)$$

$$Z[2] = p_2 \times (Z_{\text{máx}}/255)$$

$$nr[0] = \text{área do pico} = \sum_{z[0]-\Delta Z}^{z[0]+\Delta Z} nr(z)$$

$nr[1]$ = valor médio de 'Nr' da faixa suavizada entre p_1 e p_2

$nr[2]$ = valor da projecção da área na horizontal (integral da faixa entre p_1 e p_2)
(a área real da superfície poderá ser obtida pela razão entre esta área e o cosseno do ângulo de rotação no espaço)

$nr_peaks = 1$

$name = "rotated"$

Mais uma vez, para um caso concreto, obteve-se o seguinte resultado:

```

1 Range Blob

----- Blob 0 -----
      jmin=206
imin=272  imax=342
      jmax=352
      area=6596
      x=327.4, y=277.5
----- signature -----
sumz2 = 1809433 (mm2)
sumNr2 = 28046 (pixel2)
nr_peaks = 1
----- peak 0 -----
Z[0] = 26; nr[0] = 2627
----- band -----
Z[1] = 30; nr[1] = 27
Z[2] = 59; nr[2] = 3219
-----

```

10.2.5 A Filtragem dos Histogramas:

A Nova Abordagem e o Processo de Filtragem Associado

A versão anterior, embora processasse apenas picos, já requeria uma filtragem prévia do histograma devido ao carácter descontínuo de registo de altura em apenas 256 níveis de cinzento, e também do pixel na imagem a partir do qual se avalia a altura associada, resulta sempre uma dispersão e oscilação de valores nos histogramas de altura directos. Houve então que filtrar convenientemente estes histogramas por forma a suavizá-los não só para evitar a detecção errónea de falsos picos como eliminar aquela oscilação.

Os filtros anteriormente utilizados, *smooth_hist()* com dimensão variável de máscara, e *quick_smooth()* que utiliza uma máscara linear $\{1,2,1\}$, já permitiam essa suavização (os picos apareciam com a forma conveniente da curva de Gauss). Desse modo os Zs ficavam melhor definidos. A avaliação de *nr[]*, ou seja da área real da superfície, era contudo efectuada com o histograma original (Fig.9), embora o histograma filtrado fosse usado auxiliarmente para definir as fronteiras dos picos (definição do início e fim dos picos).

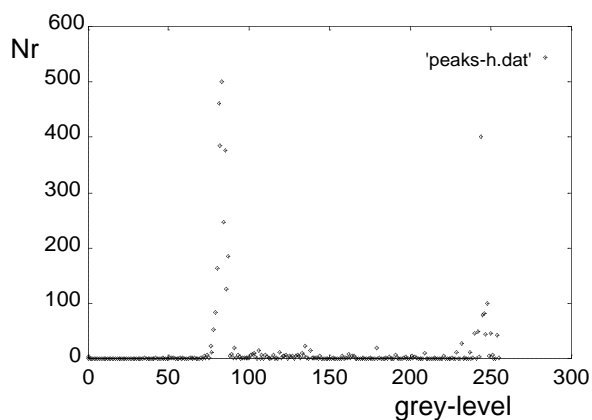


Fig. 9 - Histograma 3D não filtrado (2 picos).

O aparecimento de bandas veio trazer novas exigências aos filtros, uma vez que o carácter descontínuo dos pixels de imagem, que já provocava efeitos de dispersão nos picos, tornava aquelas figuras inutilizáveis para extracção de parâmetros, tal era a dispersão de pontos, como se mostra na Fig.10.

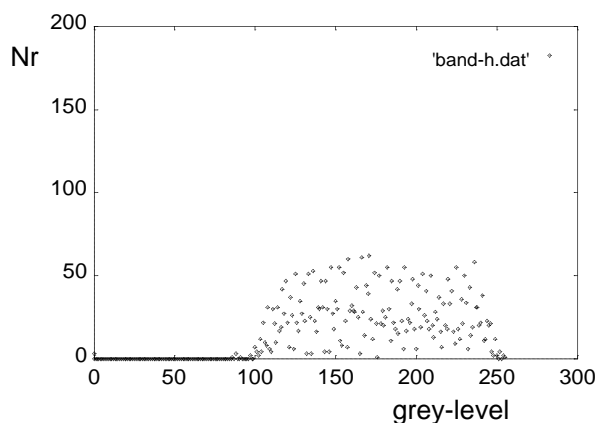


Fig. 10 - Histograma 3D não filtrado (1 faixa).

Houve então que procurar um filtro eficiente para todas as situações e daí a construção de um programa interactivo mais ou menos complexo que permitia o uso de vários filtros à escolha do utilizador e o "display" do resultado de cada aplicação, sem

contudo corromper o histograma original, permitindo assim experimentar sucessivamente vários filtros. Na sequência dos testes efectuados (10.2.5.2) veio a adoptar-se um deles como o mais eficiente para os de menor dimensão de máscara que envolve também funções de suavização das 2 bordas. Após a aplicação deste filtro num histograma de faixa como o mostrado na figura anterior, aparece já uma forma suavizada que permite a detecção dos extremos da faixa (embora se mantenha o histograma original para a contagem de $nr[]$ - medição da projecção XOY da área visível (Fig.11).

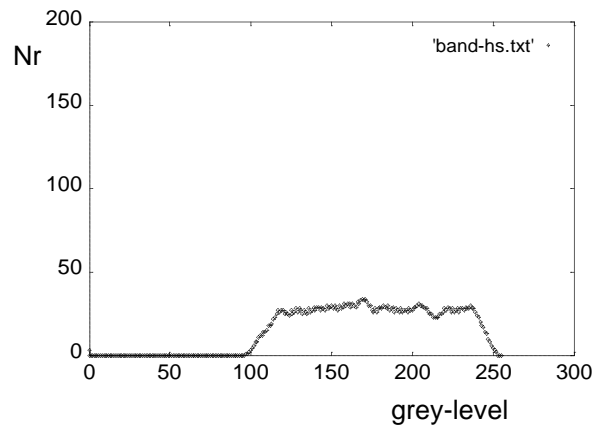


Fig. 11 - Histograma suavizado referente a 1 faixa.

No caso de ocorrer simultaneamente 1 pico e 1 faixa é o seguinte o resultado da filtragem com o referido filtro (Fig.12):

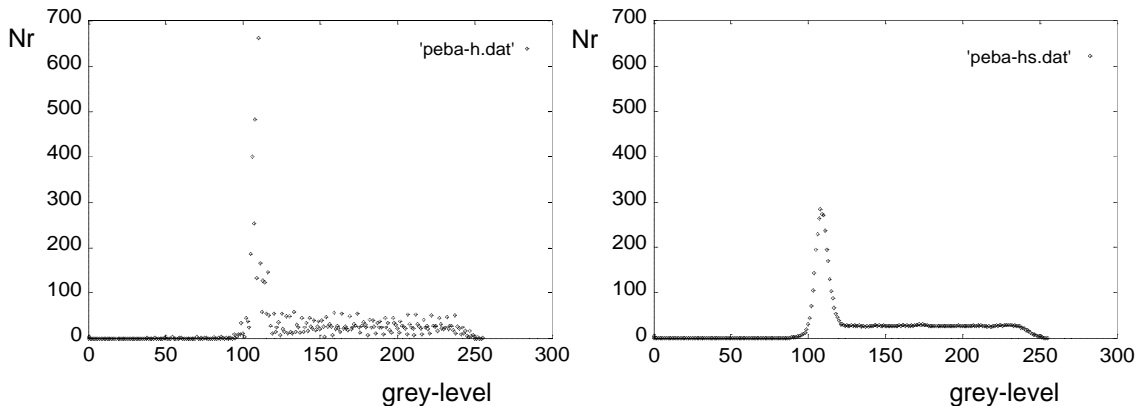


Fig. 12 - Histograma 3D referente a 1 pico + 1 faixa, não filtrado (à esquerda) e filtrado (à direita).

10.2.5.1 As Primeiras Versões do Filtro de Histogramas 3D

1. suavização do histograma

Neste filtro considera-se uma máscara linear de dimensão n variável (a escolher de início pelo utilizador) e preenchida apenas com 1's. O vector que contém o histograma mantém os valores das extremidades e vai fornecendo valores a um vector de dimensão igual à da dimensão escolhida à medida que se vão percorrendo os níveis de cinzento em ordem crescente. Uma vez que os novos valores obtidos para o histograma (resultantes da média aritmética do conteúdo daquele vector) não podem ser utilizados para novos cálculos (a convolução utiliza apenas os valores originais), apenas o 1º valor daquele vector é registado no histograma, imediatamente antes do vector considerar os n valores seguintes: é efectuado um deslocamento de todos os valores contidos no vector no sentido decrescente, sendo o 1º, já utilizado para registo, ignorado, e o elemento $n-1$ lido do histograma. Todo este procedimento é repetido até que o elemento de entrada $n-1$ do vector coincida com 255. Os últimos valores do histograma em nº de $(n-1)/2$, tais como os primeiros e em igual número, não são sujeitos a alterações.

2. suavização rápida

Este filtro veio substituir o anterior pela sua rapidez de execução. Embora seja limitado na escolha da dimensão da máscara (utiliza uma máscara fixa de 3×1 : {1,2,1}), dispensa as inúmeras operações de deslocamento no vector auxiliar: existe um segundo histograma onde o resultado das convoluções vai sendo registado. A operação de convolução é efectuada 6 vezes sendo os apontadores para os dois histogramas trocados entre cada duas delas (o que implica que o nº total deva ser sempre par para que o histograma original venha no fim a conter o resultado final). O histograma auxiliar é libertado da memória e não é retornado qualquer valor uma vez que como foi dito o histograma filtrado é sobreposto ao original.

10.2.5.2 O Programa Interactivo para Determinação do Melhor Filtro de Histogramas 3D

Por forma a poder-se determinar a melhor máscara de convolução e o nº de iterações ideal, foi desenvolvido um programa específico.

Em primeiro lugar ele pede ao utilizador que indique as coordenadas delimitadoras do rectângulo que confina a figura a analisar. Desta forma dispensa-se a segmentação da imagem que deve preceder toda a medição de "Assinaturas". A escolha dessas coordenadas é efectuada com a ajuda de um outro programa interactivo que permite ao utilizador mover um ponto brilhante no ecrã do monitor. Os pontos a escolher devem ser o canto superior esquerdo e o canto inferior direito que definem por completo aquele rectângulo. Quando qualquer daqueles pontos é atingido, o utilizador dá a indicação da sua escolha de que resulta a memorização dos valores do par (x,y)

definido, em coordenadas da imagem. Após a escolha do 2º par de valores é desenhado no ecrã o rectângulo delimitador após o que o programa retorna os dois pares de valores para uso no programa de segmentação.

Como seria de esperar aquele programa só detecta uma figura, mas a sua execução apenas para a área da imagem delimitada por aquele rectângulo por um lado acelera o processo de medida das características geométricas e da construção do histograma 3D e por outro permite efectuar a análise de apenas uma dada figura pretendida quando a imagem apresenta várias. Aquele histograma, que é a base de toda a análise de "Assinaturas", consiste, como seria de esperar, no nº total encontrado de pixels para cada intervalo estreito de valores de altura, esta codificado em níveis de cinzento entre 0 e 255, e é registado como um vector de 256 inteiros. Cada posição do vector contém então o nº total de pixels encontrados para o nível correspondente de cinzento i , com $i \in [0,255]$ em que o valor real de Z que lhe corresponde pode ser facilmente calculado sabendo $Z_{máx}$:

$$Z = \frac{Z_{máx}}{255} \times i$$

Posteriormente é feito o "display" em monitor do histograma 3D calculado, usando para o efeito apenas as primeiras 128 linhas por forma a não se sobrepor à imagem de alturas.

Partindo então desse histograma inicial, o utilizador é convidado a experimentar a acção provocada por um de vários filtros à disposição. Esses filtros são obviamente lineares e correspondem à convolução de todo o histograma com uma das seguintes máscaras:

máscara com dimensões 3×1:	{1,2,1}
máscara com dimensões 5×1:	{1,1,2,1,1}
máscara com dimensões 7×1:	{1,1,1,2,1,1,1}
máscara com dimensões 9×1:	{1,1,1,1,2,1,1,1,1}

É permitido ao utilizador aplicar qualquer uma daquelas máscaras, por qualquer ordem, e qualquer nº de vezes; e sempre que uma delas é aplicada, o novo histograma 3D filtrado é imediatamente mostrado no ecrã. Qualquer filtragem julgada inadequada pode ser eliminada e o anterior histograma recuperado uma vez que ele é sempre mantido em memória. Do mesmo modo é mantido em memória o histograma 3D inicial para permitir reiniciar o processo e assim experimentar uma outra sequência de filtrações. Esse histograma pode ser chamado em qualquer altura da experiência. Deste modo o utilizador pode executar tantas experiências quantas as pretendidas, todas a partir do mesmo histograma 3D inicial, para assim escolher o melhor filtro a ser aplicado assim como o nº de iterações.

Apresenta-se de seguida o resultado de algumas experiências efectuadas com este programa.

Experiência 1: 2 picos

a) Escolha do rectângulo delimitador e execução de testes de filtragem:

t-29.txt:

```

cursor step = 64

0 0
64 0
(...)
384 64
384 128
lower step selected: 16
384 128
(...)
416 176
top left corner = (416,176)

set 1 : x=416, y=176

416 176
(...)
512 304
528 304
lower step selected: 4
528 304
528 308
bottom right corner = (528,308)

set 2 : x=528, y=308
    
```

```

rectangle defined!

Computing histogram from range
image ...
Please observe hist3D while beeing
smoothed! (Esc to end)

('3' for mask3, '5' for mask5, '7'
for mask7, 'p' for previous, 'i' for initial)

3x1 mask selected!
Initial hist3D selected!
5x1 mask selected!
Initial hist3D selected!
7x1 mask selected!
Previous hist3D selected!
9x1 mask selected!
Previous hist3D selected!
7x1 mask selected!

No more smoothings!

store smoothed hist3D to file? (y/n) y
Filename for formatted output - h7.dat
Clean hist3D on monitor...
    
```

O registo em ficheiro do histograma filtrado permitirá obter a sua representação gráfica.

b) Conteúdo do histograma filtrado:

grey-level N°

0	11366	78	114	92	3	245	88
1	0	79	157	93	1	246	100
2	0	80	201	94	0	247	109
(...)		81	242	95	0	248	101
69	0	82	258	(...)		249	89
70	0	83	260	236	0	250	73
71	0	84	243	237	0	251	51
72	1	85	205	238	2	252	21
73	4	86	161	239	4	253	12
74	8	87	116	240	9	254	5
75	22	88	75	241	20	255	0
76	42	89	43	242	33		
77	70	90	22	243	49		
		91	9	244	72		

c) Para comparação observe-se o conjunto completo de histogramas para os filtros 3×1 , 5×1 e 7×1 (Fig.13):

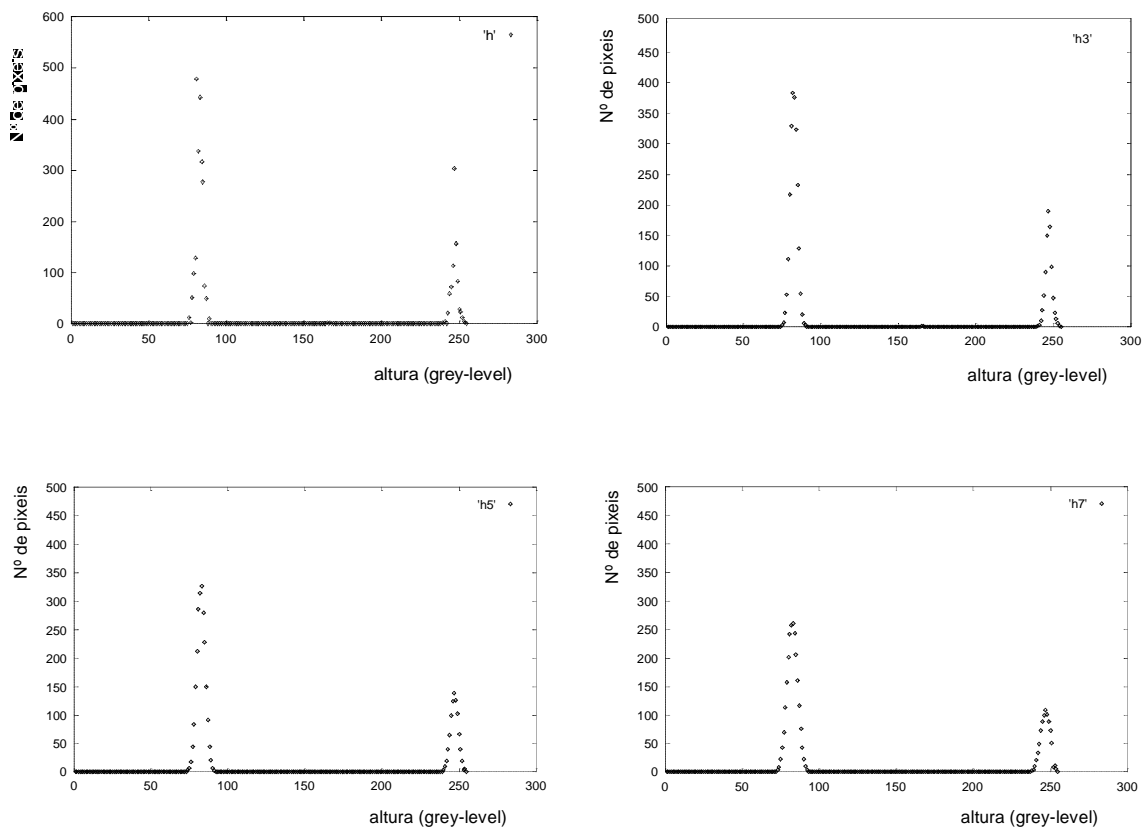


Fig. 13 - Ilustração comparativa do resultado de 3 filtragens diferentes para 2 picos.

Por análise de todos os resultados apresentados atrás, pode-se verificar que a máscara de dimensão 7×1 aplicada uma única vez é perfeitamente satisfatória pelo que se adopta a função correspondente para a filtragem dos histogramas 3D.

Experiência 2: 1 faixa

a) Escolha do rectângulo delimitador e execução de testes de filtragem:

't-29.txt':

```

cursor step = 64

0 0
64 0
(...)
256 128
256 192
lower step selected: 16
256 192
272 192
top left corner = (272,192)

set 1 : x=272, y=192

272 192
higher step selected: 64
272 192
(...)
400 320
bottom right corner = (400,320)

set 2 : x=400, y=320

rectangle defined!
Computing histogram from range
    
```

```

image ...
Please observe hist3D while beeing
smoothed! (Esc to end)

('3' for mask3, '5' for mask5, '7' for mask7,
'p' for previous, 'i' for initial)

3x1 mask selected!
Initial hist3D selected!
5x1 mask selected!
Previous hist3D selected!
7x1 mask selected!
Previous hist3D selected!
9x1 mask selected!
Initial hist3D selected!
7x1 mask selected!

No more smoothings!

store smoothed hist3D to file? (y/n) y
Filename for formatted output -- h7.dat
Clean hist3D on monitor...
    
```

O registo em ficheiro do histograma filtrado permitirá obter a sua representação gráfica.

b) Conteúdo do histograma filtrado:

grey-level N°

0	20281	106	12	123	25	243	18
1	0	107	12	124	24	244	17
2	0	108	14	125	27	245	14
(...)		109	14	(...)		246	13
95	0	110	15	230	28	247	10
96	0	111	18	231	28	248	8
97	1	112	18	232	27	249	6
98	1	113	19	233	29	250	4
99	2	114	22	234	28	251	3
100	3	115	23	235	28	252	2
101	4	116	25	236	30	253	1
102	6	117	27	237	28	254	1
103	8	118	26	238	28	255	1
104	8	119	27	239	26		
105	11	120	27	240	24		
		121	25	241	23		
		122	26	242	20		

c) Para comparação observe-se o conjunto completo de histogramas para os filtros 3×1, 5×1, 7×1 e 9×1 (Fig.14):

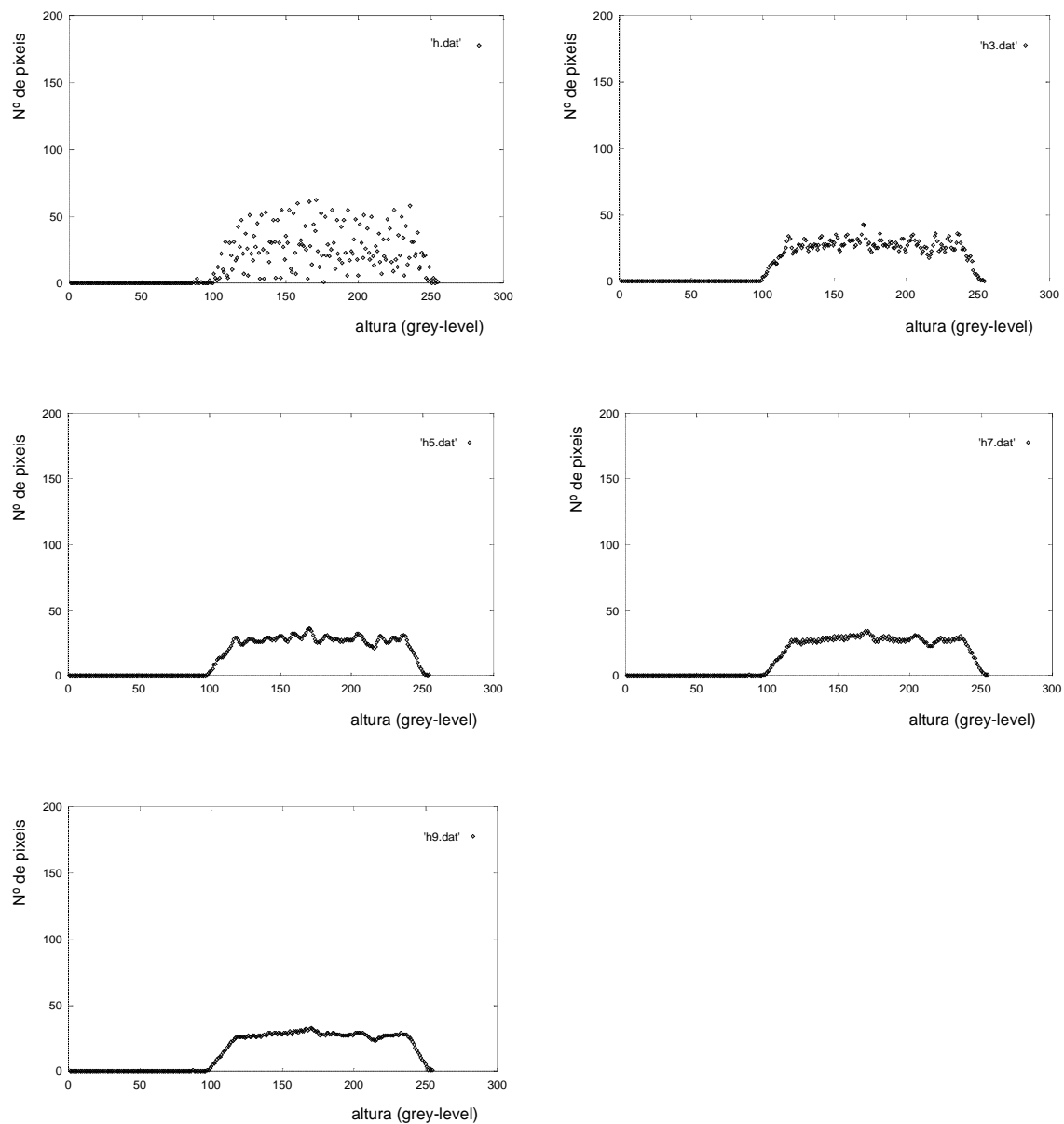


Fig. 14 - Ilustração comparativa do resultado de 4 filtragens diferentes para 1 faixa.

Pode-se verificar mais uma vez que a máscara de dimensão 7×1 é perfeitamente satisfatória pelo que se confirma a sua adoção para a filtragem dos histogramas 3D.

Experiência 3: pico+faixa

a) Escolha do rectângulo delimitador e execução de testes de filtragem:

t-29.txt:

```

cursor step = 64

0 0
64 0
(...)
256 128
256 192
top left corner = (256,192)

set 1 : x=256, y=192

256 192
(...)
384 384
lower step selected: 16
384 384
400 384
bottom right corner = (400,384)

set 2 : x=400, y=384

rectangle defined!
Computing histogram from range
    
```

```

image ...
Please observe hist3D while beeing
smoothed! (Esc to end)

('3' for mask3, '5' for mask5, '7' for mask7,
'p' for previous, 'i' for initial)

3x1 mask selected!
Initial hist3D selected!
5x1 mask selected!
Previous hist3D selected!
7x1 mask selected!
Previous hist3D selected!
9x1 mask selected!
Initial hist3D selected!
7x1 mask selected!

No more smoothings!

store smoothed hist3D to file? (y/n) y
Filename for formatted output -- h7.dat
Clean hist3D on monitor...
    
```

O registo em ficheiro do histograma filtrado permitirá obter a sua representação gráfica .

b) Conteúdo do histograma filtrado:

grey-level N°

0	20709	92	2	113	170	237	26
1	0	93	3	114	129	238	23
2	0	94	4	115	102	239	22
3	0	95	5	116	87	240	20
4	0	96	8	117	66	241	17
5	0	97	10	118	53	242	16
6	0	98	12	119	46	243	14
7	0	99	19	120	38	244	12
8	0	100	30	121	31	245	10
9	0	101	44	122	30	246	8
10	0	102	71	123	28	247	7
11	0	103	104	124	26	248	6
12	0	104	143	125	28	249	4
(...)		105	194	(...)		250	3
87	0	106	228	230	28	251	2
88	0	107	264	231	28	252	1
89	1	108	284	232	28	253	0
90	1	109	273	233	27	254	0
91	1	110	270	234	28	255	0
		111	236	235	27		
		112	194	236	26		

d) Para comparação observe-se o conjunto completo de histogramas para os filtros 3×1 , 5×1 , 7×1 e 9×1 (Fig.15):

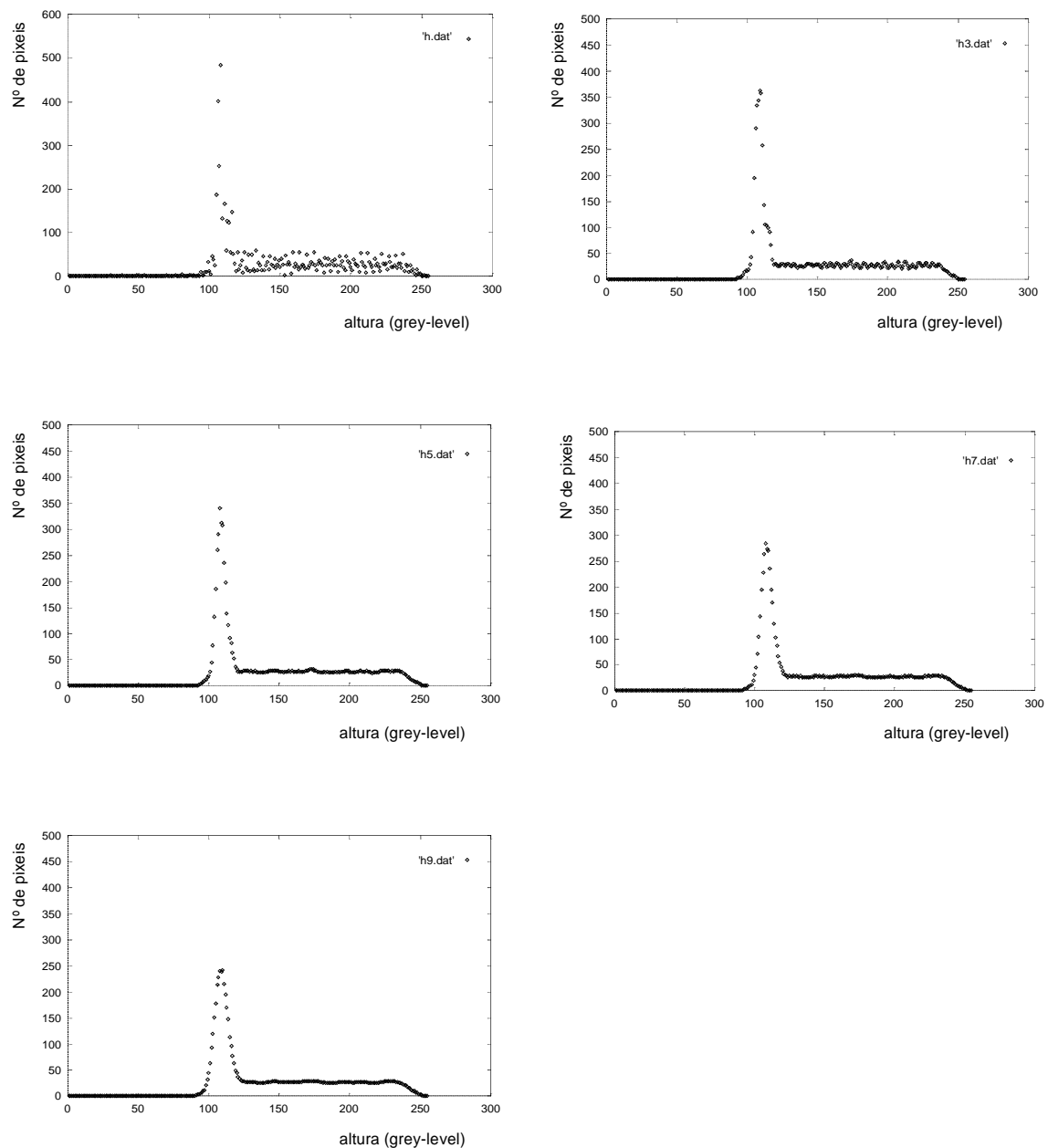


Fig. 15 - Ilustração comparativa do resultado de 4 filtragens diferentes para 1 pico+faixa.

Pode-se verificar mais uma vez que a máscara de dimensão 7×1 é perfeitamente satisfatória pelo que se confirma a sua adoção para a filtragem dos histogramas 3D.

10.2.6 Outros Programas Auxiliares: A Extracção de Valores para a Representação Bi e Tridimensional

A necessidade de ilustrar a forma dos objectos, os estados de sobreposição e ainda a imagem das alturas tal como a mostrava o monitor, levou ao desenvolvimento de 3 programas de extracção de valores. Qualquer um deles oferece a possibilidade de registar em ficheiro quer os valores de altura em mm lidos de forma sequencial, quer os valores (x,y) em coordenadas da imagem dos pixeis pertencentes à figura ou figuras seleccionadas. Ao contrário do que acontece com o ficheiro dos Zs que requer o mesmo nº de valores por linha (sendo cada conjunto separado por uma linha em branco), o das coordenadas dos pixeis das figuras dispensam o uso de todos os valores: apenas aqueles que definem as figuras no monitor devem constar da lista a registar.

Estes programas podem ser aplicados quer à imagem de alturas quer ao ficheiro de amostragem de alturas em qualquer resolução, mostrado ou não em monitor.

Embora todos os 3 programas desenvolvidos tenham o mesmo objectivo, atrás apresentado, possuem contudo variantes que interessa explicitar:

1. 1ª versão:

Este programa lê directamente no ficheiro de imagem da placa de aquisição (a partir do endereço 0xD000), ou seja, acede ao valor dos pixeis mostrados em monitor através de um varrimento para valores crescentes de x e de y (da esquerda para a direita e de cima para baixo).

Ele aceita um intervalo de valores de Z , e quando do registo em ficheiro, põe a zero todas as alturas fora desse intervalo; o objectivo é permitir a representação tridimensional de apenas um dos objectos em sobreposição, o sobreposto ou aquele em sobreposição (eles não podem ser ignorados uma vez que a aplicação gráfica exige a constância no nº de pixeis por linha); quando do registo do ficheiro com valores de (x,y) , todos os pixeis que possuam Zs fora do intervalo indicado são ignorados e portanto não aparecerão numa representação bidimensional efectuada pela aplicação gráfica.

O programa permite também isolar uma determinada figura da imagem pela aceitação de limites quer em x quer em y .

Este programa é perfeitamente amigável, permitindo extrair da imagem de alturas com facilidade, qualquer um dos dois ficheiros (ou ambos) destinados a ser lidos por uma aplicação gráfica. Apenas a representação gráfica da amostragem de alturas (e não da imagem de alturas que dela se obtém por filtragem quando ela possui a resolução máxima), com qualquer resolução, requer um pequeno truque: ela deve ilustrar as variações locais de densidade na medida dos Zs; ora, este programa, que permite a selecção do incremento na leitura dos pixeis, só as poderá mostrar se a imagem contida na memória da placa for a do ficheiro de amostragem de alturas e não a das alturas (onde todos os pixeis dentro dos limites das figuras possuem um valor de Z associado); nesta circunstância deve-se ordenar a leitura dos Zs com o incremento igual à unidade para que todos os pixeis que contenham valores definidos para Z possam ser lidos. Deve-se ainda registar em $(0,128)$ e $(1,128)$ o valor de $Z_{máx}$ como se de uma imagem de alturas se tratasse, pois o registo do ficheiro com valores de altura Z (acompanhados das respectivas coordenadas x e y) irá necessitar dele para converter níveis de cinzento em

valores reais de altura em mm. Quase todas as representações bidimensionais apresentadas atrás assim como a da Fig.17 foram obtidas deste modo.

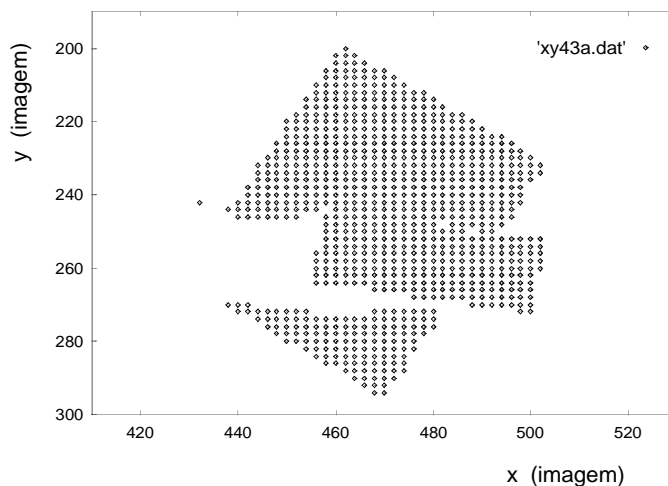


Fig. 16 - Representação bidimensional de uma imagem de alturas.

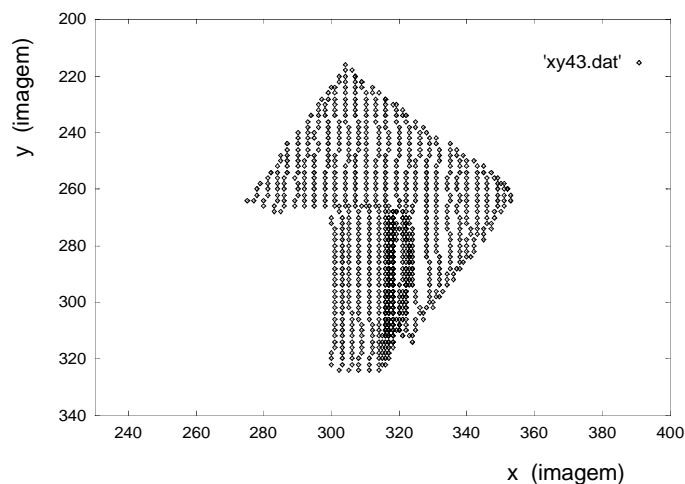


Fig. 17 - Representação bidimensional de uma amostragem de alturas na resolução máxima.

2. 2ª versão

Este programa é idêntico ao anterior, distinguindo-se apenas no tipo de varrimento de imagem: ele é efectuado sempre para valores de y crescentes e x decrescentes, ou seja, de cima para baixo e da direita para a esquerda.

O objectivo desta variação é poder obter representações tridimensionais noutra perspectiva eventualmente mais favorável à visualização de uma dada sobreposição (Fig.18).

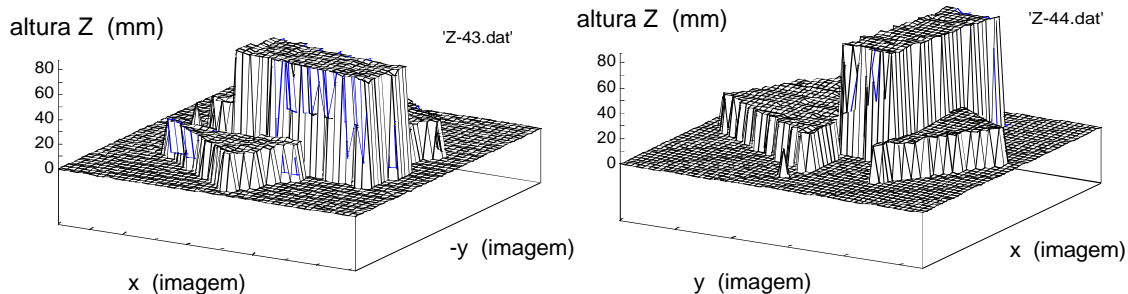


Fig. 18 - Representação tridimensional em duas perspectivas de uma situação de sobreposição.

3. 3ª versão

Nesta versão não é efectuada nenhum varrimento da imagem em monitor uma vez que se destina ao uso do ficheiro de amostragens e não da imagem de alturas. Ela utiliza portanto aquele 1º ficheiro, directamente lido do disco (ficheiro de amostragem de alturas em modo texto) dispensando tanto a leitura para a placa de aquisição de imagens (com a consequente mostra em monitor) como o truque usado nas versões anteriores para obter a "irregularidade" dos pixels na representação bidimensional.

Mantém, no entanto, a liberdade de escolha dos ficheiros a registar assim como a definição de um intervalo de valores de Z a considerar, pondo a zero todas as outras alturas no ficheiro dos Zs e ignorando os correspondentes pares (x,y) no registo bidimensional (Fig.19).

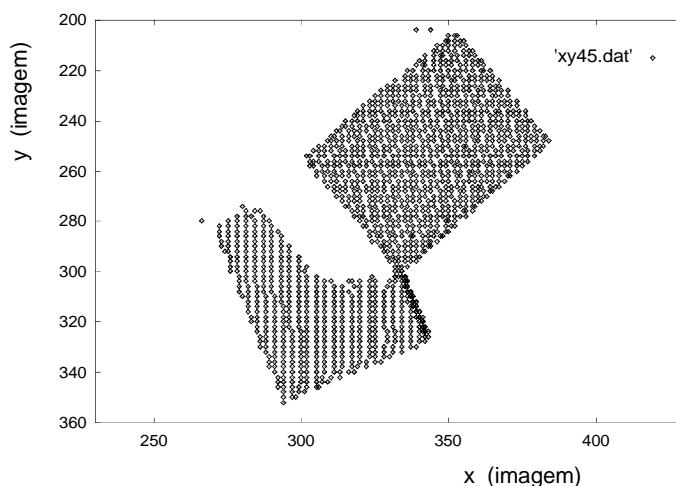


Fig. 19 - Representação bidimensional de uma amostragem de alturas na resolução máxima.

10.3 A Fiabilidade dos Parâmetros Identificadores

O uso bem sucedido dos parâmetros identificadores no processo de reconhecimento a 3D depende essencialmente da sua fiabilidade. Para um mesmo

objecto, aqueles parâmetros não devem variar muito perante alterações no seu posicionamento ou orientação. Sendo assim, há que avaliar *a priori* essa variação por forma a poder atribuir um dado grau de certeza numa qualquer identificação que os utilize. Foram então efectuadas 3 experiências distintas, não necessariamente para o mesmo objecto, nas quais se efectuou alteração no seu posicionamento:

- Translação de um poliedro simples
- Rotação de um poliedro simples
- Posicionamento aleatório de um poliedro simples

Os resultados dão uma medida da fiabilidade de cada um dos parâmetros utilizados que deverão ser comparados posteriormente com os erros percentuais obtidos em identificações reais para confirmação do valores encontrados.

As experiências efectuadas são apresentadas a seguir.

10.3.1 Translação com um Poliedro Simples

Foi colocado um mesmo objecto em várias posições do espaço de acção do robot (e visíveis pela câmara). Para cada uma delas foi ordenado o varrimento do sistema laser com a resolução máxima ($res=1$) após o que foi adquirida a correspondente imagem de amostragens. Após filtragem de cada uma dessas imagens com a sucessão de filtros de 9.10.1, obteve-se finalmente a imagem de alturas. A análise do seu histograma devidamente suavizado (10.2.5), permitiu a obtenção dos parâmetros a utilizar no processo de identificação a 3D.

Os valores obtidos em função da posição do objecto foram os mostrados na Tabela 1.

Tabela 1

Posição	$sumZ2$	$sumNr2$	$Z[0]$	$nr[0]$
Cima-Esquerda	2819332	47854	26	3728
Cima-Centro	3087051	51717	26	3819
Cima-Direita	2832209	59773	27	3801
Baixo-Esquerda	2634679	32109	26	3674
Baixo-Centro	3257611	14925	28	3787
Baixo-Direito	3061734	62448	28	3800
Centro-Esquerda	2986184	4383	27	3797
Centro-Direita	3000108	51320	28	3833

< $sumZ2$ >	< $sumNr2$ >	< $Z[0]$ >	< $nr[0]$ >
2959863	40566	27	3780

$\sigma :$	$\sigma :$	$\sigma :$	$\sigma :$
180038	19934	0,8	49
% :	% :	% :	% :
6	49	3	1,3

Como se pode observar, com excepção do 2º parâmetro, obtiveram-se baixos valores percentuais de desvio padrão.

10.3.2 Rotação com um Poliedro Simples

Do mesmo modo que anteriormente, foram feitas várias experiências de aquisição de imagens de altura após o que se extraíram os parâmetros do histogramas 3D a elas associados. Neste caso foi escolhida uma posição fixa e rodado o objecto.

A Tabela 2 apresenta os resultados obtidos.

Tabela 2

Posição	<i>sumZ2</i>	<i>sumNr2</i>	<i>Z[0]</i>	<i>nr[0]</i>
0°	2669276	42830	26	3604
45°	2929276	18296	26	3812
90°	2984850	59483	26	3797
135°	3101050	1781	28	3852
180°	3019241	45506	27	3852
225°	3124124	19285	27	3813
270°	3042857	26526	28	3765
315°	3063983	43188	27	3740

< <i>sumZ2</i> >	< <i>sumNr2</i> >	< <i>Z[0]</i> >	< <i>nr[0]</i> >
2991832	32112	26,9	3779
$\sigma :$	$\sigma :$	$\sigma :$	$\sigma :$
135090	17581	0,78	76
% :	% :	% :	% :
4,5	55	2,9	2

Uma vez mais foram obtidos baixos valores percentuais para o desvio padrão, com excepção do 2º parâmetro.

10.3.3 Poliedro Simples em Posições Aleatórias

Neste 3º estudo fez-se variar tanto a posição como a orientação do objecto. Os parâmetros obtidos após análise dos histogramas 3D das respectivas imagens de altura, são mostrados na Tabela 3.

Tabela 3

Posição	<i>sumZ2</i>	<i>sumNr2</i>	<i>Z[0]</i>	<i>nr[0]</i>
1	3772404	23693	29	4171
2	3475627	49747	28	4087
3	3264151	36579	27	4088
4	3426883	4027	28	4174
5	3180373	12775	29	3580
6	3565676	62515	29	3986
7	3580668	14296	29	4006
8	3317346	42011	27	4184

< <i>sumZ2</i> >	< <i>sumNr2</i> >	< <i>Z[0]</i> >	< <i>nr[0]</i> >
3447891	30705	28,25	4034
σ :	σ :	σ :	σ :
180413	19001	0,83	186
% :	% :	% :	% :
5,2	62	2,9	4,6

Os valores obtidos são bastante próximos dos anteriores.

Os valores que majoram os desvios padrões associados a cada um dos parâmetros são apresentados na Tabela 4.

Tabela 4

Valores Majorantes			
% (<i>sumZ2</i>)	% (<i>sumNr2</i>)	% (<i>Z[0]</i>)	% (<i>nr[0]</i>)
6	62	3	5

Da análise destes resultados pode-se concluir que afinal o parâmetro *sumNr2* referente ao momento de 2ª ordem do histograma de alturas filtrado, não é adequado para identificação de objectos dada a grande dispersão dos seus valores.

Pelo contrário, os restantes 3 parâmetros apresentam dispersões bastante pequenas pelo que se adequam perfeitamente à tarefa de identificação de objectos. Sendo assim, serão utilizados apenas estes no algoritmo correspondente.

10.4 O Ficheiro de Modelos 3D

Para que uma identificação seja possível, tal como já acontecia nos algoritmos 2D, é necessário que exista um registo contendo todos os modelos que poderão estar presentes numa determinada acção de manipulação. Para além das respectivas identidades, este ficheiro deve conter toda a informação utilizada em reconhecimento e que no caso 3D, consiste no que se convencionou chamar "Assinaturas" do objecto, que envolve a altura máxima *Z* e o número de pixeis de distância detectados (área). Uma vez que a situação mais vulgar para qualquer objecto é o do repouso numa das posições estáveis, este ficheiro conterá tantas "Assinaturas" quanto o número de estados estáveis possíveis. Por exemplo, um paralelepípedo terá 3 estados estáveis, enquanto que um cubo ou uma esfera só poderão ter 1 único estado estável. Mesmo em situações de sobreposição parcial é esperado que o objecto que se encontra no topo se mantenha num estado estável para uma boa identificação. Caso não se verifique esta condição é accionada uma manipulação de recurso de forma a ultrapassar o problema como se verá mais adiante.

A utilização de "Assinaturas" para identificação restringe obviamente o tipo de objectos que podem ser identificados. Mais concretamente, eles deverão exibir faces planas paralelas ao plano horizontal em número não superior a três. Só assim se poderá discernir entre dois ou mais objectos parcialmente sobrepostos através da análise dos respectivos histogramas 3D, aliás como já foi convenientemente descrito atrás.

A sequência de informação a registar em ficheiro será semelhante à já utilizada nos modelos 2D. Embora se trate de informação menos complexa que o dos modelos 2D, decidi manter o mesmo tipo de registo e de estrutura, ou seja um registo binário sequencial com comprimentos variáveis para cada registo, consoante o número de estados estáveis associados a cada modelo. Este processo permite poupar não só memória em disco, já que evita o registo de informação vazia, como também em RAM, visto que o ficheiro é inteiramente copiado para memória de uma só vez por uma questão de facilidade e rapidez na manipulação da informação.

Neste ficheiro, cada modelo possui um registo de 16 bytes correspondente ao seu nome, 1 byte que contém o número de estados estáveis que o modelo pode ter e de seguida, tantas estruturas de "Assinaturas" 3D quanto o número de estados estáveis atrás referido.

Para além da informação dos modelos propriamente dita, o ficheiro contém também separadores entre modelos. Esta inclusão foi extremamente útil na fase de depuração dos programas para verificar se os apontadores se posicionavam correctamente no início de cada modelo; aliás, esta verificação do correcto

posicionamento do apontador para a informação contida no "buffer" foi mantida em "run-time".

Foi também incluído no início do ficheiro (ao contrário do que aconteceu nos modelos 2D) um "byte" que contém o número total de modelos na base de dados.

As Figs. seguintes (20 e 21) ilustram o formato de registo e a estrutura de cada modelo:

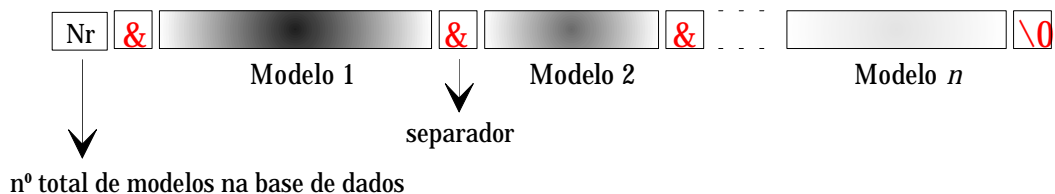


Fig. 20 - Registo em modo binário com comprimentos variáveis para cada modelo.

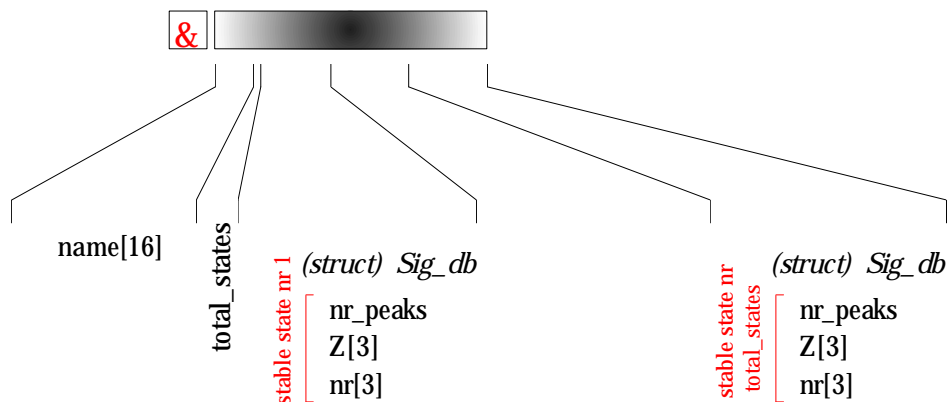


Fig. 21 - Estrutura interna de cada modelo 3D.

No Anexo H.3 é apresentada uma descrição pormenorizada da gestão "off-line" deste ficheiro de modelos 3D.

10.5 Conclusão

Foi aqui apresentado um modo de parametrizar os modelos 3D que envolve o conhecimento da altura a que superfícies planas pertencentes ao modelo se encontram do plano de suporte e da área dessas superfícies. Esses valores são medidos do histograma de alturas construído a partir da imagem de alturas e vão constituir, conjuntamente com outros dois valores (um também retirado do histograma, *sumNr2*, e outro medido quando da segmentação primária, *sumZ2*) aquilo a que se convencionou chamar "Assinaturas". Para que esses valores sejam correctos há a exigência dos objectos se encontrarem num dos seus estados estáveis. Esta abordagem permite a

separação dos dados referentes a cada um de dois objectos que se encontrem em sobreposição parcial. Ela permite ainda o cálculo de valores que permitam o manuseamento robótico seguro do objecto superior para posterior reanálise.

Esta parametrização irá permitir a identificação a 3D dos objectos contidos numa cena, desde que, como foi já dito, eles se encontrem num dos seus estados estáveis. Sendo assim, houve que estudar a fiabilidade dos vários parâmetros através do registo da sua variação para várias posições de um objecto sujeito a medida. Verificou-se que 3 dos parâmetros utilizados são adequados para uso na identificação já que apresentavam variações muito pequenas (6%, 3% e 5%) enquanto que o 4º se mostrou completamente inadequado e foi portanto abandonado na construção do algoritmo de identificação (o desvio padrão relativo a ele associado foi de 62%!).

A identificação a 3D implica também a construção de bases de dados com aquelas "Assinaturas", para todos os objectos de interesse em todos os seus estados estáveis. Para tal foi desenvolvido um programa que a permite construir quer por inserção de valores através do teclado quer por análise automática que envolve todos os sistemas 3D desenvolvidos: desde o sistema de varrimento laser até à medida das "Assinaturas", passando pela construção das imagens de altura, da segmentação primária daquelas imagens e da construção dos histogramas 3D. Para tornar o processo de identificação 3D mais eficiente foram ainda desenvolvidas funções de gestão do ficheiro de modelos que permitiram a execução de acções de consulta de actualização ou ainda de eliminação de modelos indesejados. Uma base de dados assim construída e cuidadosamente gerida com todas as opções atrás referidas ao dispor do utilizador, só poderá contribuir para uma maior eficiência no processo de identificação de objectos a 3D, essencial por sua vez ao bom desempenho e credibilidade do Agente Laser.

11.1 Introdução

Neste capítulo far-se-á a apresentação do programa de gestão da segmentação das imagens de altura, que controla também as várias chamadas ao algoritmo de identificação a 3D. Para além da segmentação primária, ele lança também, e sempre que necessário, um algoritmo especial de sub-segmentação destinado a separar objectos em estado de sobreposição parcial. No final da sua execução ele constrói os ficheiros para comunicação com a comunidade de agentes que serão enviados pelo programa de gestão do Agente Laser.

11.2 A Gestão da Segmentação

Das várias opções permitidas no Agente Laser, salienta-se aquela para a qual se dirigiu o maior esforço de programação e que permite a identificação e manipulação dos objectos na cena com base nas suas características tridimensionais.

Trata-se de um conjunto de programas, cada um com a sua função específica, gerido por uma função principal.

Embora tenha como objectivo principal a chamada encadeada da série de programas que constituem o referido conjunto, efectua também acções intercalares de verificação do tipo de objectos que se encontram na cena e prepara estruturas de dados que serão processados por cada um desses programas.

Após confirmação da existência em disco de uma imagem de "range" por parte do programa principal do Agente Laser (no caso dela não existir são desencadeados todos os meios para a criar) aquela função principal é chamada.

A sua primeira acção é a chamada ao algoritmo de análise de manchas com a informação da zona da imagem a inspeccionar (coordenadas do rectângulo que a confina) que por sua vez lhe houvera sido passada pelo programa de gestão do Agente. Este programa é uma versão adaptada do analisador de figuras bidimensionais. A sua acção traduz-se pela realização de uma segmentação primária da imagem de alturas. O resultado é um apontador para um vector de estruturas que contém toda a informação

geométrica a 2D de todas as figuras isoladas e detectadas naquela imagem de alturas. Esta função requer também o intervalo de valores de nível de cinzento a considerar. Nesta fase todos os valores são aceites, ou seja de 0 a 255. De toda a informação armazenada nesse vector, a mais importante, e na qual se irá basear a função de sub-segmentação, é sem dúvida o histograma de alturas associado a cada mancha, que é construído pelo programa de análise de manchas. Nesta fase, esses histogramas são opcionalmente desenhados no monitor, quer em "bruto" quer suavizados pela função "universalmente adoptada" para o efeito (10.2.5.2). A função chamada tem como objectivo a inspecção imediata de resultados por parte do utilizador. No modo de operação automática e autónoma do Agente Laser, estes desenhos em monitor não são efectuados por forma a minimizar o tempo de execução dos algoritmos e por conseguinte melhorar a sua capacidade de resposta.

O passo seguinte é a chamada ao algoritmo de construção de "Assinaturas" na qual se irá basear quer a sub-segmentação quer a identificação dos objectos. A informação fornecida é a altura máxima medida (em *mm*) e o apontador para o vector de "Assinaturas" que entretanto é criado e armazenado em memória. Este armazenamento é provisório uma vez que seguidamente é criado um outro vector onde ficarão de futuro registados todos os dados relativos aos objectos 3D, que para além da localização, orientação, nomes dos dois modelos mais prováveis acompanhados dos respectivos coeficientes de "matching", contém também um campo correspondente à "Assinatura". Sendo assim, o vector de "Assinaturas" é copiado integralmente para o dos objectos 3D e libertado de memória.

É permitida a visualização da informação de manchas e "Assinaturas" assim como o registo destas últimas quer em modo binário quer em texto. Mas das acções intercalares a que se fez referência, a que se destaca pela sua importância é o preenchimento por fases das estruturas dos objectos 3D, sempre que nova informação fica disponível. Nesta fase, por exemplo, o algoritmo copia alguns ângulos e posições de da estrutura referente às manchas detectadas na imagem para os respectivos campos da estrutura de objectos 3D. As manchas considerados são aquelas que apresentam no seu histograma de alturas apenas 1 pico ou 1 faixa e por isso não serão sujeitos a sub-segmentação.

Ainda nesta fase os objectos 3D (excluindo os atrás referidos) são sujeitos a uma primeira tentativa de identificação. Todos os que forem identificados (ou seja, aqueles para os quais o coeficiente de "matching" principal seja superior a um dado valor de referência) verão a respectiva "flag" posta a 1 (campo *identified* a que corresponde 1 caractere) e os elementos identificativos inseridos nos respectivos campos. De contrário, os campos correspondentes aos nomes dos dois modelos que mais se aproximam do objecto 3D e respectivos coeficientes de "matching" são mantidos a zero.

Todos os objectos 3D cujos histogramas possuam mais do que um pico, ou um pico e uma faixa e não tenham sido identificados pelo algoritmo anterior são agora sujeitos a sub-segmentação. Os vectores que contêm a informação das manchas e dos objectos 3D são aumentados consoante o número dos novos objectos encontrados. Neste programa são ainda copiados para os respectivos campos deste último vector a informação da posição e orientação corrigidos, referentes aos objectos em sobreposição mas completamente visíveis.

Não sendo mais necessária quer a informação geométrica contida no vector das manchas quer os histogramas de alturas (cujos apontadores se encontram registados num campo da estrutura daquele vector) são todos eles libertados da memória.

Resta agora considerar o caso de objectos com mais do que um pico, ou seja mais do que uma face plana paralela a XOY, e que não foram sujeitos a sub-segmentação por terem sido já identificados. Neste caso, há que corrigir os valores referentes à localização do centro de massa e à orientação. Não que eles sejam incorrectos, mas podem ser inadequados para o seu manuseamento robótico. (o registo da posição e orientação destinam-se precisamente a esse fim). E dado que o desnível entre faces pode ser apreciável, a ausência de ajuste daqueles valores pode implicar a falha na apreensão do objecto, ou pior ainda, a colisão da garra do robot com este! Sendo assim há que encontrar e medir o centro massa e orientação apenas da face superior (com maior Z).

É então chamada de novo a função de análise de manchas para obtenção de valores geométricos, mas apenas para o rectângulo na imagem que confina aquela face mais alta. Para além disso é fornecido ao algoritmo a gama de valores "grey-level" a considerar, correspondente apenas aos pixeis de "range" pertencentes a essa face. Desse modo e em termos práticos, tudo se passa como se só existisse aquela face na imagem. A posição e orientação são então corrigidas, garantindo assim um melhor manuseamento robótico do objecto.

Tal como aqueles, também os objectos associados a faixas (rodados no espaço) requerem ajuste do valor de ângulo e posição. Estando eles rodados no espaço, será mais eficiente a apreensão do objecto perto da borda de maior altura. E a garra do robot deverá estar orientada segundo a direcção dos Zs crescentes. Há então que medir essa direcção. Para esse efeito faz-se uso dos operadores de Sobel (com dimensão 3×3) já apresentados e utilizados no módulo Hough. A única diferença reside no facto de que as máscaras não serem aplicadas a pixeis adjacentes e sim distanciados de 20 pixeis para compensar o carácter discreto dos valores "grey-level" registados na imagem de "range". Deste modo, quando aplicados ao "centro de massa" cujas coordenadas se encontram registadas, fornecem, após alguns cálculos trigonométricos simples, um valor bastante aceitável para ângulo segundo a direcção dos Zs crescentes.

-1		1
-2		2
-1		1

h

-1	-2	-1
1	2	1

v

Fig. 1 - Máscaras de convolução referentes aos dois operadores de Sobel.

O novo ponto a considerar é obtido seguindo a direcção daquele ângulo através do objecto até se sair fora da área ocupada pelos pixeis de "range" pertencentes ao objecto, após o que é escolhido o último ponto considerado. Para verificação do correcto cálculo de ângulo e escolha do ponto, o processo é acompanhado pelo desenho no monitor dos vários pontos que vão sendo considerados e finalmente pelo desenho de uma cruz no ponto escolhido. Em todos os testes efectuados se obtiveram resultados

satisfatórios. De referir ainda que após este desenho em monitor para efeitos de controle e imediatamente antes do registo dos valores nos respectivos campos, o ângulo calculado (que se refere ao sistema de coordenadas do monitor) é convertido para o sistema de coordenadas do robot, aliás como acontece a todos os ângulos calculados no programa de análise de manchas. Só assim eles serão utilizáveis aquando do manuseamento robótico.

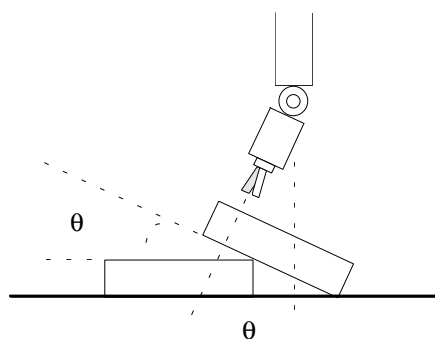


Fig. 2 - Os valores de posição e orientação a registar devem ser aqueles que facilitem o manuseamento robótico. No caso aqui apresentado de um objecto rodado no espaço, o ponto a indicar para posterior apreensão deverá estar localizado na zona de maiores Zs e a orientação deverá ser segundo os Zs crescentes. É ainda calculado um outro ângulo, θ , de rotação da garra para uma melhor apreensão.

Uma vez que os objectos considerados se encontram rodados no espaço, há também que calcular o ângulo daquela rotação, ou seja, de inclinação da face plana. Este cálculo é imediato, dada a maior diferença de Zs medida e o valor da extensão do objecto.

Estes valores de θ não são inseridos na estrutura dos objectos 3D uma vez que não existe um campo correspondente. Esse campo não foi definido pelo facto de não haver necessidade de manter esses valores de θ . Eles só serão mantidos em memória até à fase do programa em que os dados relativos aos objectos a remover são registados em disco para envio para o planeador de alto nível.

Nesta altura é chamado pela segunda vez o algoritmo de identificação que considerará todos os objectos ainda não identificados, mas que se destina principalmente aos que foram sub-segmentados. Os objectos parcialmente ocultos (*name="occluded"*) são contudo excluídos do processo assim como os rodados (*name="rotated"*) cuja ausência de informação ou posicionamento num estado "não estável" os faz sair fora do domínio e capacidade de identificação pelo processo das "Assinaturas".

Em termos de localização e orientação, todos os casos nesta fase estão já considerados:

- 1 pico ou 1 banda
- objectos sub-segmentados
- e objectos com mais de 1 pico

Para além dos ângulos, há também que converter as coordenadas registadas para o sistema do robot. Essa tarefa está a cargo de uma função própria que utiliza a matriz

com dimensão 4×3 de conversão (1.3.5). Os novos valores (X,Y) são então inseridos nos campos correspondentes da estrutura associada aos objectos 3D.

Finalmente, há que construir os ficheiros de informação a enviar para o planeador de alto nível.

Um ficheiro irá conter o nome de todos os ficheiros de informação de dados a enviar para a zona de acesso comum dos Agentes (excepto o próprio). Nesta opção, serão dois ficheiros. Um deles conterá uma lista de objectos presentes na cena, acompanhada das respectivas posições, orientações, nome dos modelos mais prováveis e coeficientes de "matching". Quanto ao outro, ele conterá as coordenadas dos pontos de apreensão dos objectos a remover (caso existam) para posterior análise, assim como o ângulo de orientação e o de inclinação.

A sintaxe geral dos ficheiros será a seguinte:

(ficheiro com informação dos objectos 3D)

```
...
position ( nome_do_objecto ; coef=92; X=50mm; Y=100mm; Z=16; angle=0.0277rad)
...
```

(ficheiro com informação para remoção de objectos)

```
...
remove (X=114.6mm; Y=13.8mm; Z=55.4mm; alpha=4,001rad; theta=-0.020)
...
```

Qualquer um destes novos ficheiros consiste numa descrição textual de comandos a introduzir no programa de transferência de ficheiros

Por fim, o vector de estruturas dos objectos 3D, é libertado da memória e a execução termina sendo o controle reenviado para o programa principal de gestão do Agente Laser.

No Anexo I.1 é apresentada uma descrição esquemática deste programa de gestão do Agente Laser assim como um esquema das suas chamadas a funções.

11.2.1 Localização do Objecto e Medição da Respectiva "Assinatura"

Uma das funções intercalares do programa de gestão, é o registo em ficheiro (modo textual) de resultados obtidos, como a localização de cada figura encontrada e a respectiva "Assinatura". No registo seguinte mostra-se um desses ficheiros referente a 1 pico + faixa, ou seja, dois objectos poliédricos com faces planas em que o objecto superior se encontra rodado no espaço:

```

1 Range Blob

----- Blob 0 -----
      jmin=262
imin=271  imax=402
      jmax=426
area=8207
x(cm)=338.7, y(cm)=338.8

----- Signature -----

```

```

sumz2 = 143458 (mm2)
sumNr2 = 39793 (pixel2)
nr_peaks = 1

----- peak 0 -----
Z[0]= 18, nr[0]=2792

----- band -----
Z[1] = 23; nr[1] = 12
Z[2] = 54; nr[2] = 2380
-----

```

11.3 A Segmentação Primária da Imagem de Alturas

11.3.1 Descrição Breve do Programa de Varrimento

Este programa é uma variante do já apresentado para a análise de figuras a 2D em imagens binarizadas. Esta versão considera imagens em nível de cinzento, que traduzem as alturas a que cada ponto visto na imagem original se situa do plano de apoio. A sua estrutura de dados foi alterada tendo em conta os novos requisitos: muitas variáveis agora desnecessárias foram eliminadas enquanto que outras foram acrescentadas. Destas salienta-se o registo de um apontador para uma área de memória a reservar que vai de encontro à necessidade de registar os histogramas 3D associados a cada figura detectada na imagem. De referir ainda a sua nova capacidade de seleccionar apenas parte dos pixels de "range" durante o processo de análise de figuras, que será aproveitado para efectuar a separação, numa segunda fase, de figuras correspondentes a objectos em sobreposição parcial: a sub-segmentação.

Nesta secção irei fazer uma descrição sumária das acções comuns àquele 1º programa e uma descrição detalhada das acções modificadas, ou novas que este agora incorpora.

No Anexo I.2 é apresentado um esquema detalhado de chamadas efectuadas por este programa e uma descrição da acção de cada uma das funções.

O tipo de imagem que ele analisa é agora do tipo "grey-level", ao contrário do que acontecia anteriormente que impunha que a imagem fosse previamente binarizada (esta imposição derivava apenas da necessidade de visualizar a correcta binarização da imagem com o nível de limiar adequado, não se devendo de forma alguma a requisitos do algoritmo). Ou seja, ele permite agora que os pixels apresentem quaisquer valores entre 0 e 255 o que é forçoso dado que se tratam de imagens com valores codificados de altura associados a cada pixel dos objectos. Para além disso há a necessidade de construir histogramas 3D a partir desses mesmos valores pelo que eles não podem ser corrompidos.

Houve portanto que introduzir alterações em todo o algoritmo começando como é de esperar pela função de aquisição de valores numa linha. Nela, a análise do bit 7 de cada pixel para detectar a presença do pixel de uma figura é substituída pela detecção de

um qualquer valor acima de 32 (00010000_B). Ou seja, qualquer pixel cuja operação AND bit a bit com a máscara 11110000_B dê como resultado 1 é considerado pertencente a uma figura e portanto incluído no conjunto consecutivo de 1's a que é dado o nome de "run-length pixels". Estes irão posteriormente ser considerados para a actualização dos parâmetros geométricos da figura a que pertencem. Obtêm-se assim dois registos da mesma linha consoante se destine à posterior análise geométrica ou à actualização do histograma 3D. O primeiro insere valores 1 ou 0 como resultado daquela operação lógica (obtendo-se assim uma binarização) enquanto o segundo mantém os valores originais dos pixeis.

Toda a análise geométrica da versão anterior é mantida, agindo como se existisse uma imagem binarizada e quase com a mesma estrutura de dados (alguns são agora desnecessários e outros são acrescentados para a análise a 3D). Alguns parâmetros, embora se mostrem posteriormente desnecessários, são mantidos prevendo uma eventual futura utilização.

11.3.2 A Construção das Figuras ou Manchas

A execução do programa no que diz respeito à construção das figuras e medição dos parâmetros geométricos é igual à da versão original; e uma vez que já foi descrito anteriormente em pormenor, ir-se-á aqui apenas descrevê-lo superficialmente.

As linhas são lidas consecutivamente desde o seu valor menor ($y=129$) até ao maior ($y=447$) sendo para cada uma agrupados os conjuntos consecutivos de pixeis 0 ou 1. Sempre que é detectado o fim de um destes conjuntos verifica-se a sua conectividade com alguma das figuras já existentes e resultantes das operações efectuadas para as linhas superiores, através da análise do conteúdo de uma janela de 2×2 . Esta janela permite para além disso (caso não haja conectividade) detectar outras situações, como é o caso por exemplo da presença de uma nova mancha, como resultado da análise da combinação de pixeis 0 e 1 (ver as funções *statei()* em 3.5.2). Caso haja essa conectividade, o conjunto referido de pixeis é incluído na figura, sendo então actualizados todos os parâmetros geométricos registados em estrutura.

11.3.3 A Actualização dos Parâmetros Geométricos

Quando se verifica que um dado conjunto consecutivo de pixeis pertencente à linha lida do ficheiro de imagem, possui conexão com uma figura já registada em vector, então há que actualizar os respectivos parâmetros geométricos. São eles os seguintes:

1. Perímetro
2. Área
3. Somatório dos valores de x
4. Somatório dos valores de y
5. Somatório dos quadrados de x

6. Somatório dos valores de $(x \times y)$
7. Somatório dos quadrados de y
8. Conteúdo do histograma 3D
9. Somatório dos quadrados de Z

Os 7 primeiros parâmetros são comuns ao algoritmo original apenas havendo a salientar o acréscimo dos pontos 8 e 9.

Alguns destes valores são posteriormente utilizados para calcular a posição e orientação da figura. O histograma 3D será utilizado para a sua identificação a 3D.

A actualização de cada um destes parâmetros geométricos é então efectuada da seguinte maneira:

1. *Perímetro:*

A inclusão de mais uma linha à figura já registada no vector de manchas, origina o incremento do perímetro em duas unidades referentes aos pixels extremos. O incremento do valor do perímetro na horizontal é efectuado dentro de algumas das funções *statei()* (3.5.2) quando se completam conjuntos consecutivos de pixels 0 ou 1 pertencentes a uma fronteira da mancha.

2. *Área:*

Se o conjunto de pixels em linha for em número de L , então será este o incremento para o valor de área da figura. A área total virá então igual ao número total de pixels constituintes da figura.

3. *Somatório dos valores de j (Σj):*

Uma vez que todos os pixels possuem o mesmo valor de ordenada j , o incremento no somatório dos js será simplesmente $L \times j$.

4. *Somatório dos quadrados de j (Σj^2):*

Do mesmo modo o incremento deste parâmetro será de $L \times j^2$.

5. *Somatório dos valores de i (Σi):* $S = L.k + \frac{L.(L-1)}{2}$

Se o 1º pixel tiver como valor de abcissa k , e o conjunto totalizar L pixels, então aquele somatório constitui uma progressão aritmética de razão 1 cujo resultado é o apresentado atrás.

6. *Somatório dos valores de $i \times j$ ($\Sigma i \times j$):* $S = j.L \left[k + \frac{(L-1)}{2} \right]$

Uma vez que a ordenada j é igual para todos os pixels, então aquele somatório vem igual ao anterior multiplicado simplesmente por j .

$$7. \text{ Somatório dos valores de } i^2 (\Sigma i^2): \quad S(n) = \sum_{i=0}^n i^2 = \frac{1}{6} n(n+1)(2n+1)$$

O resultado do somatório, cuja demonstração pode ser consultada em 3.6.7, pode ser posto numa outra forma, sabendo que o conjunto tem comprimento L e a abcissa do 1º pixel é k :

$$S = L \cdot \left[k^2 + k(L-1) + \frac{1}{6}(L-1)(2L-1) \right]$$

8. Conteúdo do histograma 3D:

A actualização do conteúdo do histograma é feita inspeccionando o valor de altura codificado em nível de cinzento de cada um dos pixels pertencentes ao conjunto. Uma vez que o nº de entradas do histograma é de 256 e portanto igual ao de níveis de cinzento, então o valor deste indica imediatamente o elemento do vector a ser incrementado. A localização em memória desse vector (valor do apontador) é obtida pela leitura num dos campos da estrutura de dados do vector de características geométricas; único valor associado ao histograma, aliás, que fica registado nesse vector.

9. Somatório dos quadrados de Z :

Este parâmetro é obtido pela soma dos quadrados dos valores de Z codificados em níveis de cinzento pertencentes a todos os pixels do conjunto consecutivo de 0's ou 1's. O resultado deve ser somado ao que já consta no campo correspondente da estrutura de dados geométricos (uma vez conhecido o objecto/mancha a que o conjunto de pixels pertence).

No fim do varrimento da imagem, todos os valores deste parâmetro são convertidos para mm^2 conhecido o valor $Z_{máx}$, obtendo-se assim valores passíveis de comparação com os obtidos por avaliação analítica, conhecidas as dimensões dos objectos. Estes valores finais vão consistir no momento de inércia geométrico referente ao plano horizontal, e correspondem portanto a um parâmetro potencialmente identificador dos objectos.

11.3.4 As Funções de Manipulação de Histogramas 3D

Para além de acções isoladas de correcção ou actualização dos histogramas 3D que são inseridas em funções já existentes, foram criadas e acrescentadas às já existentes mais duas funções directamente ligadas àqueles histogramas: a de criação e fusão de histogramas. Sempre que é detectada a presença de uma nova figura ou mancha, um

novo elemento com estrutura de dados geométricos, é acrescentada ao conjunto já existente, sendo todos os respectivos valores geométricos postos a zero. É também criado um novo vector com 256 inteiros que irá conter o respectivo histograma 3D e cujo apontador é copiado para aquela estrutura.

A segunda função destina-se a fundir dois histogramas quando o mesmo se verifica com as respectivos manchas. Esta operação consiste simplesmente no somatório dos valores de entrada de ambos os vectores sendo estes registados naquele que prevalece. O outro vector é simplesmente libertado de memória e o valor do apontador (num campo do vector das manchas) é posto a NULL por uma questão de segurança. A entrada no vector das manchas referente àquela que foi fundida, irá simplesmente ser ignorada por já não corresponder a nenhum objecto.

11.4 O Programa de Sub-Segmentação

11.4.1 Apresentação do Programa

Este é o programa que analisa os parâmetros extraídos previamente dos histogramas 3D e de acordo com eles efectua a separação dos dois ou mais objectos de cada situação detectada de sobreposição.

Para além dessa separação, que envolve inclusivé a eliminação dos pixeis de altura associados ao objecto sobreposto, não só para confirmar que o algoritmo considerou os pixeis correctos como também para melhor visualização de todo o processo, este programa ordena uma nova análise de parâmetros geométricos com construção de novo histograma de alturas. Essa análise fornecerá sem dúvida elementos úteis para a manipulação robótica, mas não menos importante que isso, confirma ou actualiza os parâmetros que serão utilizados posteriormente numa nova tentativa de identificação.

11.4.2 A Acção do Algoritmo

Este algoritmo actua em apenas um elemento do vector de objectos de cada vez, sendo chamado tantas vezes quantos os casos detectados de possível oclusão.

Os dados por ele recebidos são o nº da figura ou mancha a sub-segmentar, um apontador para uma estrutura auxiliar que contém os apontadores para os vectores de manchas (contendo a análise geométrica integral para as manchas isoladas detectadas na imagem de alturas) e dos objectos (que consiste numa série de estruturas completas onde serão registados todos os dados relativos aos objectos encontrados, como a sua posição e orientação, a sua identificação acompanhada do respectivo coeficiente de "matching" e a sua "Assinatura" obtida a partir do histograma de alturas construído aquando do varrimento da imagem de alturas).

A sua acção resume-se essencialmente a dois pontos.

1. Tendo como base de análise a "Assinatura" da mancha detectada, cujas situações contempladas são a ocorrência de dois a três picos no histograma (que correspondem a duas a três superfícies planas paralelas a XOY), ou um pico acompanhado de faixa (que corresponde ao objecto supostamente rodado no espaço devido a sobreposição, estando o 2º objecto parcialmente oculto por aquele e acompanhada de informação incompleta deste último, possuidor de uma superfície paralela a XOY).

De posse então da "Assinatura", ele separa a informação supostamente associada a dois (ou três) objectos e regista-os em elementos separados do vector que entretanto é aumentado em número igual ao total de novos objectos distintos encontrados menos um, sendo o elemento onde se encontrava a "Assinatura" original devidamente actualizado apenas com a informação referente ao objecto visível (Fig.3).

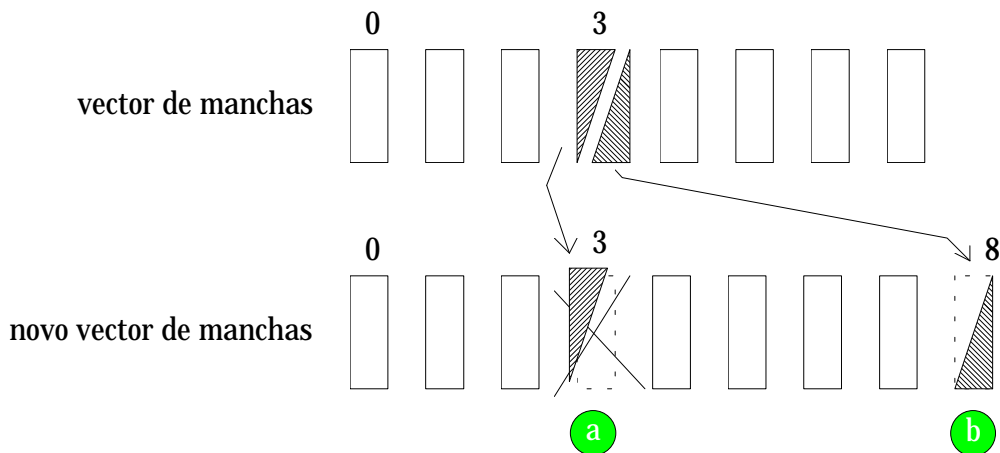


Fig. 3 - Separação da informação referente aos dois objectos;

- (a) o antigo elemento de entrada vai estar associado ao objecto visível, sendo necessário um novo varrimento para obter os seus parâmetros característicos;
- (b) é acrescentado ao vector um novo elemento referente ao objecto sobreposto não só para salientar a sua existência como para uma eventual futura inserção de valores (que requer a retirada do objecto em sobreposição e um novo varrimento).

Os novos elementos do vector referentes ao objecto ou objectos ocultos vão conter informação incompleta ou inadequada, pelo que são postos a zero para salientar bem aquele facto. A razão da sua inclusão no vector deve-se à eventual futura inserção de dados correctos após a remoção dos objectos sobrepostos e sua consequente separação. Por este motivo é mantida na informação da mancha as coordenadas do rectângulo delimitador.

2. No segundo ponto referido, é feita de novo a análise da mancha com, mas agora limitada no espaço, cujas coordenadas se encontram registadas no vector das

manchas, e na altura, cujos limites de nível de cinzento foram registados no campo da "Assinatura" referente a $Z[]$ (e obtidos da análise do histograma de alturas) após conversão para mm com $Z_{máx}$ (esses limites de nível de cinzento são reconvertidos a partir do mesmo $Z_{máx}$). A razão para esta nova análise baseia-se no facto da "Assinatura" conter, para além das alturas $Z[]$ e áreas $nr[]$ que são inalteradas (os Zs apenas são subtraídos aos valores anteriores para corresponderem à verdadeira altura de cada um dos objectos 3D detectados), também alguns valores auxiliares como o momento de inércia geométrico da superfície visível, relativamente a XOY ($sumZ2$) e o momento de 2ª ordem do histograma relativamente ao eixo "grey-level" ($sumNr2$). Estes valores vêm corrompidos pela existência do objecto oculto pelo que têm que ser corrigidos. E isto só é conseguido mediante nova chamada ao programa de varrimento e medida de características geométricas, aplicado apenas aos pixels da imagem de "range" que pertencem ao objecto de cima; e daí a utilização dos limites de "grey-level" que o algoritmo reconhece e contempla na sua análise.

Por outro lado, há também que corrigir tanto a posição do centro de massa como a orientação do objecto, valores que também são medidos por aquele algoritmo. Estes novos valores são então inseridos no elemento do vector dos objectos correspondente ao objecto visível, sendo essenciais para a sua remoção após pedido expresso ao Planeador de Alto Nível (HLP) nesse sentido. No caso do objecto se encontrar rodado (a que corresponde a faixa no histograma de alturas) o ângulo é posteriormente corrigido (não neste programa) e corresponderá à direcção do aumento das alturas, para melhor manuseamento robótico.

Por último, relembra-se com brevidade o modo de medir $sumNr2$ a partir do histograma de alturas obtido no programa de varrimento após a nova chamada. Atendendo ao carácter descontínuo dos pixels de "range", o histograma de alturas apresenta uma oscilação nos seus valores que dificulta a obtenção do valor pretendido. Sendo assim, ele é sujeito a suavização prévia com um algoritmo especialmente desenvolvido para o efeito, após o que aparecem já bem definidos os picos e as faixas (correspondendo respectivamente a objectos não rodados e rodados no espaço, possuidores de superfícies planas, que nos seus estados estáveis são paralelos a XOY). Das várias funções testadas foi seleccionada a apresentada em 10.2.5.2. Esta escolha seguiu-se à utilização do programa interactivo, tal como se mostra naquela secção, que utiliza aquelas quatro funções e as aplica ao histograma original (ou já previamente sujeito a suavização) e as chama por escolha do utilizador, por qualquer ordem e em qualquer situação pretendida. Essa total liberdade facultada ao utilizador deve-se ao facto do algoritmo manter sempre em memória, quer o histograma de alturas original, quer o imediatamente anterior. Cada nova suavização ou retoma do histograma em memória, é acompanhada do "display" em monitor do novo histograma, onde as alterações serão visíveis, o que torna este algoritmo agradável e totalmente interactivo!

Qual a necessidade de uma escolha cuidada da função de suavização? Ela deve-se decerto à necessidade de melhor suavizar o histograma de modo a poder-se extrair correctamente os parâmetros que caracterizam quer os picos quer as faixas. Após a sua escolha, a função de suavização será "universalmente" adoptada em todo e qualquer algoritmo que envolva o uso de histogramas de altura! Refiro-me em particular aos algoritmos de cálculo de $sumNr2$, de identificação 3D e de construção de modelos 3D. a

única forma de poder utilizar aquele dado como parâmetro de identificação, constando portanto do conjunto de características do modelo a registar no respectivo ficheiro de modelos, é ele ser obtido sempre do mesmo algoritmo de suavização; caso contrário os valores calculados e registados em ficheiro não teriam correspondência possível. A função suavizadora seleccionada, que utiliza uma máscara de convolução 7×1 , foi portanto adoptada em todo e qualquer algoritmo que utilizasse o parâmetro *sumNr2*.

No Anexo I.3 apresenta-se uma descrição esquemática do programa de sub-segmentação e o seu esquema de chamadas a funções.

11.4.3 1º Exemplo: Caso Real de dois Poliedros Sobrepostos e nos seus Estados Estáveis (esquemático atrás)

Esta constitui a situação mais frequente de sobreposição de poliedros simples que os algoritmos 3D têm de resolver. A Fig.4 apresenta um desenho de uma dessas situações. Nas Figs. 5 e 6 são mostradas as representações bidimensional, tal como visto no ecrã do monitor, e tridimensional.

Ambas as representações são obtidas a partir do conteúdo de qualquer dos ficheiros de amostragem de alturas (em registo em binário ou textual).

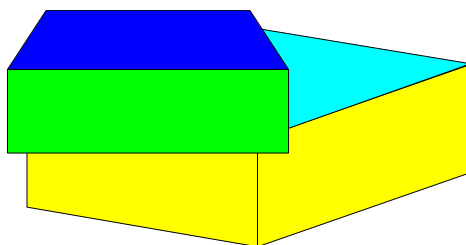


Fig. 4 - Caso simples de uma sobreposição de dois poliedros.

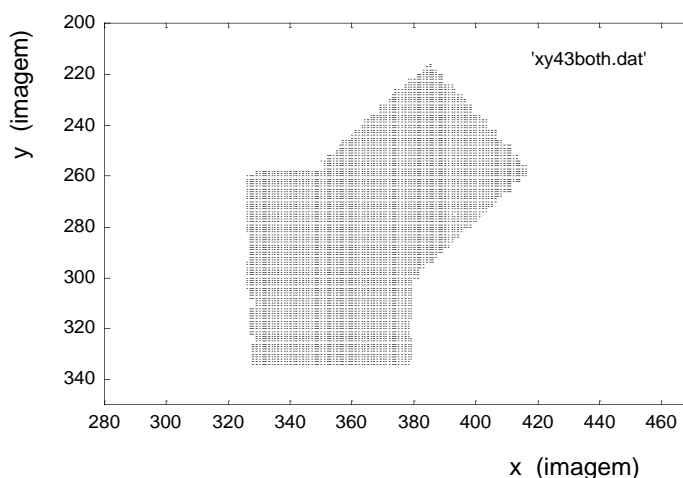


Fig. 5 - Imagem de alturas referente a dois poliedros em sobreposição. Esta representação bidimensional dos objectos foi obtida por aplicação de um programa auxiliar que acede à imagem registada na memória da placa de aquisição.

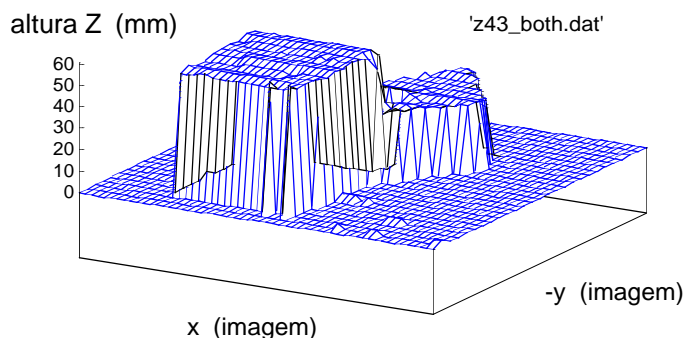


Fig. 6 - Representação tridimensional dos mesmos objectos (também obtida por aquele programa).

11.4.4 A Separação dos Dois Objectos

Após a execução do algoritmo de análise de "Assinaturas", e consequente preenchimento dos respectivos campos da estrutura de dados associada, constata-se a existência de dois picos. A cada um está associada um valor de altura $Z[i]$ diferente a que irão corresponder dois valores diferentes de nível de cinzento por utilização de $Z_{máx}$. A semi-diferença daqueles valores define então uma fronteira que permitirá separar os dois conjuntos de pixels de altura (Fig.7).

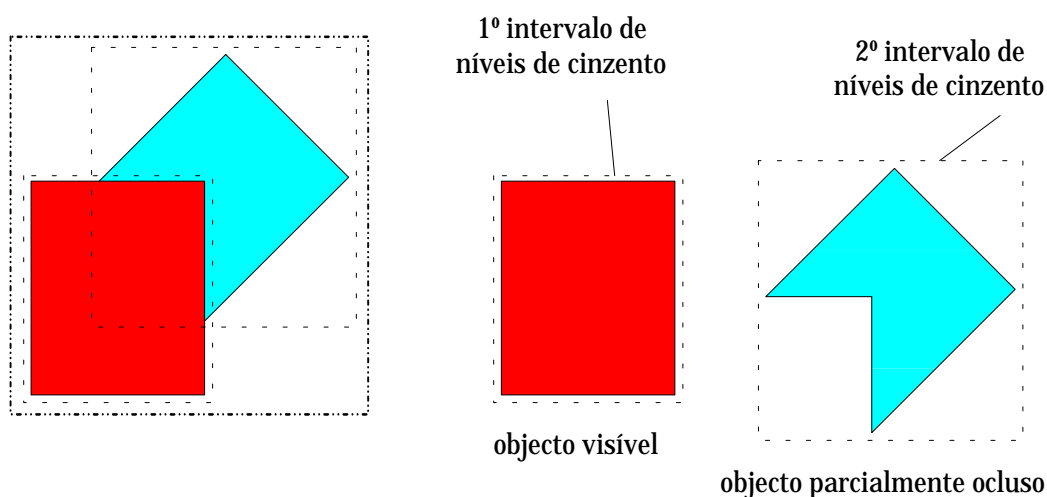


Fig. 7 - Vista dos objectos pela câmara. Os parâmetros extraídos do histograma 3D original permitem a separação em dois intervalos distintos de altura, referentes a cada uma das figuras.

A separação efectiva só é efectuada após um novo varrimento da imagem para a análise de características geométricas e a execução do programa de extracção de "Assinaturas". Aquele varrimento só contempla então o 2º conjunto de pixels, ou seja aqueles situados acima daquela fronteira, e que correspondem ao objecto completamente visível. A área

de varrimento mantém-se contudo a original, com a certeza de que não irá considerar os pixels pertencentes ao objecto parcialmente oculto (Fig.8).

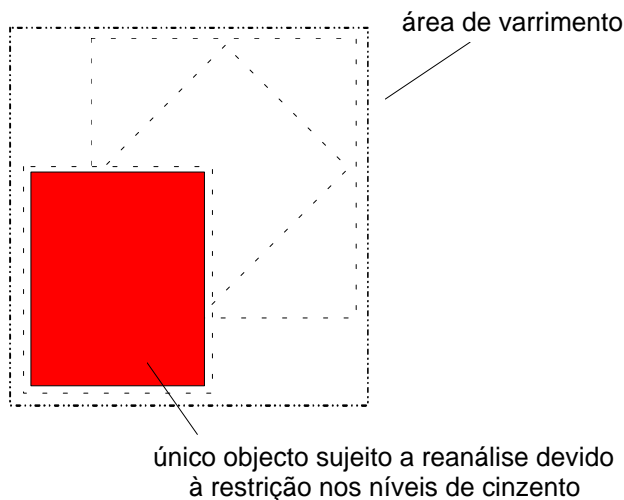


Fig. 8 - O novo varrimento ordenado para a sub-segmentação, contempla apenas o objecto assinalado que está associado ao 2º pico do histograma 3D uma vez que impõe restrição ao nível de cinzento dos pixels; a área considerada, contudo, mantém-se, correspondendo ao rectângulo delimitador da figura original.

11.4.5 Resultados Experimentais da Separação dos dois Objectos

As Figs. 9 e 10 correspondem às representações bi e tridimensional do objecto completamente visível e que é sujeito a reanálise. Os valores numéricos utilizados nestas representações foram obtidos pela aplicação directa do programa auxiliar que acede ao ficheiro de imagem registado na placa de aquisição, e aceita um intervalo de valores de nível de cinzento.

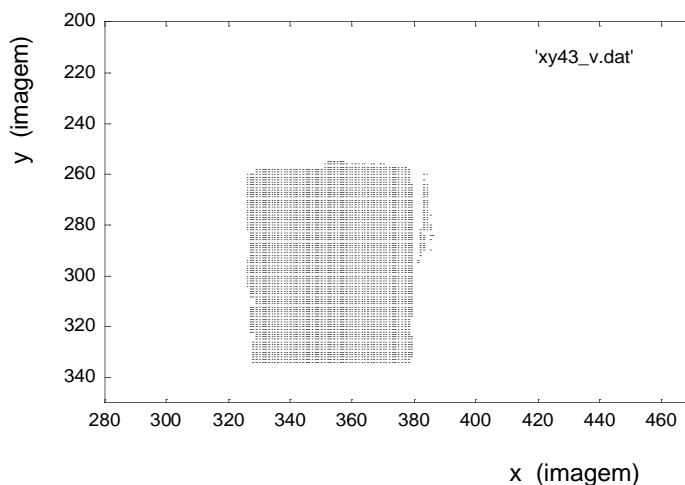


Fig. 9 - Esta imagem de alturas representa apenas o objecto visível que será considerado num novo varrimento para medição de parâmetros; esta representação é possível graças à liberdade de escolha do intervalo de níveis de cinzento pretendidos.

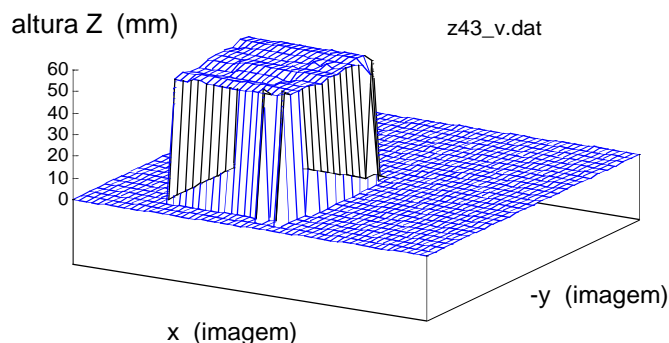


Fig. 10 - Representação tridimensional do objecto completamente visível

O mesmo programa auxiliar permite obter a representação do objecto parcialmente oculto pela selecção de valores de nível de cinzento delimitadores correspondentes aos seus pixels de altura.

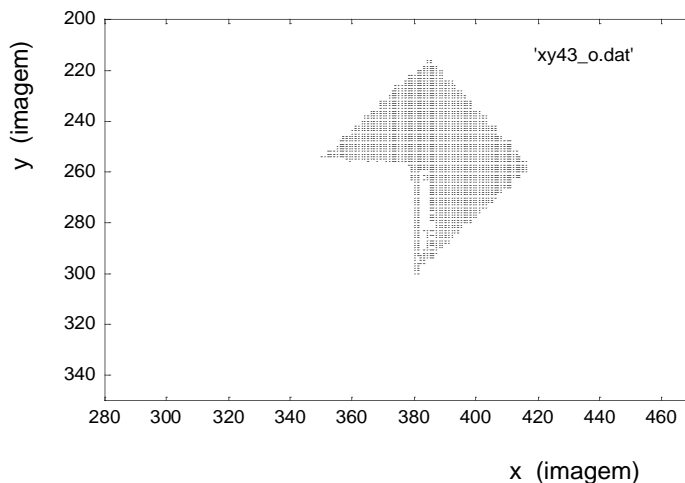


Fig. 11 - Esta imagem de alturas representa apenas o objecto parcialmente oculto; de novo, esta representação é possível graças à liberdade de escolha do intervalo de níveis de cinzento pretendidos. Esta figura já não será alvo de novo varrimento uma vez que se encontra incompleta. Ela é inclusivamente apagada quando da análise da 1ª figura para confirmação visual do bom funcionamento do algoritmo de sub-segmentação.

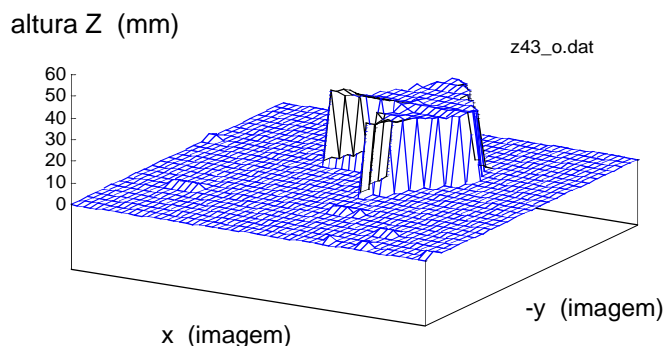


Fig. 12 - Representação tridimensional do objecto parcialmente oculto. Apenas a parte visível se encontra representada.

Da aplicação do programa de sub-segmentação resulta a modificação do conteúdo e extensão dos vectores de manchas e de objectos 3D devido à separação e correcção de parâmetros relativos a cada um dos objectos envolvidos na sobreposição. Na Fig.13 pode-se observar o resultado da sua aplicação no exemplo apresentado.

1 Range Blob	2 Range Blobs
<pre> ----- Blob 0 ----- jmin=216 imin=326 imax=416 jmax=334 area=6383 x(cm)=365.0, y(cm)=280.5 ----- Signature ----- name = sumz2 = 12047647 (mm²) sumNr2 = 46970 (pixel²) nr_peaks = 2 ----- peak 0 ----- Z[0]= 25, nr[0]=2122 ----- peak 1 ----- Z[1]= 49, nr[1]=4025 ----- </pre>	<pre> ----- Blob 0 ----- jmin=255 imin=326 imax=379 jmax=334 area=4125 x(cm)=352.9, y(cm)=295.3 ----- Signature ----- name = visible sumz2 = 10327152 (mm²) sumNr2 = 26272 (pixel²) nr_peaks = 1 ----- peak 0 ----- Z[0]= 24, nr[0]=4025 ----- ----- Blob 1 ----- jmin=216 imin=326 imax=416 jmax=334 area=0 x(cm)=0.0, y(cm)=0.0 ----- Signature ----- name = occluded sumz2 = 0 (mm²) sumNr2 = 0 (pixel²) nr_peaks = 1 ----- peak 0 ----- Z[0]= 25, nr[0]=2122 ----- </pre>

Fig. 13 -Registo em texto dos dados relativos a ambos os objectos. De notar que os parâmetros de identificação auxiliares, *sumZ2* e *sumNr2*, são postos a zero, assim como a posição e a área. Apenas as coordenadas do rectângulo delimitador é mantido para uma eventual futura reanálise após retirada do objecto superior.

11.4.6 2º Exemplo: Sobreposição de dois Poliedros com Rotação no Espaço do que se Encontra Visível

Do mesmo modo que anteriormente pode-se obter a representação bi e tridimensional dos pixels de altura. As Figs. 14 e 15 correspondem a essas representações.

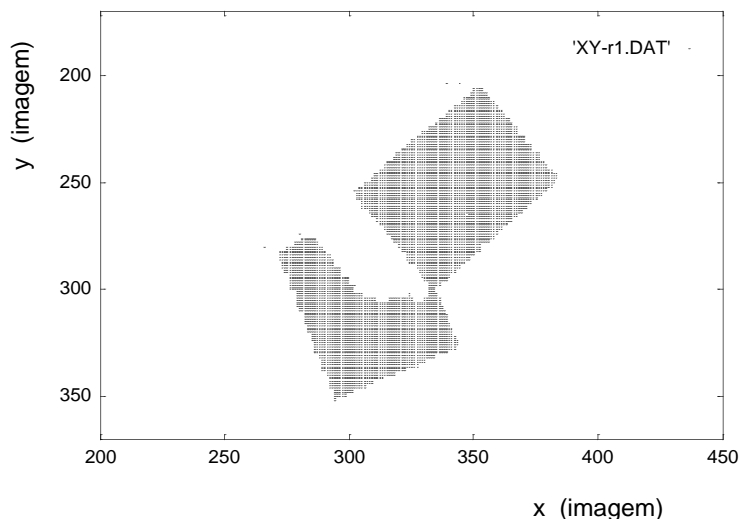


Fig. 14 - Imagem de alturas referente a dois poliedros em sobreposição encontrando-se um deles rodado no espaço (de notar a falta de informação de alturas na zona da "sombra" provocada pelo objecto superior). Esta representação bidimensional dos objectos foi mais uma vez obtida por aplicação de um programa auxiliar que acede à imagem registada na memória da placa de aquisição.

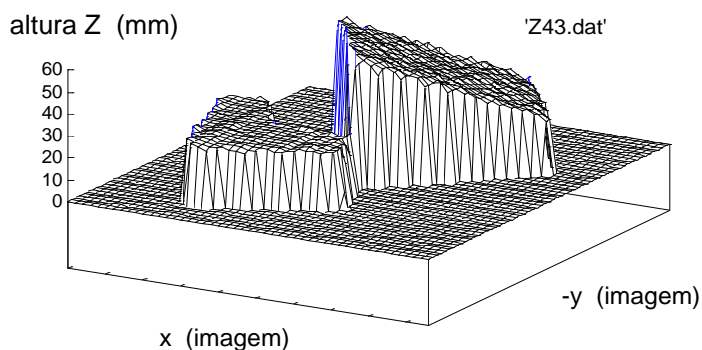


Fig. 15 - Representação tridimensional dos mesmos objectos (também obtida com o programa auxiliar já referido atrás).

11.4.7 Resultados Experimentais da Separação dos dois Objectos

Nas Figs. 16 e 17 encontram-se representados apenas os pixels de altura pertencentes ao objecto completamente visível e rodado no espaço e que são considerados no novo varrimento e medida de "Assinatura".

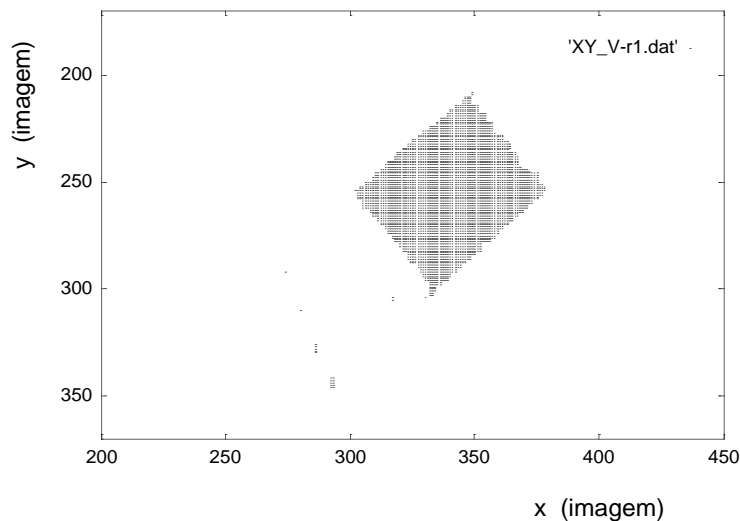


Fig. 16 - Esta imagem de alturas representa apenas o objecto visível que será considerado num novo varrimento para medição de parâmetros; esta representação é possível graças à liberdade de escolha do intervalo de níveis de cinzento pretendidos.

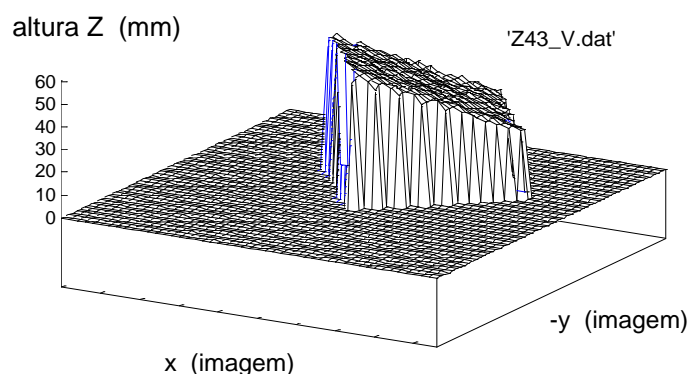


Fig. 17 - Representação tridimensional do objecto completamente visível.

Nas Figs. 18 e 19 encontram-se representados os pixels não considerados no novo varrimento, ou seja, todos aqueles situados abaixo da fronteira que separa o pico da banda no histograma 3D. De notar que parte do objecto visível não foi contemplado no novo varrimento por possuir pixels com alturas iguais às do objecto parcialmente

oculto. Isto não traz consequências de maior uma vez que a "Assinatura" medida não pode ser utilizada para identificação por se referir a um objecto rodado no espaço e portanto num estado não estável. Ela contém na realidade dados numéricos quer retirados directamente do histograma 3D quer calculados a partir de alguns pixels da imagem, que permitem o posterior manuseamento robótico do objecto; e os pixels ignorados não interferem no cálculo daqueles dados.

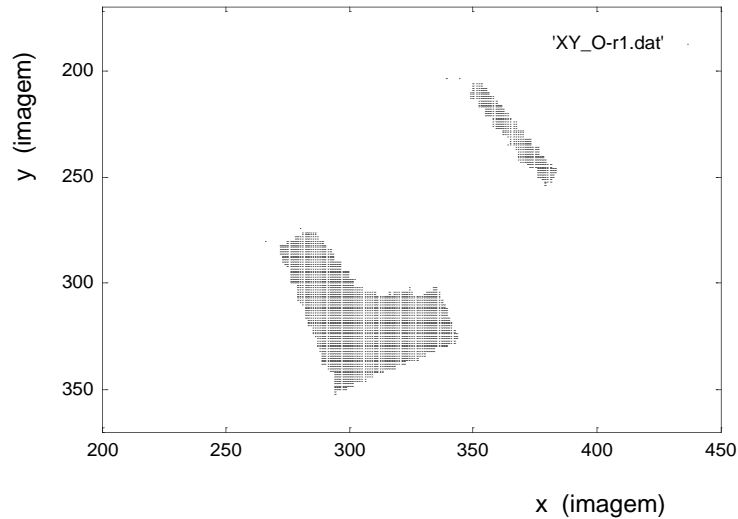


Fig. 18 - Esta imagem de alturas representa apenas o objecto parcialmente oculto; de novo, esta representação é possível graças à liberdade de escolha do intervalo de níveis de cinzento pretendidos. Esta figura já não será alvo de novo varrimento uma vez que se encontra incompleta. De notar que a sub-segmentação retirou parte do objecto superior por esta se encontrar à mesma altura do objecto parcialmente oculto. Este facto não interfere contudo na escolha do melhor ponto para apreensão do objecto que se deverá encontrar na sua parte mais alta.

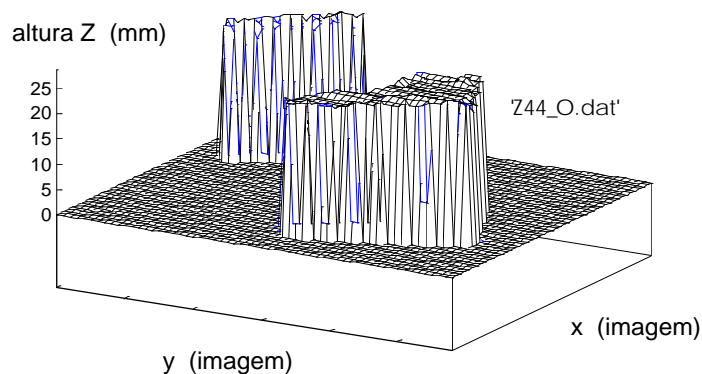


Fig. 19 - Representação tridimensional do objecto parcialmente oculto assim como da parte rejeitada do objecto superior.

Mais uma vez, da aplicação deste algoritmo obtém-se a separação dos dados relativos a cada um dos objectos com o reconhecimento da respectiva situação: *name*="rotated" para o objecto visível e *name*="occluded" para o objecto parcialmente oculto (Fig.20).

1 Range Blob	2 Range Blobs
<pre> ----- Blob 0 ----- jmin=206 imin=272 imax=383 jmax=352 area=6596 x(cm)=327.4, y(cm)=277.5 ----- Signature ----- name = sumz2 = 1809433 (mm²) sumNr2 = 11112 (pixel²) nr_peaks = 1 ----- peak 0 ----- Z[0]= 26, nr[0]=2627 ----- band ----- Z[1] = 29; nr[1] = 27 Z[2] = 59; nr[2] = 3352 ----- </pre>	<pre> ----- Blob 0 ----- jmin=210 imin=302 imax=377 jmax=303 area=3486 x(cm)=339.6, y(cm)=255.3 ----- Signature ----- name = rotated sumz2 = 7051674 (mm²) sumNr2 = 5713 (pixel²) nr_peaks = 0 ----- band ----- Z[0] = 29; nr[0] = 27 Z[1] = 59; nr[1] = 3352 ----- ----- Blob 1 ----- jmin=206 imin=272 imax=383 jmax=352 area=0 x(cm)=0.0, y(cm)=0.0 ----- Signature ----- name = occluded sumz2 = 0 (mm²) sumNr2 = 0 (pixel²) nr_peaks = 1 ----- peak 0 ----- Z[0]= 26, nr[0]=2627 ----- </pre>

Fig. 20 - Registo em texto dos dados relativos a ambos os objectos. De notar que no caso do objecto parcialmente oculto os parâmetros de identificação auxiliares, *sumZ2* e *sumNr2*, são postos a zero, assim como a posição e a área. Apenas as coordenadas do rectângulo delimitador é mantido para uma eventual futura reanálise após retirada do objecto superior.

Os novos valores encontrados para *area* e *sumNr2*, naquele caso, podem ser usados para limitar a escolha de possibilidades de identificação uma vez que não diferem muito dos valores registados para o estado estável correspondente. Os valores de *Z[0]* e *Z[1]* são utilizados posteriormente para calcular o ponto de apreensão e a orientação para um manuseamento robótico eficaz.

11.5 Conclusão

Por análise de todos os resultados anteriores, pode-se concluir que o processo de segmentação da imagem de alturas mostrou-se eficiente, quer na sua acção primária quer na sub-segmentação posterior sempre que se verificava a ocorrência de sobreposição parcial.

A gestão de todo este processo, acompanhada de acções intercalares tais como chamadas a identificações parciais, cálculo de parâmetros auxiliares para manipulação robótica ou ainda o preenchimento das estruturas referentes aos objectos 3D, mostrou-se igualmente eficiente como o comprovam os resultados das experiências executadas.

12.1 Introdução

O objectivo deste programa é identificar um ou mais objectos presentes na cena, por comparação com modelos registados previamente numa base de dados. No algoritmo que lhe corresponde é feita a suposição de que o objecto se encontra num dos seus estados estáveis como já foi dito, pelo que aquele ficheiro tem de contemplar todas as possibilidades de orientação no espaço para cada modelo. E de facto (ver Anexo H.3) toda essa informação está contida no ficheiro por ordem inversa de estabilidade: a cada conjunto de informações relativas a um dado estado estável será atribuído um número entre 1 e 3, sendo ele tanto menor quanto maior for a sua estabilidade. Sendo assim o 1º modelo a ser lido refere-se sempre ao estado estável nº1.

A medida da proximidade de um objecto a um modelo, num dos seus estados estáveis, pode ser dada pelas diferenças relativas entre os parâmetros identificadores correspondentes de um e outro, restando apenas definir uma função de erro a partir de todos aqueles valores. Por exemplo, se definir um espaço Cartesiano com dimensão $(1+nr_peaks \times 2)$ correspondendo aos parâmetros identificadores $sumZ2$, $Z[nr_peaks]$ e $nr[nr_peaks]$, então, os seus valores no caso do objecto, representados em cada um dos eixos, vão definir um vector naquele espaço (Fig.1).

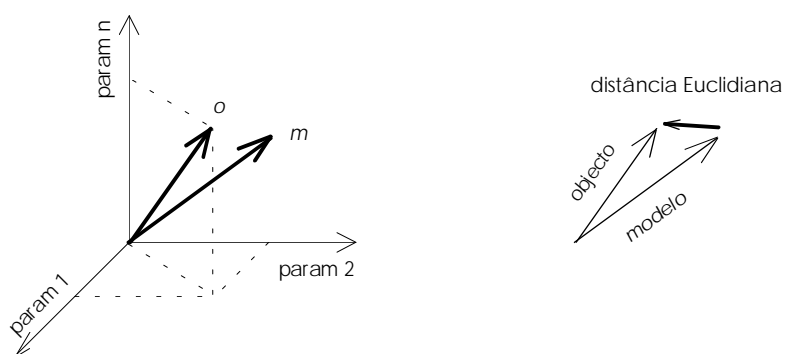


Fig. 1 - Representação da distância Euclidiana no espaço multidimensional dos parâmetros identificadores

Se designar cada um desses parâmetros por uma variável genérica x_{obj_i} , então esse vector poderá ser escrito na seguinte forma:

$$\vec{obj} = \sqrt{\sum_{i=1}^n (x_{obj_i})^2} \cdot \vec{u}$$

Da mesma forma pode-se definir um vector modelo:

$$\vec{mod} = \sqrt{\sum_{i=1}^n (x_{mod_i})^2} \cdot \vec{u}$$

A distância Euclidiana entre objecto e modelo poderá então ser dada pelo módulo do vector diferença:

$$\Delta d = \sqrt{\sum_{i=1}^n (x_{obj_i} - x_{mod_i})^2}$$

E o erro relativo por:

$$\frac{\Delta d}{mod} = \frac{\sqrt{\sum_{i=1}^n (x_{obj_i} - x_{mod_i})^2}}{\sqrt{\sum_{i=1}^n (x_{obj_i})^2}}$$

Alternativamente posso definir a medida da aproximação pelo coeficiente de "matching" c através da relação:

$$c = 100 \times (1 - \Delta d / mod)$$

que tomará valores entre 0% e 100%.

A semelhança do objecto com o modelo será tanto maior quanto maior for esse coeficiente c . Prevendo que possa haver alguma ambiguidade na atribuição ao objecto de um de dois modelos muito próximos, inclui-se na sua estrutura de informação a referência não a um mas a dois modelos, ficando o algoritmo de procura encarregado de actualizar essas referências em cada comparação, fornecendo no final os dois melhores modelos encontrados.

Abaixo de que valor deverá o modelo ser completamente rejeitado e o objecto ser considerado, nessa comparação, como não identificado? Decerto ele deverá ser uma função das tolerâncias medidas dos parâmetros identificadores. Contudo essa função não é óbvia nem imediata pelo que opto por um outro critério de rejeição substancialmente mais transparente: se σ_0 , σ_1 e σ_2 forem as tolerâncias encontradas para os 3 parâmetros identificadores, então defino um σ global dado por:

$$\sigma = \sigma_0 + nr_peaks \times (\sigma_1 + \sigma_2)$$

No caso dos parâmetros utilizados os valores numéricos são respectivamente 6%, 3% e 5% (Fig.2), pelo que σ poderá variar entre 6% e $6+3 \times (3+5)=30\%$ consoante o nº de faces planas que o modelo apresenta. Para cada modelo a comparar dever-se-á então calcular previamente o valor do σ global. Este valor, como é óbvio, não traduz nenhum erro real mas apenas um valor calculado com o objectivo estrito de efectuar uma comparação.

	<i>sumZ2</i>	<i>sumNr2</i>	<i>Z[j]</i>	<i>nr[j]</i>
σ	6%	62%	3%	5%

Fig. 2 - Quadro de resultados experimentais para a tolerância dos parâmetros.

Naquele σ não foi considerado o parâmetro *sumNr2*: embora tenha havido um esforço no sentido de minorar o desvio relativo associado a ele, nomeadamente adoptando uma função única para a suavização dos histogramas de altura (10.2.5.2), verificou-se que para aquele parâmetro, não se conseguiu obter uma tolerância satisfatória pelo que ele teve de ser eliminado do processo de identificação.

Ficamos portanto com apenas 7 parâmetros de identificação (máximo):

$$sumZ2, Z[0], Z[1], Z[3], nr[0], nr[1], nr[3]$$

a partir dos quais se obtém o σ global.

O valor a ser comparado com ele deverá então ser o somatório dos erros relativos medidos apenas naqueles parâmetros (do objecto relativamente ao modelo a ser considerado) multiplicados por 100. Prevendo que os parâmetros tenham sido medidos numa circunstância desfavorável convencionou-se que aquela soma não poderá exceder $1,5 \times \sigma$:

$$100 \times \frac{\Delta d}{\text{mod}} \leq \sigma \times 1,5$$

A comparação com os valores globais (em oposição à comparação individual dos parâmetros) contempla a possibilidade de um dos parâmetros conter um erro de medida que exceda aquele factor de 1,5; no entanto deve haver uma ou mais medidas com erros inferiores àquele factor para compensar devidamente o erro na 1ª medida.

Deverá a comparação individual ser excluída? Decerto que não. Se qualquer uma das medidas exceder um determinado erro relativo, isso significará que num parâmetro em particular o objecto não corresponde ao modelo (mesmo que os outros parâmetros se equivalem) o que deverá implicar a rejeição desse modelo. Teremos portanto que definir um limite de aceitação:

$$100 \times \frac{\Delta x}{x} \leq 2,0 \times \sigma$$

Este critério de rejeição deverá ser aplicado a toda e qualquer comparação individual de parâmetros.

Um último critério de rejeição a ser considerado é o nº de picos do histograma 3D, ou seja o nº de faces planas encontradas. Se este nº for diferente do objecto para o modelo, então a comparação deve cessar de imediato. Este é o 1º critério a utilizar no processo de comparação.

Se todos os 3 critérios forem obedecidos, então o objecto é considerado identificado e a referência do modelo é registada, acompanhada do respectivo coeficiente de "matching". No final das comparações, as referências aos dois melhores modelos encontrados assim como os respectivos coeficiente de "matching" são copiados para a estrutura do objecto 3D e a flag *'identified'* é posta a 1.

No Anexo J.1 é apresentado um esquema das chamadas efectuadas pelo programa principal de identificação a 3D.

12.2 Descrição das Várias Funções Envolvidas

12.2.1 Leitura do Ficheiro em Disco

Nesta função, o ficheiro de modelos 3D registado em modo binário no disco, é integralmente lido para memória RAM. Nesse sentido é previamente reservada memória dinâmica que possa conter todo aquele ficheiro. A falta de espaço em memória faz parar de imediato a execução do algoritmo de identificação.

A decisão de ler o ficheiro integralmente em vez de ler os modelos individualmente à medida que iam sendo necessários, deveu-se ao facto do acesso ao disco ser lento comparativamente com o acesso à RAM. E para além disso, todos os modelos são acedidos para qualquer identificação a efectuar no sentido de escolher os dois melhores, pelo que a sua leitura individual do disco seria bastante consumidora de tempo. Sendo assim, todo o ficheiro é lido de uma só vez o que reduz substancialmente o tempo gasto em acesso ao disco.

12.2.2 Obtenção da Informação do Modelo

Nesta função efectua-se a leitura da memória RAM dos valores relativos a um dado modelo, identificado pelo seu índice.

A primeira leitura efectuada é a da "identidade", que consiste no nome do modelo e no número de estados estáveis que o objecto possui. Consoante aquele número n de estados estáveis, são lidas n estruturas de "Assinatura" 3D. Uma vez preenchida a variável que contém o modelo, fica-se em condições de efectuar a comparação com a "Assinatura" previamente medida do objecto a identificar.

12.2.3 Comparação da Informação do Modelo com a do Objecto

Nesta função é feita a comparação dos parâmetros correspondentes do objecto e do modelo considerado, e calculado o respectivo coeficiente de "matching" que tomará valores entre 0 e 100. De todos aqueles apenas não é considerado o parâmetro $sumNr2$ por não possuir credibilidade (ver 10.3). Restam então entre 1 e $(1 + 2 \times nr_peaks)$ parâmetros a ser utilizados para comparação.

O cálculo do coeficiente baseia-se na medida da distância Euclidiana entre objecto e modelo no espaço de dimensões $(1 + 2 \times nr_peaks)$, ou seja, em $R^{(1 + 2 \times nr_peaks)}$. Se cada parâmetro definir uma dimensão, então o conjunto de parâmetros identificadores associado ao modelo, definirá um vector nesse espaço. Da mesma forma poderá ser definido um outro vector associado ao objecto. Uma boa medida da proximidade deste em relação ao modelo considerado é a razão entre os módulos dos vectores diferença e modelo que tomará sempre valores entre 0 e 1. O coeficiente de "matching" c poderá ser definido como o complementar deste valor para 1 multiplicado por 100, podendo então tomar valores entre 0 e 100.

Toda e qualquer comparação dará origem a um coeficiente c . Alguns valores, dentro de uma determinada gama, poderá no entanto significar que o modelo não é aceitável pelo que a sua referência não é inserida na informação do objecto mesmo que ele seja o melhor encontrado, nem a "flag" *identified* é posta a um. E tem interesse manter um modo imediato de consulta da situação de eventual identificação em vez de analisar sistematicamente o valor de c . O valor de fronteira entre os estados "identificado" e "não identificado" é convencional, mas tem uma forte dependência na tolerância que os parâmetros identificadores apresentam e que podem ser medidos. A definição dessa fronteira é calculada matematicamente de um modo também convencional como se verá de seguida. Por outro lado a utilização de certos critérios de rejeição imediata, permitem eliminar possibilidades poupando-se assim tempo precioso na execução do algoritmo. No caso da informação utilizada, o nº de faces planas paralelas ao plano horizontal parece ser um bom critério de rejeição imediata, enquanto que uma grande diferença encontrada para um dado parâmetro pode constituir um segundo critério de rejeição.

No caso de nenhuma daquelas situações ocorrer, deve-se no entanto efectuar ainda uma avaliação global das comparações entre todos os parâmetros correspondentes que constituirá o 3º e último critério de rejeição.

Os critérios 2 e 3 referidos serão então definidos como apresento a seguir.

A cada parâmetro é associada uma tolerância σ medida experimentalmente (ver 10.3) que é considerada primeiramente numa segunda verificação de eventual rejeição do modelo. Nesta acção, cada parâmetro é verificado de *per si*. Concretamente, se a distância "escalar" entre dois determinados parâmetros relativos ao objecto e ao modelo exceder $2 \times \sigma$, então os cálculos são imediatamente interrompidos e a função retorna o valor 0 significando que o modelo foi completamente rejeitado. Este valor $2 \times \sigma$ é convencional e pode ser alterado se os resultados experimentais assim o exigirem.

Se aquele valor não for excedido para nenhum parâmetro, então o modelo só poderá ser rejeitado após uma avaliação global dos desvios encontrados. Se os desvios

padrão σ referentes a $sumZ2$, $Z[i]$ e $nr[i]$ forem respectivamente σ_0 , σ_1 e σ_2 , então o desvio total máximo pode ser dado por:

$$\sigma = \sigma_0 + nr_peaks \times (\sigma_1 + \sigma_2)$$

De acordo com a secção referente à fiabilidade dos parâmetros identificadores (10.3),

$$\begin{cases} \sigma_0 = 6\% \\ \sigma_1 = 3\% \\ \sigma_2 = 5\% \end{cases}$$

pelo que o valor máximo de σ variará entre 6% e $6+3 \times (3+5)=30\%$. Este valor σ terá que ser avaliado para cada objecto a ser identificado (pois ele depende do nº de picos). Ele terá que ser comparado com o somatório de todos os desvios relativos encontrados na comparação, ou seja:

$$desv_rel = desv_rel(sumZ2) + \sum_i[desv_rel(Z[i])] + \sum_i[desv_rel(nr[i])]$$

O critério de rejeição poderá então ser:

$$desv_rel \geq 1,5 \times \sigma$$

(com "desv_rel" dado em percentagem). Se aquela condição não se verificar, então o modelo pode ser considerado e é calculado um coeficiente de "matching" c dado por:

$$c = (100 - desv_rel)$$

Este coeficiente será posteriormente comparado com os associados àqueles que são considerados os 2 melhores modelos até então encontrados. O modelo tomará o lugar de qualquer um daqueles se o c calculado for superior a um deles.

De referir ainda que aquele σ não corresponde a nenhum erro real de medida: se o nº de parâmetros identificadores fosse elevado, ele poderia aproximar-se, ou exceder os 100% sem que isso significasse imprecisão desses parâmetros; ele constitui tão somente um valor para comparação. Por exemplo, se para todos os parâmetros medidos os desvios relativos não excederem $1,5 \times$ dos valores correspondentes de desvio padrão, então o objecto é considerado identificado qualquer que seja o valor de σ .

12.2.4 A Procura do Melhor Modelo

Nesta função de procura do melhor modelo, são lidos em sequência todos os modelos do registo em RAM, para uma variável com a estrutura adequada.

Por cada modelo considerado é efectuada uma comparação com a informação medida do objecto em questão, de que resulta um coeficiente (de "matching") entre 0 e 100. Entretanto são registadas numa estrutura adequada as referências aos dois melhores modelos até então encontrados.

No Anexo J.2 apresenta-se uma descrição esquemática desta função.

Na Fig.3 encontra-se representado todo o processo de procura do melhor modelo: para cada modelo i registado na base de dados é lida a sua informação para estruturas de dados compreensíveis pelo algoritmo que após comparadas com as suas equivalentes do objecto a identificar, dá origem a um coeficiente de comparação c . Os dois maiores coeficientes c permitem obter a identificação dos 2 melhores modelos encontrados.

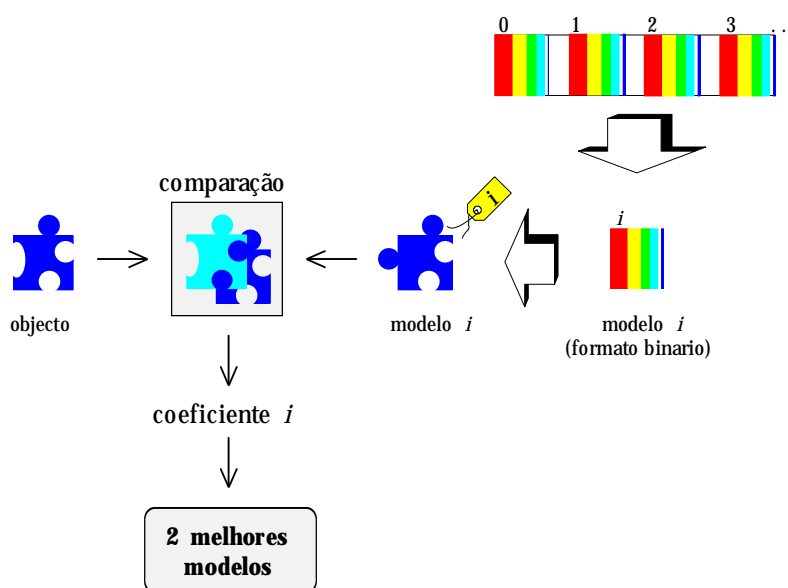


Fig. 3 - Ciclo de procura dos dois melhores modelos.

O procedimento de actualização desses 2 melhores modelos é o que se apresenta de seguida. Por cada comparação efectuada com um modelo, são confrontados ambos os valores $coef[i]$ daquela estrutura com o coeficiente acabado de calcular. Se este for superior àqueles dois, então o novo modelo é considerado como o 1º melhor sendo todos os campos com índice 0 actualizados com a nova informação. O 2º melhor modelo passa então a ser o que anteriormente era considerado como 1º, enquanto que o anteriormente considerado como 2º é ignorado (Fig.4).

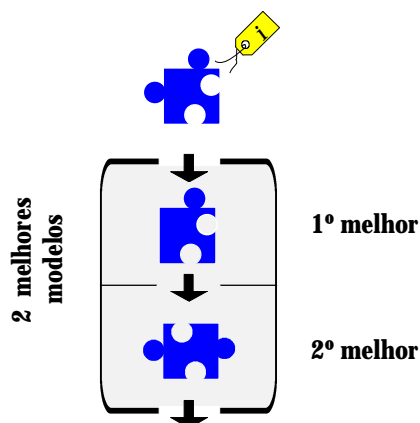


Fig. 4 - Procedimento quando é encontrado um modelo melhor que o n°1: o 1º é substituído pelo novo modelo e toma o lugar do 2º.

Se for superior ao 2º mas inferior ao 1º, então o 1º melhor modelo é mantido enquanto que o 2º é substituído pelo modelo considerado (Fig.5).

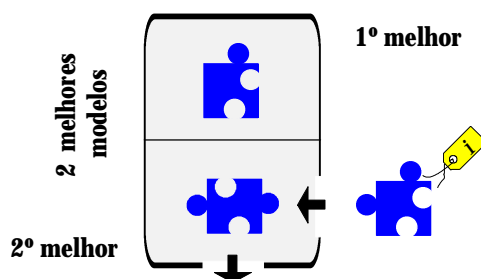


Fig. 5 - Procedimento quando é encontrado um modelo melhor que o n°2: apenas o 2º modelo é substituído.

No final, obtém-se então um registo com a referência aos dois melhores modelos que é copiada para a estrutura do objecto a identificar. De salientar que apenas as referências aos modelos (e não as "Assinaturas") são copiadas para essa estrutura que já contém a "Assinatura" do objecto.

Poder-se-ia pôr a questão de se efectuar uma procura indexada, por exemplo em *sumZ2*. Dessa forma apenas algumas comparações seriam efectuadas tornando o processo de identificação ligeiramente mais rápido. Contudo, para uma base de dados de dimensão realista, o varrimento de toda ela em RAM é extremamente rápida e em termos humanos, é imperceptível, pelo que decidi mantê-la. Por outro lado, para aquela indexação, apenas alguns valores de *sumZ2* seriam verificados em torno do valor encontrado para o objecto. Mas o parâmetro *sumZ2*, embora normalmente preciso (ver 10.3), poderia por qualquer razão ter sido mal medido ao contrário de outros, pelo que o modelo que corresponderia ao objecto poderia ser erradamente excluído do conjunto a considerar para comparação. Sendo assim, abandonou-se definitivamente a possibilidade de efectuar uma procura indexada, preferindo-se manter o varrimento de todo o ficheiro de modelos.

12.2.5 A Função Principal de Identificação a 3D

Nesta função é inicialmente ordenada a leitura do ficheiro de modelos integralmente para memória RAM.

Ela permite obter a identificação de qualquer objecto contido num vector através da indicação do respectivo índice nr , ou de todos eles se este for igual a zero. Em qualquer destas duas opções, para um dado objecto considerado, é ordenada a procura dos dois melhores modelos através da função descrita em 12.2.4. após o que as suas referências são copiadas para os respectivos campos da estrutura referente ao objecto 3D. De salientar que apenas elas são copiadas para a estrutura do objecto, não sendo portanto copiada a informação integral das respectivas "Assinaturas" (Fig.6).

12.3 Descrição Esquemática do Programa de Identificação

Inclui-se neste texto um esquema das acções efectuadas pelo algoritmo principal de identificação a 3D para ilustrar o que foi descrito nas secções anteriores (Fig.7).

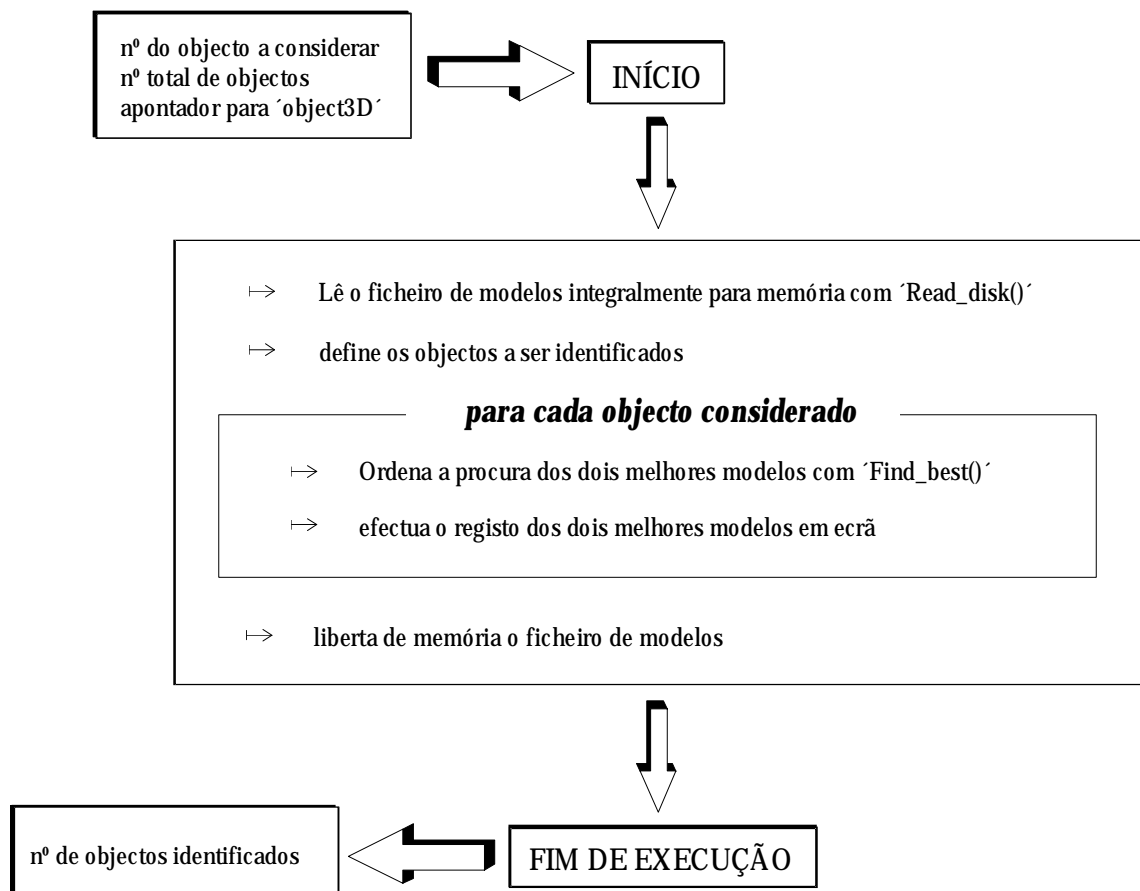


Fig. 7 - Descrição esquemática da função principal de identificação a 3D.

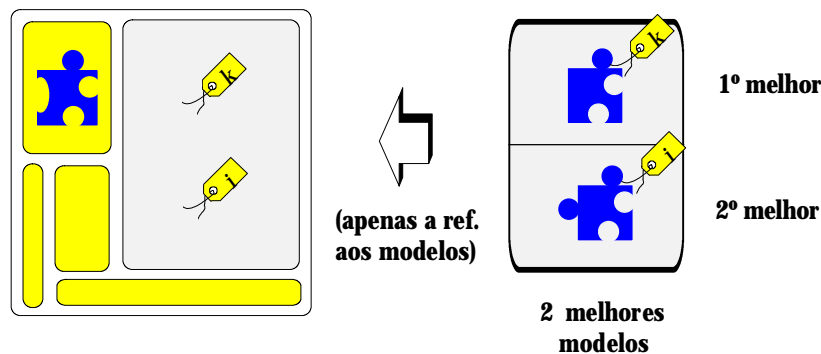


Fig. 6 - Registo final dos dois melhores modelos encontrados na base de dados. (É apenas feita referência aos 2 melhores modelos; não é inserida a informação completa das respectivas "Assinaturas").

12.4 Conclusão

Para a construção deste algoritmo de identificação a 3D foi aproveitada a estrutura anteriormente desenvolvida para o seu equivalente a 2D e daí o nome em comum da maior parte das funções que a integram. Por esse motivo não foi apresentada a sua descrição em detalhe.

O processo de "matching" utilizado neste programa é, contudo, diferente do anteriormente adoptado no algoritmo a 2D e daí a sua descrição já com algum detalhe.

Com a apresentação deste novo processo, pretende-se demonstrar que o modo de se obter uma identificação e portanto de medir os coeficientes de comparação não é único. A medida da sua aplicabilidade deve ser antes dada pelo tipo de sucesso obtido em experiências controladas de identificação.

Algumas experiências deverão ainda no futuro ser levadas a cabo de modo a ajustar os valores de fiabilidade dos parâmetros identificadores para um universo mais alargado de objectos; Do mesmo modo deverá ser avaliada a eficácia dos valores utilizados nos critérios de rejeição. Todavia, os valores que foram utilizados, permitiram já atingir níveis elevados de sucesso na identificação a 3D, o que pelo menos demonstra a eficácia da abordagem utilizada.

CONCLUSÕES E TRABALHO FUTURO

1. Conclusões Gerais

Em face de todos os resultados obtidos nas várias experiências e testes levados a cabo com os algoritmos desenvolvidos, é de concluir que se conseguiu de facto construir um sistema de Visão robusto para localização e identificação de objectos.

Embora ele seja apenas parte de um sistema robótico mais amplo que inclui um planeador de tarefas de alto nível e de executor de baixo nível associado ao próprio robot, a sua eficiência é essencial para que aquele possa estruturar convenientemente o estado inicial do "mundo" e dar o seguimento adequado a uma qualquer tarefa de montagem.

Apesar do grande domínio de aplicação dos algoritmos, devo nesta parte final fazer uma breve referência às suas limitações. A que mais se salienta não só no caso dos algoritmos a 2D como nos a 3D, é a obrigatoriedade dos objectos se encontrarem num dos seus estados estáveis para que a identificação seja possível. No caso do algoritmo 3D a rotação no espaço, isto não é forçosamente um impedimento à identificação, mas requer uma manipulação robótica intercalar: uma vez que se possui a informação sobre a 3ª dimensão, é possível calcular parâmetros que permitam a remoção segura do objecto rodado no espaço e a colocação em local mais favorável à sua identificação. No caso da análise a 2D, qualquer rotação no espaço altera a perspectiva do objecto e consequentemente as características geométricas que são a base da identificação.

Os melhoramentos que entretanto se possam vir a introduzir neste conjunto de algoritmos assim como os novos que venham a ser acrescentados, tal como referirei mais adiante, poderão contudo levantar aquelas limitações tornando assim o sistema aplicável a cenários robóticos mais complexos.

2. Discussão Final sobre a Abordagem Adoptada no Algoritmo de Inspecção a 3D

A questão que se põe a respeito desta abordagem 3D é se realmente se adequa a situações reais de manipulação robótica. É um facto que muitas peças a manipular possuem planos paralelos a XOY quando se encontram num dos seus estados estáveis e

dessa forma o processamento é eficaz (conforme o demonstra as experiências efectuadas e apresentadas neste texto. A grande vantagem deste processo é não só o modo simples de representar modelos mas também a rápida execução dos algoritmos de localização e identificação.

Outras abordagens, como as que se apresentaram no início deste capítulo, consideram de facto várias perspectivas 3D e aparentemente apresentam maior probabilidade de sucesso na identificação. A rotação da peça no espaço devido a sobreposição pode ser frequente (consoante as situações de aplicação). Contudo, este processo sofisticado a representação do modelo. Como consequência destes dois factos, o processo de "matching" torna-se complexo e como tal fica sujeito a mais erros. Por outro lado, a própria necessidade do algoritmo detectar e medir as várias características, tipo arestas, planos e vértices e as relações de ligação entre eles (por utilização de grafos, com medição adicional de ângulos entre arestas ou planos) vem aumentar drasticamente o tempo necessário para a execução do algoritmo que se pode tornar excessivo para uma aplicação prática de robótica de manipulação. Para além disso, por muito complexa e aparentemente completa que a análise seja, existem sempre situações e objectos que não são contemplados por ela, ou seja, que escapam ao seu domínio de aplicação. E sendo assim, tudo depende da situação em particular à qual se pretenda aplicar o algoritmo de Visão, podendo a mais completa das abordagens mostrar-se inadequada enquanto que outra muito mais simples e rápida se mostre perfeitamente adequada.

Nesta abordagem, o algoritmo mostra-se particularmente eficiente para objectos com faces planas paralelas a XOY, estando estes num dos seus estados estáveis como já se anunciou acima. Considera-se também que situações de sobreposição são esporádicas não contemplando por isso casos de empilhamento de peças em recipientes ("bin picking"). A resolução dessas situações vem portanto simplificada: dado que a ocorrência de oclusão parcial vem dificultar senão impossibilitar a identificação e medida da posição do objecto parcialmente oculto e atendendo a que este pode ser um dos elementos a utilizar na montagem, o algoritmo 3D efectua as medições necessárias para a manipulação do objecto sobreposto e envia os dados necessários ao planeador de alto nível. O objectivo é remover de imediato todos os objectos que se encontrem em situação de oclusão e colocá-los em posições livres do espaço de acção do robot, numa das suas posições estáveis.

O passo seguinte seria uma nova análise da cena numa situação bem mais favorável que a anterior dado que já não existiriam sobreposições.

E em casos de objectos um pouco mais complexos que os que constituem o domínio descrito de aplicação? O algoritmo decerto falharia a sua análise. Contudo, a prévia remoção dos objectos sobrepostos não apresenta só a vantagem da sua rápida execução e de redução de erros de medida de posição e identificação devido à rotação no espaço! É que a partir do momento em que os objectos se encontrem separados não só é possível correr de novo o algoritmo 3D, mas também fazer entrar em acção o algoritmo de Visão a 2D. Este faz levantar de imediato a restrição imposta dos objectos possuírem planos paralelos a XOY, atendendo a que este algoritmo suporta qualquer grau de complexidade do objecto pois mede as suas características geométricas!

Resta agora fazer referência ao modo como o algoritmo aborda a questão dos objectos em sobreposição (objecto de cima) dado que não apresentam nenhum plano horizontal (embora se mantenha a condição de que o possui quando se encontra num dos

seus estados estáveis) devido a rotação no espaço. Nesta circunstância são efectuadas medidas de altura, assim como dos ângulos de rotação em relação a OX e de inclinação em relação a XOY, cujos valores são registados nas respectivas "Assinaturas" ou em vectores auxiliares (caso deste último) para posterior uso. São estes valores que são enviados ao planeador de alto nível para remoção do objecto.

Mais uma vez, como se pode constatar, sacrificou-se a complexidade de análise a 3D em favor da rapidez e facilidade de execução o que torna esta abordagem significativamente mais eficiente.

3. Trabalho Futuro

Embora o sistema de Visão construído esteja perfeitamente operacional, existem alguns melhoramentos que poderão vir a aumentar a sua eficácia. Nesta secção fazer-se-á referência não só a esses melhoramentos pontuais como também ao desenvolvimento no futuro de outros algoritmos complementares que poderão alargar a gama de objectos processáveis pelo sistema e situações do estado inicial do "mundo".

3.1 Algoritmo de Sequenciação

No capítulo referente à sequenciação de segmentos foi apresentado um algoritmo alternativo de identificação de objectos. Uma vez que se encontravam já operacionais dois sistemas de identificação, não foi construída o respectivo ficheiro de modelos nem efectuadas experiências, embora o grande sucesso na detecção e medida daqueles segmentos provasse o seu grande potencial. Sendo assim, seria desejável a finalização do algoritmo e consequente activação deste processo de identificação

Como complementarização, prevê-se ainda a extensão desta abordagem a outras figuras geométricas, nomeadamente a arcos de circunferência e de elipse, após o que muitos mais tipos de objectos seriam identificáveis pelo algoritmo.

3.2 Sensores de "Range"

Como ficou visto, o sistema de varrimento laser é bastante preciso e eficiente: ele fornece amostragens de altura em pontos regularmente espaçados o que no caso de ser usada a resolução máxima permite obter imagens de altura de boa qualidade. A sua única falha reside na impossibilidade de efectuar aquelas amostragens em zonas de "sombra" e daí a falta de informação associada a objectos parcialmente ocultos. Futuramente poderá ser instalado um segundo reflector motorizado (associado a um segundo motor passo-a-passo) que permitisse dirigir o feixe para essas zonas. As duas amostragens obtidas com cada um dos dispositivos dariam então origem a uma só, libertada das referidas "sombras".

3.3 Imagens de "Range"

Um desenvolvimento que se prevê de grande interesse é a aplicação da Transformada de Hough Melhorada na detecção e parametrização de planos na imagem de alturas. Seria então utilizada uma tabela de acumulação tridimensional correspondente ao espaço discretizado dos três parâmetros que definem um plano. Sendo assim, e se um algoritmo equivalente de pós-processamento para selecção e fusão de sub-células pudesse ser construído, então um qualquer objecto poliédrico poderia já ser identificado através da análise das suas superfícies visíveis qualquer que fosse a sua orientação no espaço. Ou seja, a exigência das superfícies serem paralelas ao plano horizontal seria levantada. Mais ainda, deixaria de ser exigível que o objecto se encontrasse num dos seus estados estáveis, aumentando assim significativamente o leque de situações em que seria possível identificar os objectos numa cena.

Anexos

Anexo A

A.1 O Método Usado na Inversão da Matriz de Transformação A

Para se obter a matriz inversa de A , quer ela represente a matriz transformação, quer a matriz de $2N \times 11$ de calibração da câmara, foi utilizado o método de Gauss-Jordan.

Tendo em conta que $AA^{-1}=I$, começa-se por formar uma matriz única composta quer por A quer por I :

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & 1 & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 & 1 & 0 \\ a_{31} & a_{32} & a_{33} & 0 & 0 & 1 \end{bmatrix}$$

Aplicam-se agora coeficientes a estas linhas por forma a poder-se aplicar a eliminação sucessiva de variáveis pela soma membro a membro, até que se obtenha para A uma matriz triangular. Todas as multiplicações e somas membro a membro são aplicadas também a I . Nesta fase a matriz global terá a seguinte forma:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & e_{11}' & 0 & 0 \\ 0 & a_{22}' & a_{23}' & e_{21}' & e_{22}' & 0 \\ 0 & 0 & a_{33}' & e_{31}' & e_{32}' & e_{33}' \end{bmatrix}$$

Nesta forma (triângulo superior), quando aplicada a um 2º membro, permite obter a solução $x = (x_1, \dots, x_n)$ por substituições sucessivas desde a última até à primeira equação.

Numa segunda fase criam-se coeficientes nulos também acima dos elementos pivot até se obter a matriz identidade I (a parte direita da matriz global fica sujeita às mesmas operações efectuadas na primeira):

$$\begin{bmatrix} 1 & 0 & 0 & e_{11}'' & e_{12}'' & e_{12}'' \\ 0 & 1 & 0 & e_{21}'' & e_{22}'' & e_{23}'' \\ 0 & 0 & 1 & e_{31}'' & e_{32}'' & e_{33}'' \end{bmatrix}$$

Ora, esta série de operações que transformam A em I , sendo também aplicadas a I (parte direita da matriz global) transformam-na obrigatoriamente em A^{-1} ! Obtém-se então o resultado final:

$$A^{-1} = \begin{bmatrix} e_{11}'' & e_{12}'' & e_{12}'' \\ e_{21}'' & e_{22}'' & e_{23}'' \\ e_{31}'' & e_{32}'' & e_{33}'' \end{bmatrix}$$

O conhecimento da matriz inversa de A no caso desta ser a matriz transformação, pode de facto permitir conhecer (X,Y) a partir de (x,y, Z) , embora isso implique o arbítrio da 3ª linha daquela (ver 1.3.9). Contudo isso envolve todo o processo de cálculo atrás apresentado. Ele pode ser evitado se se adoptar um outro meio de cálculo que para além de dispensar o arbítrio da 3ª linha, utiliza directamente os 12 elementos conhecidos da matriz A e que apresento em 1.3.9.1.

No caso da matriz de calibração, o cálculo do inverso de A é obviamente inevitável.

A.2 A Descrição do Software para Calibração da Câmara

A.2.1 Os Programas

Com base no desenvolvimento teórico apresentado atrás, foram escritos vários programas para concretizar os dois grandes objectivos deste capítulo: o cálculo da matriz A de transformação a partir da aquisição experimental das coordenadas de pontos de calibração e a conversão de coordenadas a partir daquela matriz. Os programas são os que constam da lista seguinte:

files-c

CAMERA_M	C
CAMERA	C
POINTS	C
BLOB2	C
ALLOC_P	C
REORDER	C
MAKE_AIJ	C
MAKE_BJ	C
BEST_SOL	C
INVERT	C
LUT-2	C

READ_M	C
COMP_XY	C
BACK_XY	C
T-1	C
T-2	C
T-3	C
T-4	C
T-5	C
T-6	C
T-7	C

A.2.2 Esquema de Chamadas do Programa de Calibração

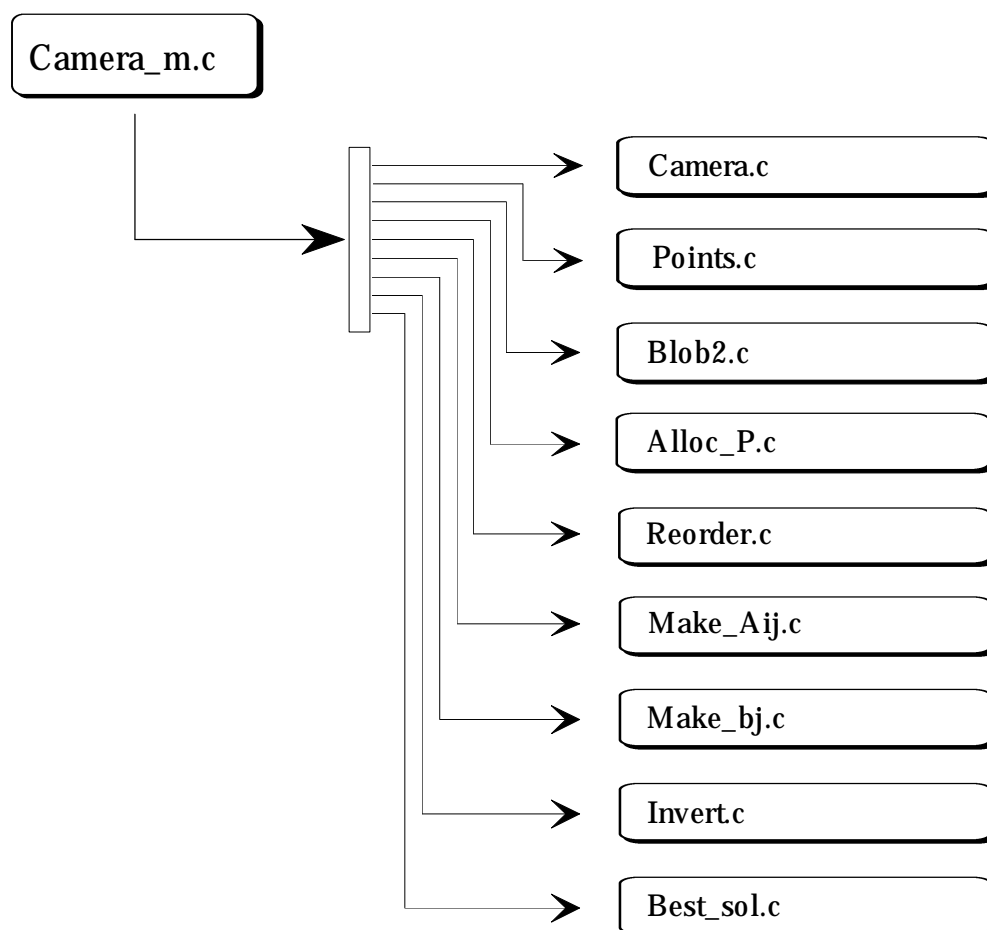


Fig. 1 - Esquema de chamadas do programa de calibração.

A.2.3 Descrição das Várias Funções

1. *Camera_m.c*: chama '*Camera.c*' e regista a matriz A em disco.
2. *Camera.c*: programa principal de calibração da câmara; efectua todas as aquisições de valores experimentais, toda a manipulação matemática e calcula a matriz de transformação A .
3. *Points.c*: programa de aquisição das coordenadas (x,y, Z) ; ele chama o programa anterior tantas vezes quantas as alturas a que se irá colocar a grelha; após cada aquisição verifica o nº correcto de pontos de calibração e reordena os pontos para atribuição automática de valores (X,Y) ; funde todos os conjuntos obtidos de valores num só vector para posterior processamento; aceita imagens dada pela câmara ou

registadas em disco (o que permite uma calibração *a posteriori* após adquiridas as imagens de todas as alturas pretendidas para a grelha facilitando assim todo o processo de calibração; na opção de aquisição de imagens, esta é feita após selecção de uma Lut de entrada (tabela de conversão de níveis de cinzento, programada por acção directa nos registos da placa de aquisição de imagem) que as binariza em tempo real; a tabela de conversão linear (ausência de alteração nos níveis de cinzento) é sempre repostada após a execução deste programa.

4. *Blob2.c*: adaptação de '*Blob.c*' para a aquisição das coordenadas na imagem de todos os pontos pertencentes às grelhas utilizadas para calibração.
5. *Alloc_P.c*: efectua a selecção das figuras encontradas em '*Blob2.c*' e regista-as em memória Ram.
6. *Reorder.c*: reordena todos os pontos de calibração de um determinado plano (associado a uma dada altura da grelha) da esquerda para a direita e de cima para baixo, por forma a atribuir-lhes automaticamente os respectivos valores, pré-conhecidos, de (X,Y) .
7. *Make_Aij.c*: constrói a matriz A_{ij} com dimensão $2N \times 11$, sendo N o nº total de pontos de calibração, com base nos respectivos valores de (x,y, Z) .
8. *Make_bj.c*: constrói o vector bj com dimensão $2N$ com base nas coordenadas (x,y) de todos os pontos de calibração e que vão constituir o 2º membro da operação matricial $A.x=b$.
9. *Best_sol.c*: este programa chamado por *Camera.c* recebe a matriz A_{ij} e o vector bj e efectua todas as chamadas de função necessárias para efectuar a operação $\bar{x} = (A^T A)^{-1} A^T b$, ou seja o cálculo da melhor solução para o sistema sobredimensionado.
10. *Invert.c*: programa de inversão de matrizes usado para o cálculo de $(A^T A)^{-1}$; aceita o apontador para o vector contendo o resultado da operação matricial $(A^T A)$ assim como a dimensão desta matriz quadrada (embora se saiba que no caso da calibração se trata de 11×11 , este programa permite a inversão de qualquer matriz, tornando-o utilizável noutras aplicações).
11. *Comp_XY.c*: calcula (X,Y) em função de (x,y, Z) com base na operação com a matriz de 2×2 obtida em função dos coeficientes a_{ij} e naquelas coordenadas; cada ponto de calibração implica pois uma matriz diferente.

12. *Back_xy.c*: calcula de volta as coordenadas (x,y) em função de (X,Y,Z) pela operação matricial $(A.w_h)$.

A.2.4 Descrição dos Programas de Teste "*T-?.c*"

Foram ainda escritos vários programas de teste.

1. *T-1.c*: define a matriz A e o vector 2º membro b , e chama '*Best_sol.c*' para calcular a melhor solução x do sistema sobredimensionado; pretende-se com este programa testar a eficácia do algoritmo partindo de uma solução x pré-conhecida e de um segundo membro numericamente próximo do resultado que ela daria.

Partindo de um vector $x=(1,2,3)$, vou construir uma matriz A sobredimensionada à qual vou atribuir um 2º membro composto pelos resultados da aplicação de cada linha de coeficientes à solução pré-conhecida x , ligeiramente desviados. Por exemplo, para uma 1ª linha composta por $(5,-1,3)$, teria um resultado de 12:

$$5x_1 - x_2 + 3x_3 = 12$$

Em vez dele introduzo por exemplo 11,5. Para outras linhas, calculo o respectivo valor de 2º membro e altero-o ligeiramente. Como resultado da aplicação de *Best_sol.c* devo obter a solução x mais aproximada ao vector 2º membro construído.

Considere-se por exemplo a seguinte matriz A e o respectivo 2º membro construído da forma indicada acima:

$$A = \begin{bmatrix} 5 & -1 & 3 \\ 7 & 2 & -4 \\ 2 & 3 & -9 \\ -1 & -3 & 6 \\ 3 & -2 & 2 \\ 4 & 2 & -1 \\ 9 & -7 & 1 \\ 6 & 1 & -2 \\ 2 & 4 & 9 \\ 2 & -3 & 2 \\ 7 & -2 & -1 \\ 1 & 3 & -2 \\ 3 & 1 & -2 \\ 5 & 2 & -3 \\ 8 & 5 & -3 \\ 9 & 3 & -5 \\ 4 & -1 & -2 \end{bmatrix} \quad b = \begin{bmatrix} 11,5 \\ -1,2 \\ -19,5 \\ 10,4 \\ 5,1 \\ 4,8 \\ -2,3 \\ 2,1 \\ 36,1 \\ 2,3 \\ 0,2 \\ 1,8 \\ -1,3 \\ -0,1 \\ 8,7 \\ 0,3 \\ -4,1 \end{bmatrix}$$

Ora, por aplicação de '*Best_sol.c*' chamado por '*T-1.c*' obtém-se o vector solução:

$$x = (0,973554; 1,970639; 2,951088)$$

que é de facto muito próximo do vector x considerado para construir b .

2. *T-2.c*: lê os conteúdos dos ficheiros em disco *A.dat* e *b.dat* correspondentes ao registo da matriz A e do vector 2º membro b , para verificação dos valores neles contidos.
3. *T-3.c*: lê o ficheiro *matrix_M.dat* que contém os 12 coeficientes relevantes da matriz A de transformação; chama *read_M()* de '*Read_M.c*';
4. *T-4.c*: 1ª tentativa de conversão de coordenadas, substituída por '*Camera.c*';
5. *T-5.c*: adquire as imagens da grelha nas suas várias alturas, calcula o limiar de binarização, programa uma Lut de entrada de acordo com esse valor de limiar, e binariza cada uma das imagens; inverte ainda as imagens para as poder submeter ao programa de análise de figuras geométricas, '*Blob2.c*'; este programa permite efectuar o registo em disco de uma série de imagens de uma só vez, para serem processadas posteriormente para a calibração da câmara, tornando todo o processo muito mais prático.
6. *T-6.c*: calcula coordenadas (X,Y) a partir de (x,y, Z) fazendo uma chamada a *compute_XY()* de '*Comp_XY.c*'; aquelas segundas coordenadas são introduzidas pelo utilizador.

Anexo B

B.1 Esquema de Chamadas para as Funções Principais e Secundárias na Análise Bidimensional

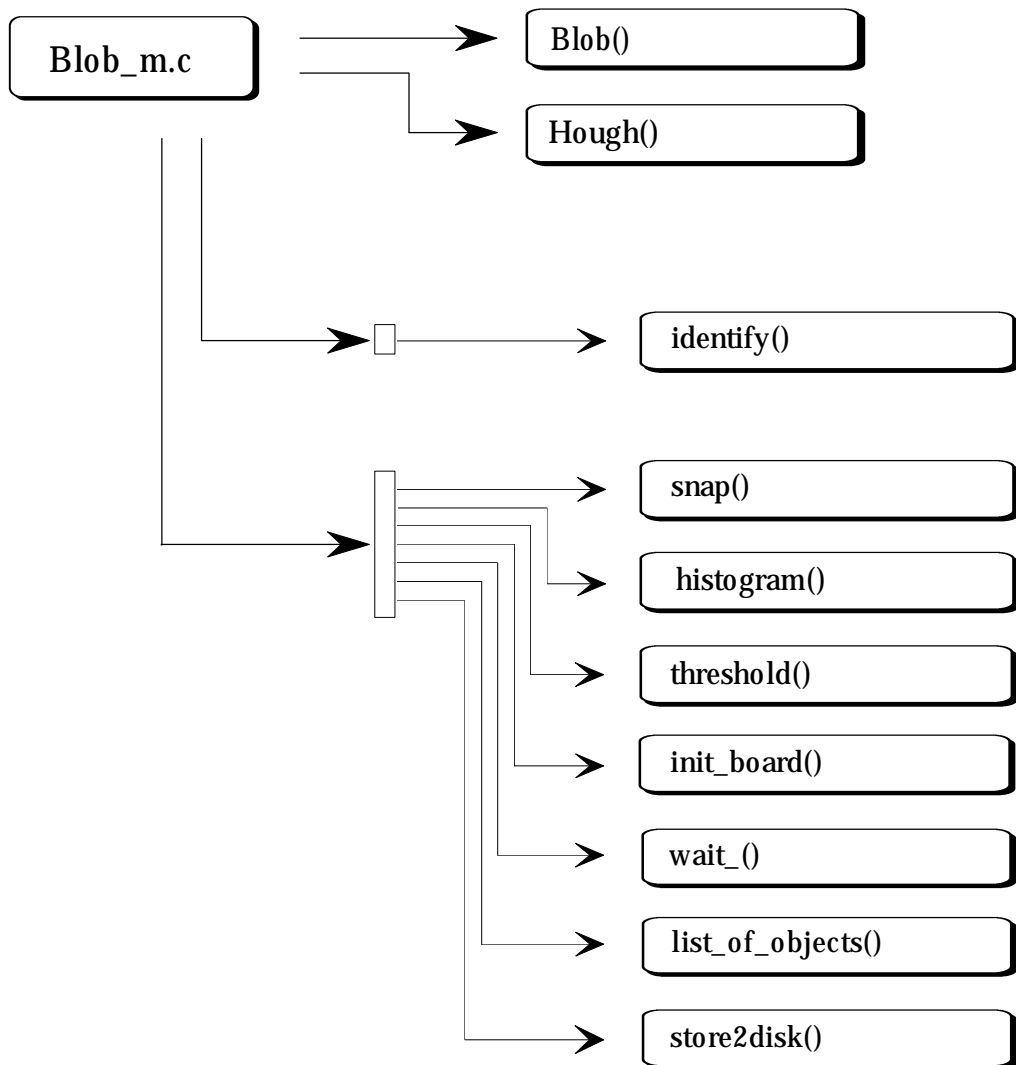


Fig. 1 - Esquema de chamadas no Agente Visão.

Anexo C

C.1

O Significado das Funções no Programa de Análise Geométrica a 2D

O programa desenvolvido para a extracção das características geométricas bidimensionais dos objectos foi designado por 'Blob.c' e na tabela abaixo são descritas as funções de que ele é composto.

Nome da função	Descrição
Blob()	programa principal
Blobscan()	programa de varrimento da imagem e medida dos parâmetros geométricos
RPIXEL()	função de leitura aleatória do conteúdo de pixels da imagem
RHLINE()	função de leitura de uma linha integral de pixels da imagem
ZeroItAll()	inicialização de todas as variáveis globais
Getlin()	função de aquisição de uma linha de pixels com transformação destes em 0's ou 1's
Update()	função de actualização dos parâmetros geométricos de uma figura ou "blob"
State i ()	uma de 6 funções chamada sempre que se detecte uma transição na janela de 2×2 que varre a imagem; consoante a combinação de pixels 0 ou 1 na janela, assim se identifica a situação e se chama a função adequada que altera o conteúdo de variáveis de controle ou efectua actualização de valores.
Allocate()	acréscimo de mais um "blob" com inicialização de alguns campos.
Iniblob()	inicialização dos campos dos novo "blob"; é chamado pela função anterior.
Lookup()	consulta do nº do "blob" válido a que o pixel pertence; esta verificação é necessária dada a constante fusão de "blobs" que invalida sempre um dos intervenientes; o vector $BP[]$ mantém sempre informação dos "blobs" que absorveram outros.

Anexo D

D.1 Esquema de Chamadas no Software Desenvolvido para a Transformada de Hough Modificada com Pós-Processamento

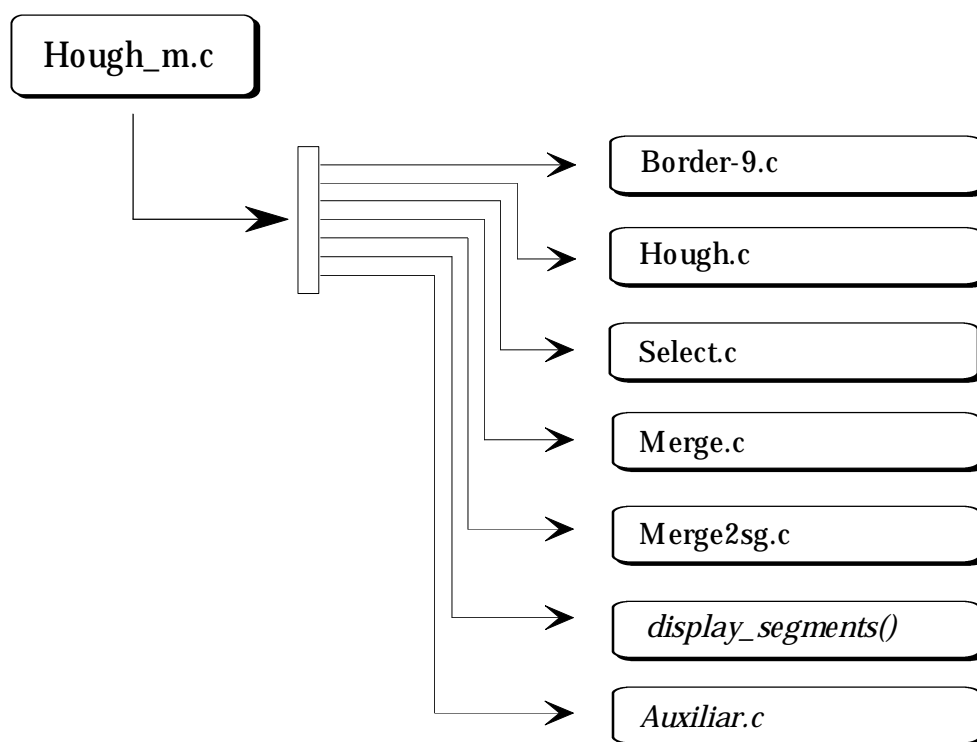


Fig. 1 - Esquema geral de chamadas no algoritmo de Hough modificado com pós-processamento. De cada vez que é finalizada uma acção, é chamada a função *display_segments()* que ilustra o seu efeito.

D.2 Descrição das Funções

- *Hough_m.c*: Esta é a função principal que efectua as chamadas necessárias a todos os algoritmos utilizados, nomeadamente de procura de pontos de fronteira a ser sujeitos à medida do gradiente (*border()*), da análise pela transformada de Hough modificada (*hough()*) e construção da tabela de acumulação, da selecção de sub-células (*select_cells()*) e pela fusão destas em segmentos (*merge_segments()*).
- *Border-9.c*: Este é o programa que contém a função *border()* responsável pelo varrimento da imagem com uma máscara de 2×2 já descrita acima, com o objectivo de detectar pontos de fronteira. Quando tal acontece, são passados a *hough()* apontadores que permitem determinar o valor dos pixels (0 ou 1) a ser operados pelo conteúdo das máscaras *h* e *v* utilizadas. Uma vez que se pretende acompanhar o sucesso na detecção desses pontos, esses são postos a 255 (valor lógico 1) enquanto que os restantes são postos a 0. Esta acção de modificação do ficheiro de imagem implica, tal como acontece com todas as funções de filtragem por convolução, que ela seja feita em zonas de memória temporárias que registem as alterações a efectuar em cada linha, sendo elas copiadas para memória quando as posições correspondentes já não sejam necessárias para mais cálculos.
- *Hough.c*: Esta é a função que calcula as componentes horizontal e vertical do gradiente no ponto de fronteira, calcula os valores aproximados de ρ e θ da linha tangente à fronteira e incrementa a célula correspondente na tabela de acumulação. Para cada incremento verifica a distância do ponto considerado ao último ponto registado na célula e se o seu valor for superior ao permitido, considera um novo segmento na mesma célula através da criação de uma sub-célula ligada à anterior. O resultado final dos incrementos referentes a todos os pontos de fronteira é uma tabela de onde "partem" listas ligadas de sub-células, cada uma em princípio associada a um segmento distinto.
- *Select.c*: Nesta função, são seleccionadas todas as sub-células com um número de incrementos superior a um dado valor mínimo (suficientemente baixo para detectar segmentos pequenos) e inseridos numa lista ligada única.
- *Merge.c*: Esta função é responsável pela comparação, duas a duas, de todas as sub-células com o objectivo de as fundir e reconstruir os segmentos originais.

- *Merge2sg.c*: Este programa procede à verificação dos três critérios para a fusão de duas sub-células. Sempre que eles são verificados toda a informação nelas contida (com especial destaque para as coordenadas dos pontos extremos) é actualizada e registada na que irá prevalecer.
- *display_segments(struct sel_cell_*,int, char)*: Função de desenho dos segmentos correspondentes às sub-células seleccionadas que compõem a lista ligada. Aceita texto descritivo do resultado a observar que é registado na parte inferior do ecrã assim como um inteiro que controla o tempo de espera no desenho entre cada dois segmentos.
- *Auxiliar.c*: Este programa é na realidade um módulo que contém a maioria das funções auxiliares utilizadas que serão descritas em seguida.

C.2

Demonstração do Teorema de Steiner para Massas Pontuais

Considere-se um conjunto de pontos materiais de massa discreta m_i comum, dispostos no espaço tridimensional (embora os pixels se encontrem no mesmo plano, ficará provado que o resultado é válido para as 3 dimensões do espaço cartesiano).

O momento de inércia total do conjunto relativamente a um eixo E distanciado do centro de massa de R, é dado pela soma de todos os momentos de inércia discretos (ver Fig.).

No caso da imagem composta por pixels, $m_i=1$ e $m=area$ uma vez que nos interessa considerar momentos de inércia geométricos ($m=\rho V$, onde ρ é a massa volúmica no caso tridimensional, ou $m=\rho S$, onde ρ é a massa superficial no caso bidimensional; neste último caso, fica $m=S=area$ ignorando ρ para se obter o momento de inércia geométrico em vez do momento de inércia).

$$I = \sum_i m_i r_i^2 = \sum_i m_i (\vec{r}_i | \vec{r}_i) = \sum_i m_i (\vec{R} + \vec{r}_{oi}) | (\vec{R} + \vec{r}_{oi}) = \sum_i m_i R^2 + \sum_i m_i r_{oi}^2 + 2 \sum_i m_i (\vec{r}_{oi} | \vec{R})$$

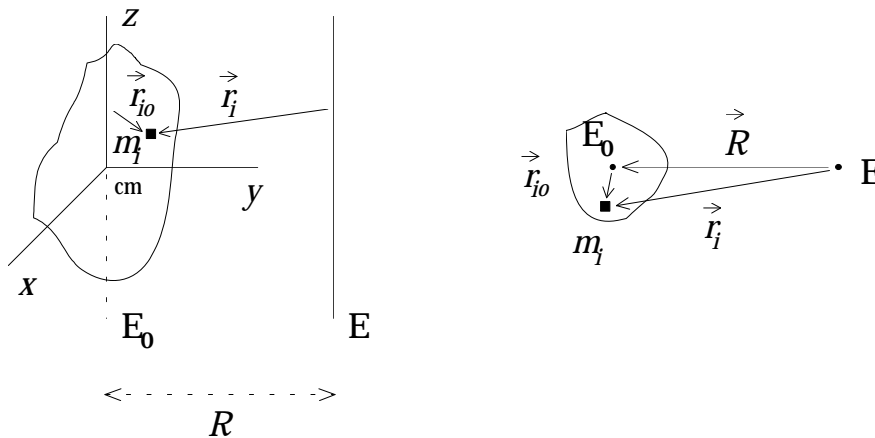


Fig. 1 - Momentos de inércia de uma massa discreta m_i relativamente a dois eixos paralelos, calculados a partir dos respectivos vectores de posição: r_i e r_{io} .

A 1ª parcela é simplesmente mR^2 enquanto que a 2ª é precisamente o momento de inércia geométrico calculado em relação ao eixo paralelo a E que passa pelo centro de massa, E_0 .

No caso da 3ª parcela, \vec{R} pode ser posto em evidência ficando então, $\sum_i (m_i \vec{r}_{oi}) | \vec{R}$. Ora, o 1º termo do produto interno é, a menos de m , a expressão que dá o vector de posição do centro de massa do sistema... relativamente ao sistema de coordenadas centrado precisamente no centro de massa! O resultado é portanto o vector nulo, pelo que restam apenas as duas parcelas já referidas:

$$I = \sum_i m_i R^2 + \sum_i m_i r_{oi}^2 = mR^2 + I_o \Leftrightarrow I = mR^2 + I_o$$

Ou seja, o momento de inércia de um conjunto de massas calculado relativamente a um eixo E qualquer, é igual ao momento de inércia das mesmas massas calculado relativamente a um outro eixo E_o paralelo a E que passe pelo centro de massa, somado ao produto da massa total m pela distância entre os dois eixos.

C.3 Demonstração do Resultado

$$XY = \text{sum}ij/area = \text{sum}ij/area - \langle x \rangle \langle y \rangle$$

XY	é valor médio do somatório do produto $(x_i \times y_i)$ calculado relativamente ao referencial que passa pelo centro de massa, estendido a todos os pontos i ; pode também ser escrito $\langle XY \rangle_o$.
$\text{sum}ij/area$	é o mesmo valor médio mas calculado relativamente ao referencial absoluto (x,y,z) .
$\langle x \rangle$	valor médio de x calculado no referencial absoluto (referencial da imagem).
$\langle y \rangle$	valor médio de y calculado no referencial absoluto (referencial da imagem).

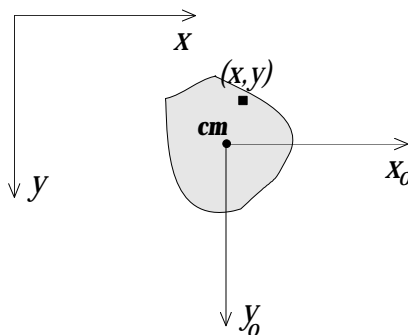


Fig. 2 - Representação do sistema de referência centrado no centro de massa do objecto.

Ora, tendo em conta que, $\begin{cases} x = x_{cm} + x_o \\ y = y_{cm} + y_o \end{cases}$ temos que,

$$\begin{aligned} \sum xy &= \sum [(x_{cm} + x_o)(y_{cm} + y_o)] = \sum x_{cm}y_{cm} + \sum x_o y_o + \sum x_{cm}y_o + \sum x_o y_{cm} = \\ &= x_{cm}y_{cm} \cdot N + \sum x_o y_o + x_{cm} \sum y_o + y_{cm} \sum x_o = x_{cm}y_{cm} \cdot area + \sum x_o y_o + x_{cm} \cdot 0 + y_{cm} \cdot 0 \end{aligned}$$

Os valores 0 devem-se ao facto dos somatórios se referirem às coordenadas do centro de massa no referencial do centro de massa. A equação fica então com a seguinte forma:

$$\begin{aligned} \sum xy &= x_{cm}y_{cm} \cdot area + \sum x_o y_o \Leftrightarrow \sum x_o y_o = \sum xy - x_{cm}y_{cm} \cdot area \Leftrightarrow \\ &\Leftrightarrow \frac{\sum x_o y_o}{area} = \frac{\sum xy}{area} - \langle x \rangle \langle y \rangle \end{aligned}$$

Ou seja, $\langle XY \rangle_o = \langle XY \rangle - \langle x \rangle \langle y \rangle$

Este resultado pode ser enunciado da seguinte forma:

O valor médio do produto $x \cdot y$ calculado relativamente ao referencial do centro de massa é igual ao mesmo valor médio calculado no referencial absoluto da imagem subtraído do produto das coordenadas do centro de massa.

Blob() e *Hough()* são os dois algoritmos fundamentais do agente Visão. A função *Identify()*, que se encontra destacada, destina-se a complementar aquele primeiro algoritmo com a identificação dos objectos por ele detectados.

São ainda utilizadas algumas funções auxiliares tais como *list_of_objects()* e *store2disk()*. A sua função é sugerida pelos próprios nomes. No primeiro caso temos a obtenção de uma lista de objectos que são extraídos a partir do vector de "blobs" e o registo em disco (em formato texto) das respectivas posições e orientações. No segundo, a função regista em disco dois ficheiros que serão posteriormente utilizados: um que contém a descrição dos objectos encontrados e das respectivas características assim como identidades, em binário; e outro que contém um registo idêntico excepto que o modo texto é usado com uma sintaxe compreensível para os restantes agentes; este último é o principal ficheiro de resultados que é enviado para a zona de acesso comum dos agentes.

Para além destas são ainda utilizadas funções que se destinam a executar pequenas tarefas tais como a aquisição de uma imagem em níveis de cinzento (*snap()*), a medição do valor de limiar para binarização daquela imagem (*histogram()*), a binarização propriamente dita (*threshold()*), a inicialização da placa de aquisição de imagens (*init_board()*) e a introdução de um intervalo de espera (*wait_()*).

Aquelas funções fazem parte de um pacote de programas desenvolvidos para o controle do hardware (por exemplo da placa de aquisição de imagem), execução de registos em disco e da sua leitura (especialmente de ficheiros de imagem) e de filtragem, quer pontuais quer por convolução. Estes últimos destinam-se essencialmente à eliminação de ruídos mas também à sua suavização e transformação.

Anexo E

E.1 Fluxograma do Algoritmo de Agrupamento e Encadeamento de Segmentos

Chain.c

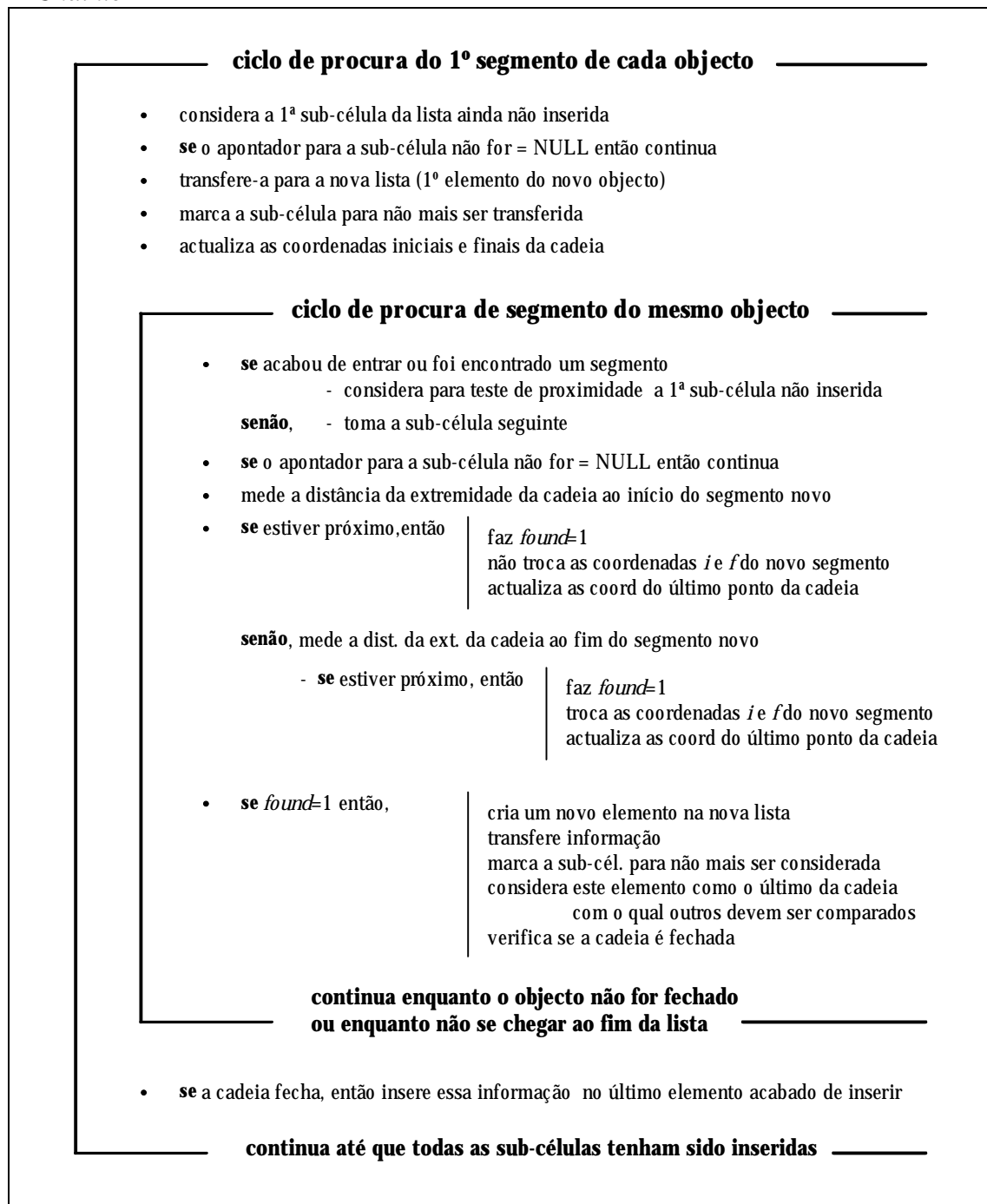


Fig. 1 - Descrição do algoritmo de encadeamento de segmentos.

E.2 Fluxograma do Algoritmo de Extrapolação de Segmentos

Extra.c

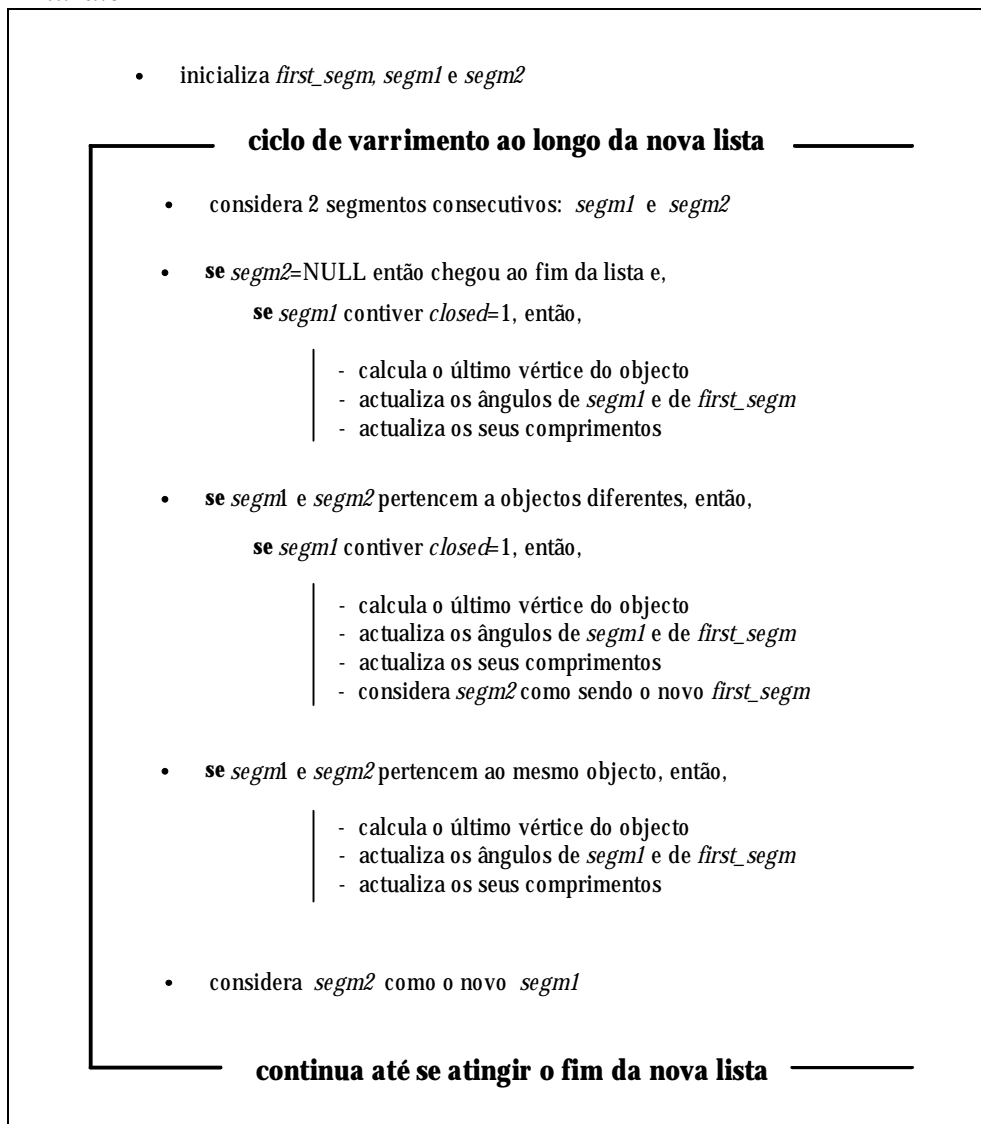


Fig. 2 - Descrição do algoritmo de extrapolação de segmentos.

E.3 Teste de Ligação entre 2 Grupos de Segmentos

one_iteration()

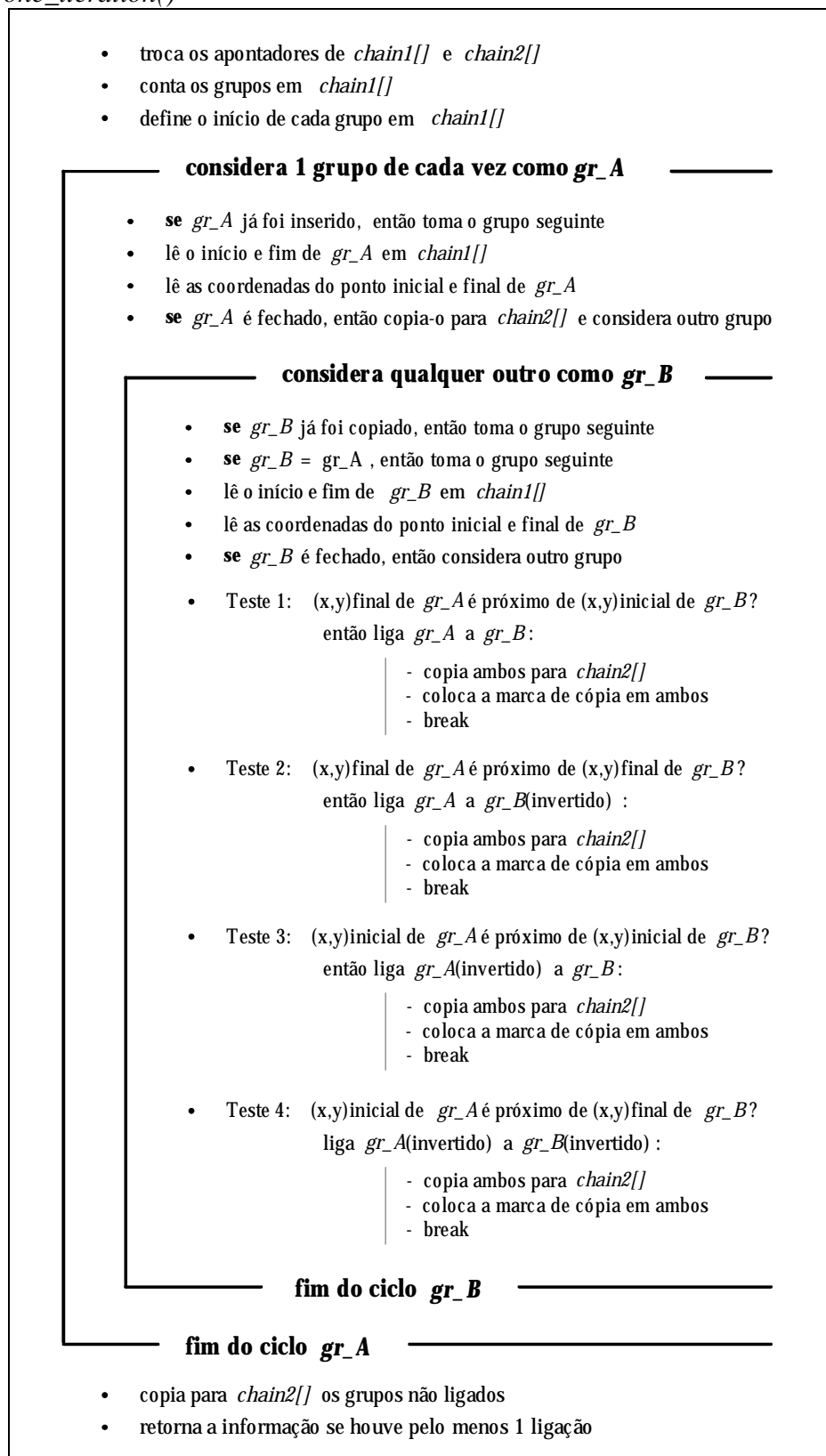


Fig. 3 - Descrição da função *one_iteration()* contida pelo algoritmo de ligação entre grupos.

Anexo F

F.1 Esquema Geral de Chamadas no Agente Laser

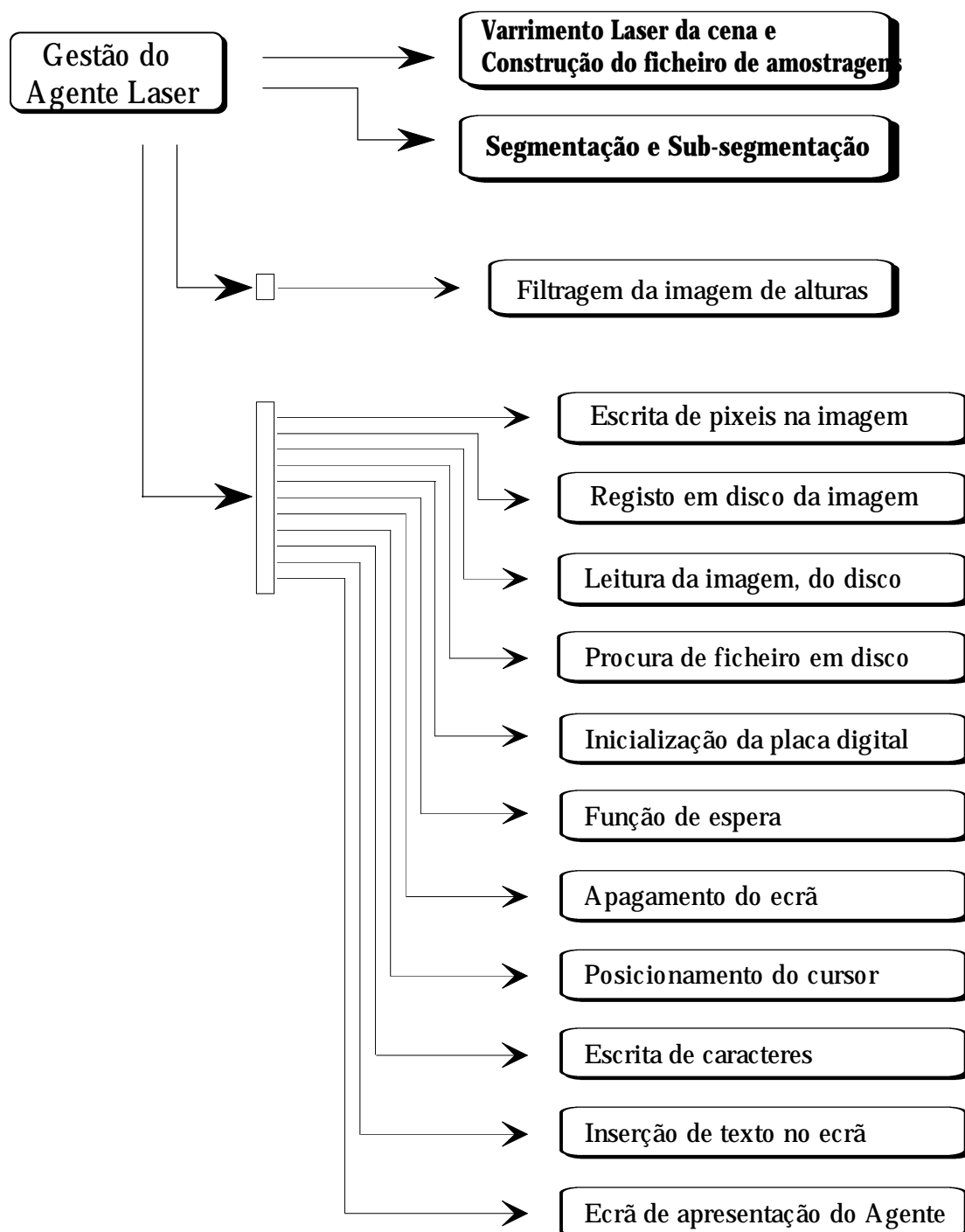


Fig. 1 - Esquema de chamadas a programas e funções no Agente Laser.

Anexo G

G.1 Estrutura de Dados no Vector de Objectos

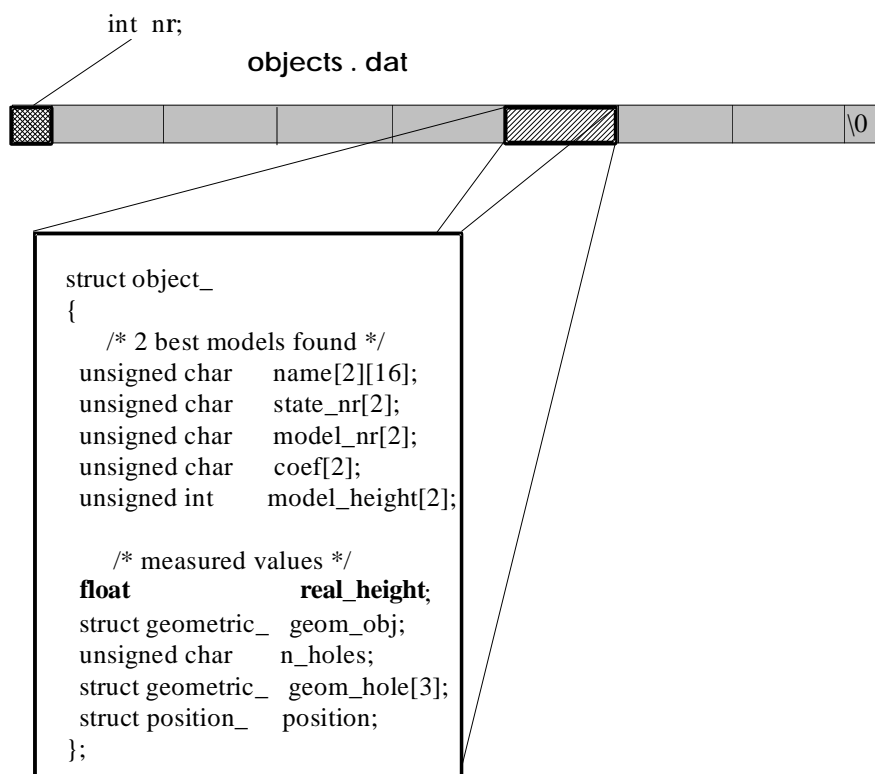


Fig. 1 - Estrutura de cada elemento de entrada do vector de objectos registado em 'objects.dat'.

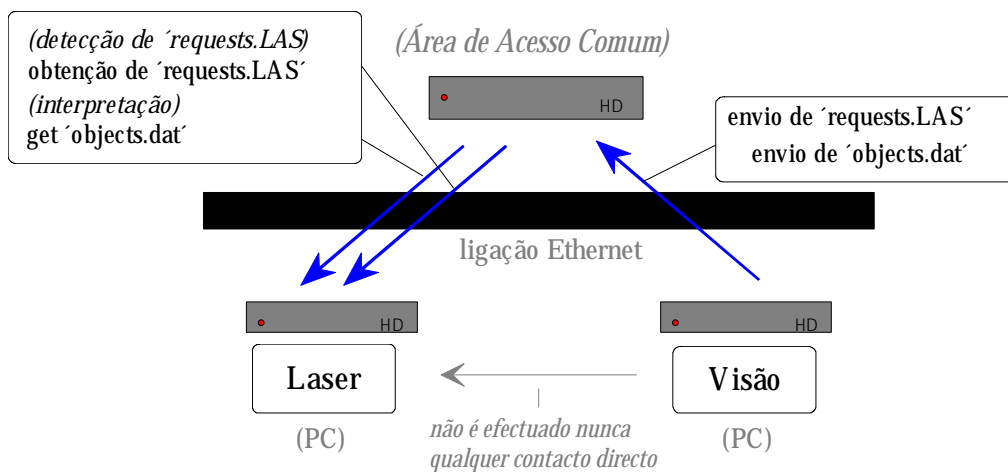


Fig. 2 - Envio de pedido pelo Agente Visão assim como do ficheiro de objectos para a área de acesso comum e recepção destes pelo Agente Laser.

A acção do algoritmo é a seguinte:

1. Através de FTP, usando os comandos contidos no ficheiro 'get_objs', transfere o ficheiro de objectos 'objects.dat' da zona de acesso comum para o disco local. A sua inexistência faz cessar a execução de imediato (Fig.2).

```
'get_objs'  
open turing  
user pal  
PauloLopes  
cd laser  
bin  
get objects.dat  
close  
bye
```

2. Este ficheiro é aberto e a 1ª linha lida, contendo o número de vectores registados nela.
3. Procede-se então à leitura para memória RAM do ficheiro que deverá estar disponível através do apontador *object*.
4. O campo a preencher é "*float real_height*" medido para o ponto (*float x, float y*) em coordenadas de imagem pertencente à estrutura '*struct position_*' incluída na estrutura mais genérica '*struct object_*' de que é composto o vector em 'objects.dat'.
5. Se a imagem de distâncias estiver disponível em ficheiro ('range.img') este é imediatamente lido para a memória da placa de aquisição de imagens que a mantém armazenada (programa '*load_3small()*'). Essa imagem quando transferida para memória irá conter valores "grey-level" de 0 a 255 que traduzem não o valor de intensidade luminosa associada a cada pixel como nas imagens comuns, mas sim um valor codificado da altura associada a esse pixel.

De toda essa imagem com $640 \times 512 = 327.680$ bytes, apenas uma porção de 40 Kb é directamente acessível pelo PC ou seja "espelhado" na sua própria memória RAM (correspondente a 64 das 512 linhas). A quantidade de memória acessível é na realidade $1.024 \times 64 = 64$ Kb, mas apenas $40 \text{ Kb} = 640 \times 64$ contém informação válida ou seja da imagem de alturas.

Nesta situação o programa executa a procura dos valores de CM em unidades de pixel e registados no vector de objectos (*obj->position.x* e *obj->position.y*), lê o respectivo valor "grey-level" armazenado na imagem através de "*RPIXEL(x,y)*", converte-o para um valor de altura em milímetros pela multiplicação do factor ($Z_{\text{máx}}/255$), onde $Z_{\text{máx}}$ representa o valor máximo encontrado (também em *mm*) aquando da construção da imagem e registado nesta nas coordenadas (0,128) e

(1,128) correspondentes às partes inteira e decimal, e finalmente regista o valor achado em "obj->real_height".

6. Uma vez preenchidos todos os campos ".real_height" do vector, este é gravado em 'objectos.dat' após o que este ficheiro é reenviado para a área de acesso comum através de FTP com os comandos contidos no ficheiro 'send_obj'.

'send_obj'

```
open turing
user pal
PauloLopes
cd laser
bin
send objects.dat
close
bye
```

7. É ainda construído um ficheiro ('result3D') com o nome do ficheiro enviado para a área de acesso comum, que neste caso é 'objects.dat' para que os outros agentes tomem conhecimento desse facto. Esse ficheiro é também enviado por FTP com os comandos contidos em 's_res3D' (Fig.3).

's_res3D'

```
open turing
user pal
PauloLopes
cd laser
bin
send result3D
close
bye
```

8. Se não existir o ficheiro de imagem 'range.img' então o algoritmo tem que recorrer ao ficheiro de amostragens de altura 'heights.dat'. O procedimento de procura está intimamente ligado ao modo como as amostragens foram efectuadas que determinou a ordenação de registo tanto em memória como em ficheiro. Convém referir de novo que os valores de Z são memorizados conjuntamente com as coordenadas de imagem associadas uma vez que a periodicidade em x não é constante.

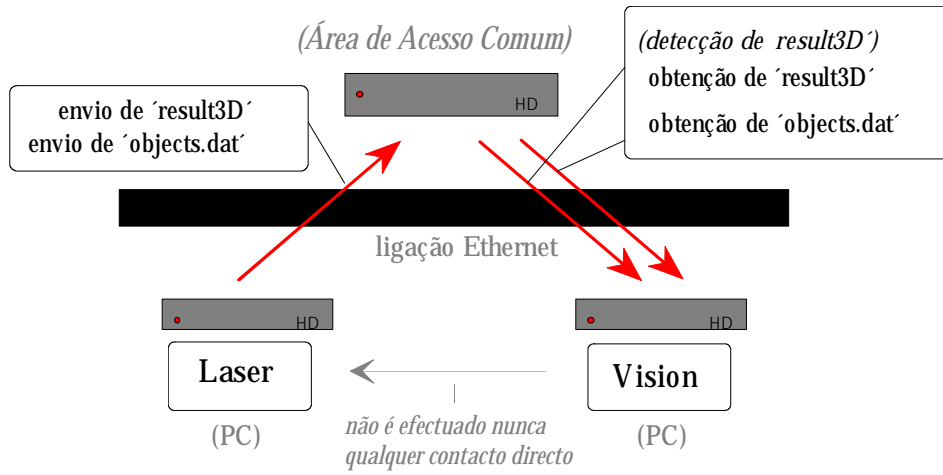


Fig. 3 - Envio de resultados do Agente Laser para a área de acesso comum e recepção destes pelo Agente Visão.

G.2 Ficheiros de Comandos para FTP

São os seguintes os ficheiros de comandos para FTP utilizados:

'g_fCoord'

```
open turing
user pal PauloLopes
cd laser
bin
get fCoord.dat
close
bye
```

's_fCoord'

```
open turing
user pal PauloLopes
cd laser
bin
send fCoord.dat
close
bye
```

A estrutura de dados no ficheiro 'fCoord.dat' é a seguinte:

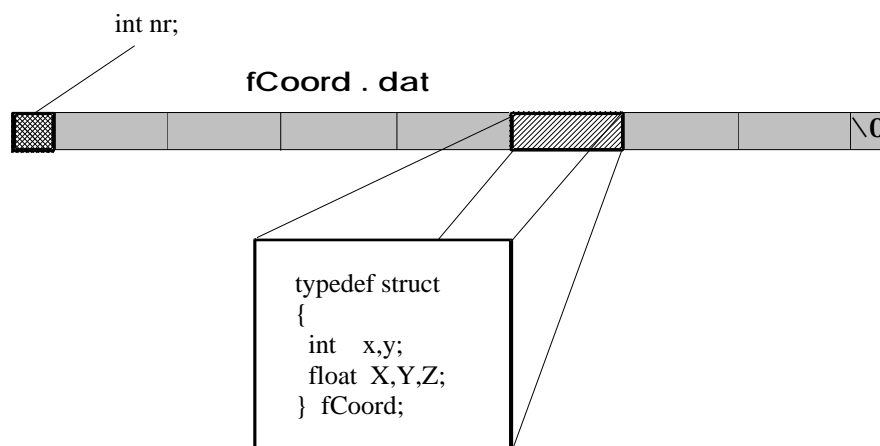


Fig. 4 - Estrutura de cada elemento de entrada do vector de posições registado em 'fCoord.dat'.

G.3 O Esquema de Chamadas da Função de Filtragem

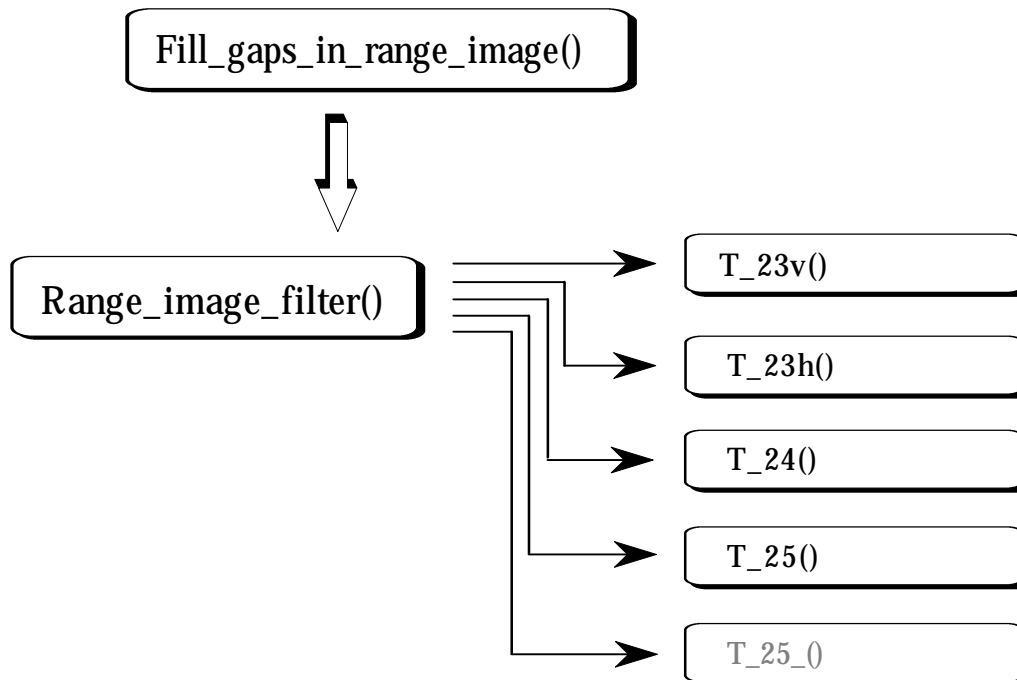


Fig. 5 - Esquema de chamadas dos vários filtros para a obtenção da imagem de alturas.

Anexo H

H.1 Esquema de Chamadas do Programa de Análise de Assinaturas

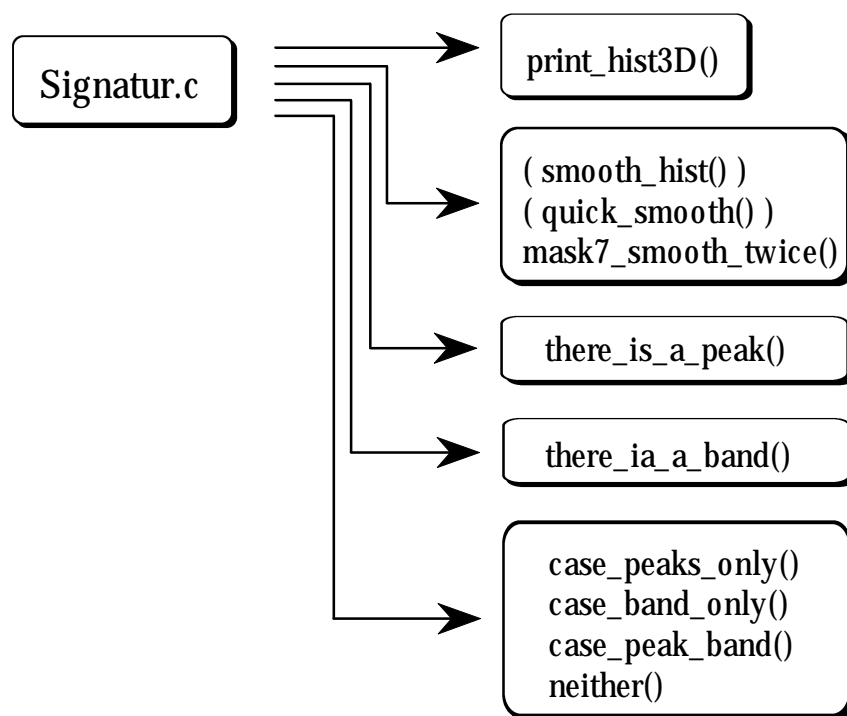


Fig. 1 - Esquema de chamadas do programa de análise de "Assinaturas", 'Signatur.c'.

H.2 Fluxograma do Programa de Análise de "Assinaturas"

Dada a importância deste algoritmo, decidiu-se incluir uma descrição esquemática detalhada que se encontra na Fig.2. Nela são também indicados os parâmetros de chamada da função assim como o valor de retorno.

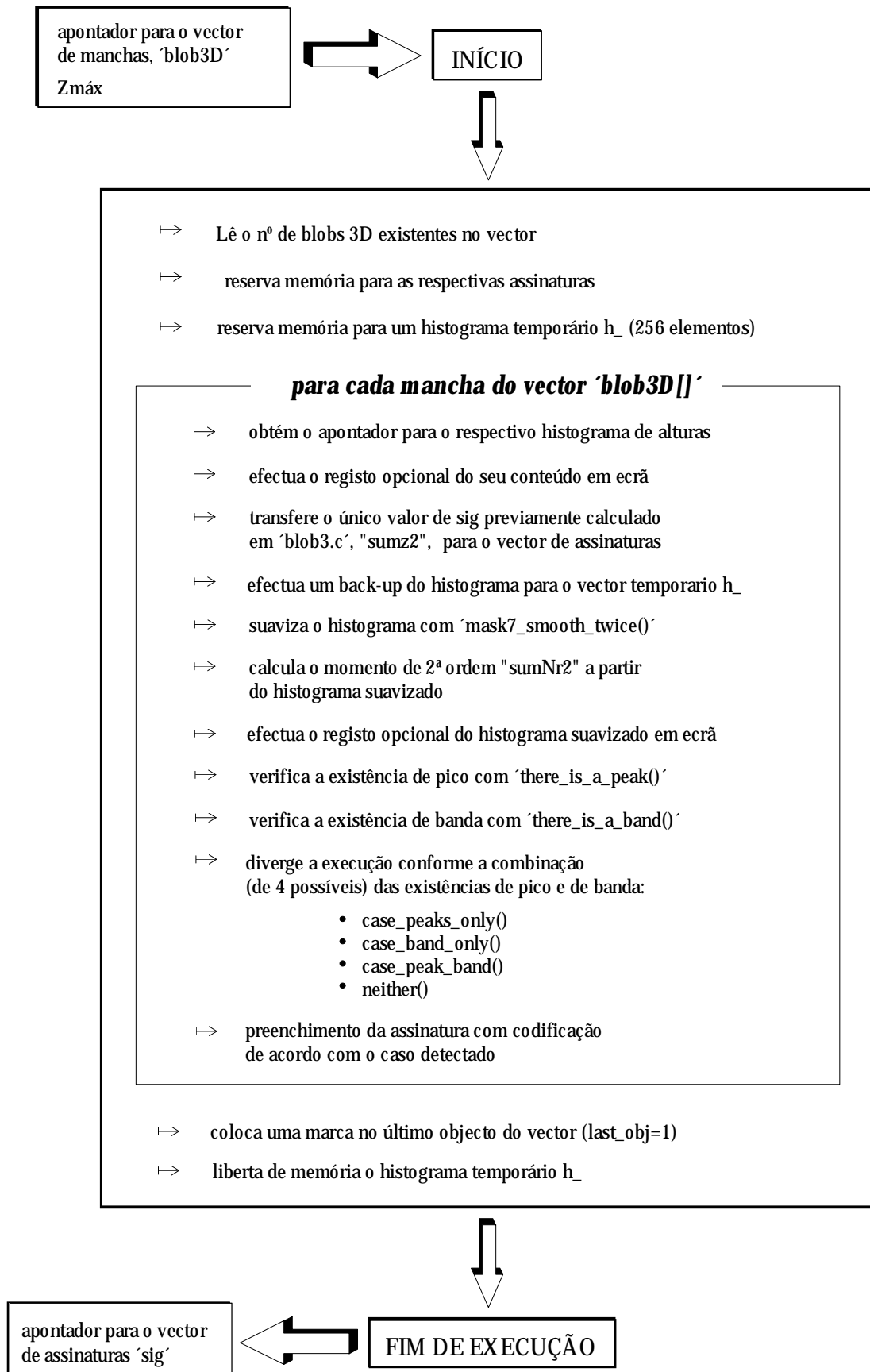


Fig. 2 - Fluxograma do programa de análise de "Assinaturas".

H.3 A Gestão do Ficheiro de Modelos 3D

Este ficheiro de modelos destina-se essencialmente a ser usado pelo algoritmo de identificação 3D, pelo que é conveniente efectuar uma gestão cuidada em "off-line". O utilizador não só deve tentar incluir um número máximo de modelos que poderão vir a estar presentes numa dada manipulação, como também manter actualizados os valores que o ficheiro contém.

Para uma tal gestão foi construído o programa '3Dmodels.c' de muito fácil utilização que para além de incluir novos modelos no ficheiro faz a sua manutenção de diversas formas.

Apresento de seguida diversas particularidades deste programa assim como uma descrição detalhada do modo de proceder em cada uma das opções.

H.3.1 A Linha de Comando

Por uma questão de facilidade e rapidez de uso, o programa admite argumentos em linha de comando que de imediato selecciona a acção a desenvolver. Por exemplo "3Dmodels **l**" selecciona a listagem de todos os modelos existentes na base de dados, mais concretamente dos nomes e número de estados estáveis.

A lista completa de opções é apresentada a seguir:

" 3Dmodels **a1 a2 a3** "

- | | |
|--------------|--|
| h | - lista as opções possíveis |
| l | - lista os nomes e nº de estados estáveis |
| g i | - lê a informação relativa ao modelo <i>i</i> |
| e i | - edita o modelo <i>i</i> e regrava todo o ficheiro |
| e i r | - edita o modelo <i>i</i> e regista em modo de acesso aleatório |
| r i | - remove toda a informação do modelo <i>i</i> |
| a | - acrescenta um modelo novo através do teclado |
| m | - constrói um modelo novo por análise automática com todo o sistema 3D |

O que está de acordo com a resposta dada pelo programa ao comando "3Dmodels **h**":

comando

```
> 3Dmodels h <enter>
```

```
> 3Dmodels <enter>
```

display

```
'3Dmodels.dat' has been found !
ptr = 26E1:0008

Listing all options ...

'h'or ": help: display all options
'l' : list all model names and s.s.
'g 1' : get and display model 1
'e 1' : edit model 1 (extra arg 'r': r.access)
'r 1' : remove model 1
'a' : append new model
'm' : make and append new model

Thank you for your kindness !
```

H.3.2 O Modo de Acesso aos Modelos

Uma vez que a manipulação directa da memória RAM é mais rápida e eficiente, optei por ler de uma só vez o conteúdo de todo o ficheiro de modelos (esta rapidez tornar-se-á notória no processo de identificação em que todos os modelos são lidos sequencialmente para comparação).

É portanto feita uma alocação dinâmica de memória em RAM para onde é copiado integralmente o ficheiro em disco (Fig.3) tal como já foi dito atrás.

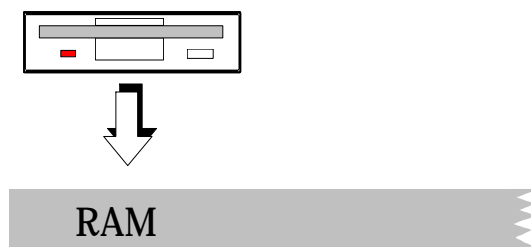


Fig. 3 - Cópia integral do ficheiro de modelos 3D para memória RAM.

Toda e qualquer acção de leitura/escrita será feita neste "buffer" de memória, efectuando-se novo registo em disco no caso de ocorrer alteração do seu conteúdo.

H.3.3 Funções Auxiliares

Para uma correcta manipulação dos modelos em RAM e também por uma questão de estruturação de todo o software desenvolvido, foram elaborados programas auxiliares que executam um determinado número de acções que são necessárias em quase todas as opções. São eles os seguintes:

```

read_disk ( )
goto_next_38 ( )
count_models ( )
goto_model ( )
ll_at_home_position ( )
adjust ( )

```

e que de seguida se descreve em pormenor.

1. read_disk()

Esta função executa a transferência de informação do disco para um "buffer" em memória RAM; verifica a correcta transferência dessa informação e retorna um apontador para aquele "buffer".

2. goto_next_38()

Partindo do pressuposto de que o apontador fornecido aponte um caractere '&', então esta função encarrega-se de o posicionar no próximo '&', ou seja no início de outro registo de modelo, ou no fim do "buffer" de memória caso não haja mais modelos. Por fim, retorna o novo valor do apontador.

Para o conseguir, a função não executa uma procura sequencial do caractere '&': sendo o registo binário, há sempre a possibilidade daquele caractere ocorrer na informação de um dos registos, desse modo assinalando erradamente o início de outro modelo. Por outro lado seria um processo de procura lento que se tornaria notório em ficheiros de modelos de grandes dimensões.

No processo adoptado, não se lêem todos os bytes na procura de '&' mas antes uma única leitura: a do byte referente ao nº total de estados estáveis 'total_states' que se encontra registado imediatamente a seguir ao nome do modelo. Conhecido este, basta incrementar o apontador em " $total_states \times sizeof(Sig_db) + 1$ " para o posicionar no caractere '&' seguinte referente a um novo modelo, ou no fim do "buffer". *Sig_db* é a estrutura que contém a informação 3D do modelo num determinado estado estável e que é a versão reduzida para registo em base de dados da estrutura "Sig" utilizada quer em 'Signatur.c' quer em 'Sub-segm.c'. Esta última, para além dos campos ilustrados numa das figuras apresentadas atrás, contém ainda o nome do modelo (que neste caso seria redundante) e um byte de controle, *last_obj*, que se destina a assinalar o fim de um array de assinaturas.

Ou seja, o apontador salta de imediato para o próximo '&' sem recorrer a qualquer outra leitura. Este processo é bastante rápido e eficiente.

3. count_models()

Se o apontador estiver posicionado no 1º caractere '&', esta função efectua a contagem dos modelos existentes na base de dados, usando para tal a função descrita

anteriormente; compara ainda o número obtido do 1º byte do ficheiro que deveria traduzir o número total de modelos. Se eles não forem iguais, estão este último é corrigido tanto no "buffer" como no ficheiro em disco.

Esta função é chamada frequentemente para confirmar ou corrigir esse 1º byte do ficheiro e mostrou-se particularmente útil na fase de construção e depuração do programa de gestão de modelos.

4. goto_model()

Esta função procede de um modo semelhante ao descrito anteriormente, excepto que pára a execução logo que atinge o nº indicado de caracteres '&'; retorna o novo apontador ou NULL se tiver atingido o fim do "buffer".

5. ll_at_home_position()

6. adjust()

Estas funções destinam-se a controlar o hardware que comanda o motor passo-a-passo, de uma forma interactiva e que são utilizadas numa das opções. A primeira questiona o utilizador sobre a posição da linha laser e opcionalmente chama a segunda. Esta aceita comandos através das duas teclas de cursor (cima e baixo) assim como das teclas de avanço ou recuo da página (PageDown ou PageUp). Enquanto que as primeiras executam o envio de 3 passos ao motor passo-a-passo em qualquer dos dois sentidos, as segundas permitem avanços da linha laser de maior amplitude já que enviam 20 passos de cada vez. Para um bom posicionamento da linha laser na posição de referência ("Home position") há que ter em conta a folga no caixa de desmultiplicação do motor pelo que é sempre aconselhável recuar pelo menos 40 passos (duas pressões na tecla "PageDown") antes de acertar a linha em movimento progressivo (com a tecla de cursor para cima ou com "PageUp"), ou seja, no sentido do varrimento da linha laser na cena.

H.3.4 Funções Principais

1. list_model_names()

Esta função destina-se a listar o conteúdo dos campos 'name[]' e 'total_states' de todo o ficheiro em memória, ou seja, dos nomes e números de estados estáveis associados a cada modelo; utiliza para tal a função auxiliar 'goto_next_38()'. A figura seguinte mostra todos os passos executados.

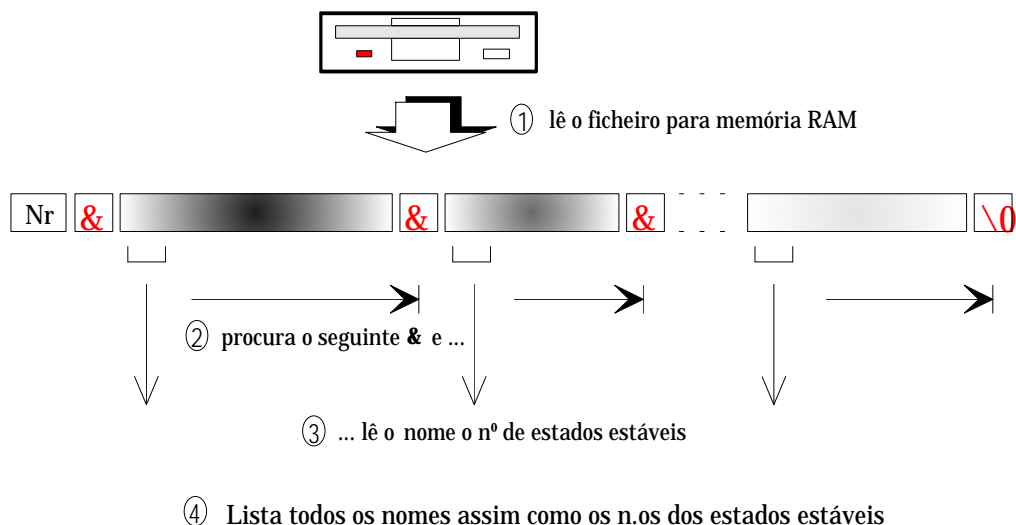


Fig. 4 - Listagem dos nomes de todos os modelos da base de dados.

A selecção desta opção poderia ter como resposta a seguinte lista:

comando

```
> 3Dmodels 1 <enter>
```

display

```
'3Dmodels.dat' has been found !
ptr = 26E1:0008

Listing all model names and s.s. ...

There are -5- models in Database!

model nr - 1- :      object 1      nr of ss = 1
model nr - 2- :      object 2      nr of ss = 1
model nr - 3- :      object 3      nr of ss = 1
model nr - 4- :      object 4      nr of ss = 3
model nr - 5- :      object 5      nr of ss = 3

Thank you for your kindness !
```

2. remove_model()

Perante a limitação de memória RAM e quando o ficheiro se torna demasiado extenso, há a necessidade de eliminar um ou mais modelos, até que o espaço disponível em RAM permita a sua memorização. Pode também haver desinteresse em manter registo de modelos que não serão mais utilizados, pelo que será mais racional retirá-los da base de dados.

Sabendo o nº do modelo a retirar, procuram-se os endereços de memória do caractere '&' correspondente fazendo uso de *goto_model(i)* e do último byte do registo, imediatamente antes do '&' seguinte ou do fim do "buffer", como mostra a Fig.5.

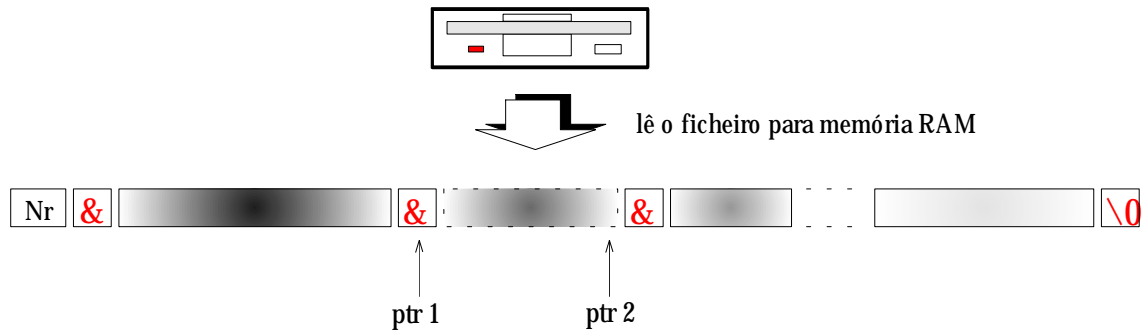


Fig. 5 - Localização dos apontadores quando da remoção de um modelo da base de dados.

Se o modelo for o último da lista, então bastará "cortar" o ficheiro em memória pelo endereço indicado por $(p_1 - 1)$ e registá-lo em disco com o novo comprimento.

Se não for esse o caso, há que calcular o comprimento do bloco imediatamente acima do modelo a retirar e movê-lo de forma a sobrepor-se-lhe, tal como a Fig.6 ilustra:

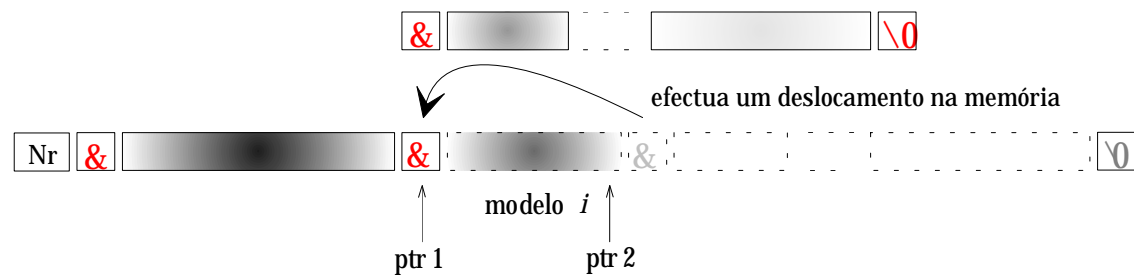


Fig. 6 - Remoção do modelo da base de dados.

Para actualizar o ficheiro em disco, basta gravar o conteúdo do "buffer" numa extensão igual ao comprimento inicial do ficheiro subtraído do comprimento do registo do modelo retirado:

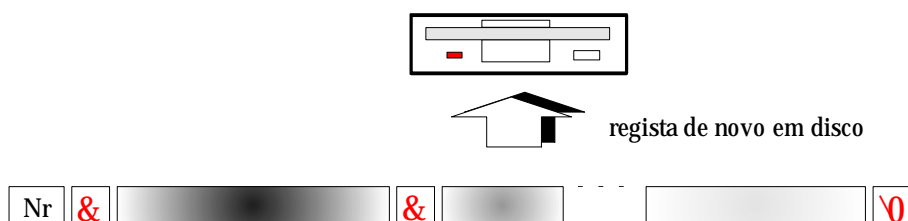


Fig. 7 - Actualização em disco do ficheiro de modelos.

Em qualquer dos casos anteriores, é acrescentado no fim da zona válida do "buffer" o caractere '\x0' de forma a eliminar qualquer '&' que lá permaneça (é o caso de se ter retirado o último modelo da lista). Se tal não for feito e se o programa de gestão de modelos for corrido de novo, a lista de modelos será provavelmente alocada no mesmo segmento. Ora, no fim do registo válido, encontrar-se-ia ainda a informação do modelo previamente retirado assim como o respectivo caractere '&' de separação. Nesta circunstância ele seria novamente considerado como válido mantendo toda a sua informação disponível, embora já não constasse do ficheiro em disco. A substituição daquele caractere '&' por '\x0' fará parar a procura e ignorar essa informação.

A ordem de remoção do modelo nº4, na sequência do exemplo apresentado atrás, daria como resposta o seguinte:

comando

```
> 3Dmodels r 4 <enter>
```

display

```
'3Dmodels.dat' has been found !
ptr = 26E1:0008

      Removing model nr -4- ...

Removal of model -4- successful !

Thank you for your kindness !
```

Uma ordem de listagem ("3Dmodels l") demonstraria que de facto o modelo fora retirado:

comando

```
> 3Dmodels l <enter>
```

display

```
'3Dmodels.dat' has been found !
ptr = 26E1:0008

      Listing all model names and s.s. ...

There are -4- models in Database!

model nr - 1- :      object 1      nr of ss = 1
model nr - 2- :      object 2      nr of ss = 1
model nr - 3- :      object 3      nr of ss = 1
model nr - 4- :      object 5      nr of ss = 3

Thank you for your kindness !
```

3. `get_model()`

Esta função destina-se a mostrar o conteúdo de um dado modelo existente na base de dados.

Usando uma das funções auxiliares, posiciona-se um apontador no início do modelo pretendido; lê-se de imediato o seu nome assim como o número de estados estáveis. De seguida, lêem-se tantas estruturas *Sig_db* quanto aquele número. Transforma-se deste modo um registo binário contínuo numa estrutura com informação compreensível e utilizável. Uma vez de posse de toda a informação do modelo, resta registá-la no ecrã para leitura (Fig.8).

A ordem de listagem do modelo nº4 após a remoção descrita atrás daria o seguinte resultado:

comando

```
> 3Dmodels g 4 <enter>
```

display

```
'3Dmodels.dat' has been found !  
ptr = 26E1:0008  
  
Reading model nr -4- ...  
displaying model info now...  
  
name = object 5  
nr. of s. states = 3  
  
State nr 1:  
  peak -1-:  Z[0] = 10,  nr[0] = 100  
State nr 2:  
  peak -1-:  Z[0] = 20,  nr[0] = 200  
  peak -2-:  Z[1] = 20,  nr[1] = 200  
State nr 3:  
  peak -1-:  Z[0] = 10,  nr[0] = 100  
  peak -2-:  Z[1] = 20,  nr[1] = 200  
  peak -3-:  Z[2] = 30,  nr[2] = 300  
  
Thank you for your kindness !
```

De notar que em virtude do modelo nº4 ter sido retirado, a posição é agora ocupada pelo modelo de nome "object 5". Aliás se for pedida a listagem do conteúdo do modelo nº5, o programa alertará para a sua inexistência:

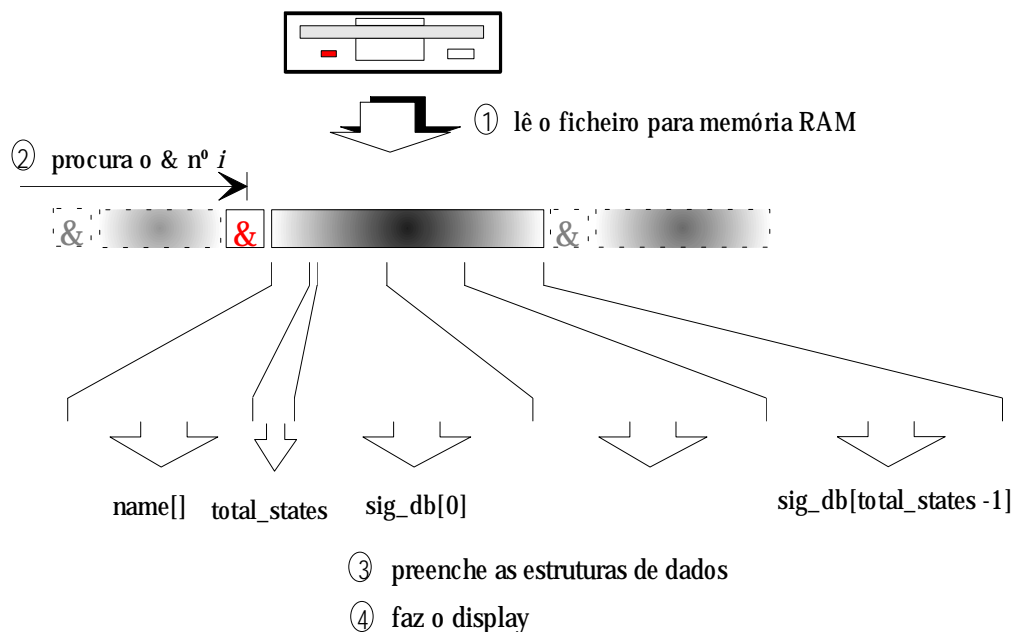


Fig. 8 - A estrutura do registo de um modelo no ficheiro e a sua leitura para variáveis conhecidas.

comando

```
> 3Dmodels g 4 <enter>
```

display

```
'3Dmodels.dat' has been found !
ptr = 26E1:0008

  Reading model nr -5- ...

There is no model nr -5- ! max nr is -4- !

Thank you for your kindness !
```

4. edit_model()

Nesta opção executam-se todos os passos da anterior, excepto o registo em ecrã. Uma vez de posse de toda a informação do modelo i , é dada ao utilizador a possibilidade de corrigir todos os valores numéricos. Mantém-se contudo a condição de se manter inalterado o número de estados estáveis (supõe-se não ter havido alteração da forma do modelo) assim como o número de faces planas visíveis em cada estado estável.

Uma vez corrigida toda a informação, esta é copiada para o "buffer" em RAM (Fig.9). Se a linha de comando não contiver mais nenhum parâmetro, então o "buffer" contendo todos os modelos é integralmente registado em disco, sendo o anterior

ficheiro, apagado. A inclusão do parâmetro "r" evita a gravação integral do ficheiro: outra secção do programa é chamada que abre o ficheiro de modelos em modo de acesso aleatório; posiciona um apontador no 1º byte a ser alterado e sobrepõe a informação do modelo editado. Quando o ficheiro de modelos é bastante extenso, esta opção mostra-se vantajosa pelo menor tempo de execução que requer.

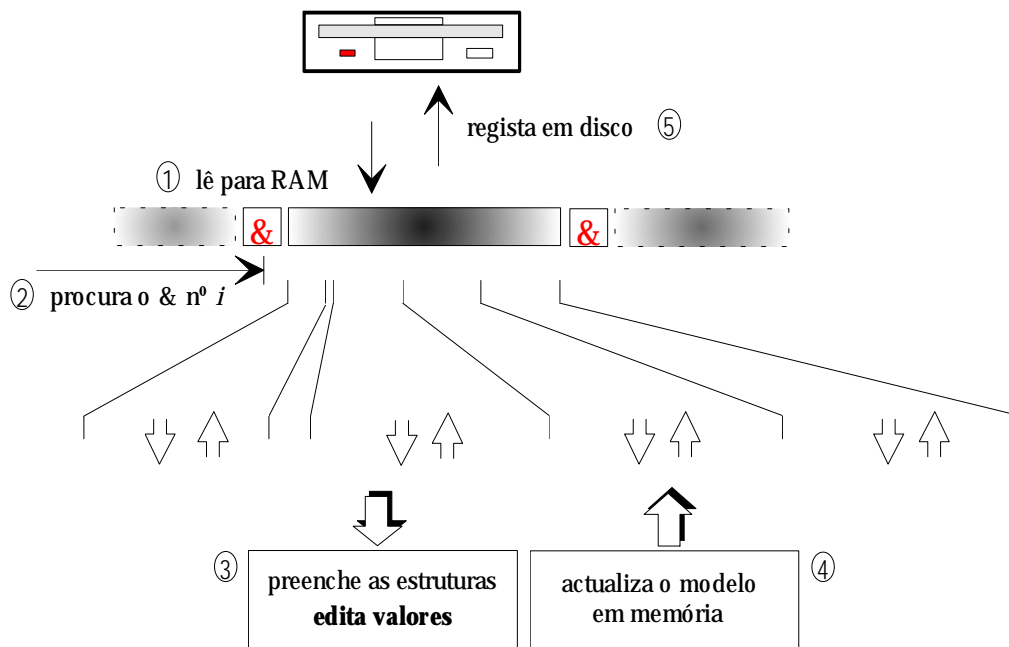


Fig. 9 - A edição de um modelo.

Em qualquer dos casos, só é permitida correcção dos campos não vazios tal como é demonstrado a seguir:

comando

```
> 3Dmodels e 2 <enter>
```

display

```
'3Dmodels.dat' has been found !
ptr = 26E1:0008

Editing model nr -2- ...

name = object 2; new name = obj 2

----- State nr -1- -----

peak 1:  Z[1] = 10;    new Z[1] = 11
         nr[1] = 100;   new nr[1] = 110
peak 2:  Z[2] = 20;    new Z[2] = 22
         nr[2] = 200;   new nr[2] = 220
```

Change model now ? (enter,Esc)

Model -2- has been successfully modified !

Thank you for your kindness !

Uma nova listagem de modelos assim como o "display" do modelo editado mostra as alterações efectuadas:

comando

> 3Dmodels l <enter>

display

'3Dmodels.dat' has been found !
ptr = 26E1:0008

Listing all model names and s.s. ...

There are -4- models in Database!

model nr - 1- :	object 1	nr of ss = 1
model nr - 2- :	obj 2	nr of ss = 1
model nr - 3- :	object 3	nr of ss = 1
model nr - 4- :	object 5	nr of ss = 3

Thank you for your kindness !

comando

> 3Dmodels g 2 <enter>

display

'3Dmodels.dat' has been found !
ptr = 26E1:0008

Reading model nr -2- ...
displaying model info now...

name = **obj 2**
nr. of s. states = **1**

State nr 1:

peak -1-:	Z[0] = 11 ,	nr[0] = 110
peak -2-:	Z[1] = 22 ,	nr[1] = 220

Thank you for your kindness !

5. `append_model()`

Esta função destina-se a acrescentar ao ficheiro um novo modelo cujas características são conhecidas pelo utilizador. Uma vez de posse do número de estados estáveis do modelo, o programa encarrega-se de pedir toda a informação relativa a cada um desses estados.

Embora a variável definida para o registo destes valores contenha o número máximo permitido de campos, apenas os estados estáveis válidos são transferidos para o ficheiro em disco, precedidos pelo separador '&'. Quanto ao 1º byte do ficheiro, ele toma logo de início o valor '\x1' caso '3Dmodels.dat' não exista e tenha por isso que ser criado; caso contrário ele é incrementado para corresponder ao novo número total de modelos contidos naquele ficheiro.

São as seguintes as estruturas onde são recolhidos os dados do novo modelo:

<pre>typedef struct { unsigned char name[16]; unsigned char total_states; Sig_db sig[max_states]; } _3Dmodel;</pre>	<pre>typedef struct { char nr_peaks; int Z[3]; int nr[3]; } Sig_db;</pre>
---	---

Sig_db é a versão reduzida da estrutura referente à "assinatura" 3D utilizada em 'Signatur.c' e 'Sub-segm.c'.

A experiência seguinte corresponde à inclusão no ficheiro de um 6º objecto:

comando

```
> 3Dmodels a <enter>
```

display

```
'3Dmodels.dat' has been found !
ptr = 26E1:0008

  Appending new model ...

== Please supply the NEW MODEL ==

name = object 6
Nr. of stable states = 3

----- State nr -1- -----
nr_peaks = 1
peak 1:  Z[1] = 12
         nr[1] = 123
```

```

----- State nr -2- -----
nr_peaks = 2
peak 1:  Z[1] = 23
          nr[1] = 234
peak 2:  Z[2] = 24
          nr[2] = 245

```

```

----- State nr -3- -----
nr_peaks = 3
peak 1:  Z[1] = 12
          nr[1] = 123
peak 2:  Z[2] = 23
          nr[2] = 234
peak 3:  Z[3] = 34
          nr[3] = 345

```

Save model now ? (enter,Esc)

Nr of models in database has been increased from -4- to -5- !

New model has been appended to Database.

Thank you for your kindness !

Uma listagem mostra que de facto ele foi incluído:

comando

```
> 3Dmodels l <enter>
```

display

```
'3Dmodels.dat' has been found !
ptr = 26E1:0008
```

Listing all model names and s.s. ...

There are -5- models in Database!

model nr - 1- :	object 1	nr of ss = 1
model nr - 2- :	obj 2	nr of ss = 1
model nr - 3- :	object 3	nr of ss = 1
model nr - 4- :	object 5	nr of ss = 3
model nr - 5- :	object 6	nr of ss = 3

Thank you for your kindness !

e que os valores foram correctamente registados:

comando

```
> 3Dmodels g 5 <enter>
```

display

```
'3Dmodels.dat' has been found !
ptr = 26E1:0008

      Reading model nr -5- ...
      displaying model info now...

name = object 6
nr. of s. states = 3

State nr 1:
  peak -1-:  Z[0] = 12,  nr[0] = 123
State nr 2:
  peak -1-:  Z[0] = 23,  nr[0] = 234
  peak -2-:  Z[1] = 24,  nr[1] = 245
State nr 3:
  peak -1-:  Z[0] = 12,  nr[0] = 123
  peak -2-:  Z[1] = 23,  nr[1] = 234
  peak -3-:  Z[2] = 34,  nr[2] = 345

Thank you for your kindness !
```

6. `make_new_model()`

Esta opção permite acrescentar um novo modelo à base de dados existente, mas ao contrário da anterior, não aceita valores do teclado com as únicas excepções do nome e do nº total de estados estáveis. De facto ela efectua todas as operações e medidas: controla o varrimento do feixe laser (a baixo, médio e alto nível), adquire imagens sucessivas, binariza-as em tempo real, efectua a aquisição de alturas ('3D.c'), faz o "display" da matriz de alturas, filtra essa imagem de forma a obter a imagem de "range" ('Filt-obj.c'), chama o algoritmo de segmentação no qual é construído o histograma 3D ('Blob3.c') e finalmente extrai dele os parâmetros identificadores que vão caracterizar o modelo ('Signatur.c'). É um processo inteiramente automático que envolve todo o software 3D desenvolvido!

Este conjunto de parâmetros é registado na base de dados sem qualquer outra intervenção do utilizador que não seja a de recolocar o modelo noutra estado estável para nova inspecção (Fig.10).

Nesta opção é dada ao utilizador a liberdade de posicionar a linha laser correctamente no ponto origem de referência ("home position"). Se bem que após cada varrimento a linha retorne automaticamente ao ponto inicial, pronta para novo varrimento, o facto é que no início ela pode não se encontrar bem posicionada e daí o uso das funções '`ll_at_home_position()`' e '`adjust()`'. Estas são funções interactivas que utilizam software de controle do motor de baixo e médio nível e permitem pela simples pressão de algumas teclas, deslocar a linha laser e assim posicioná-la correctamente no ponto de referência pretendido. Para maior rapidez de ajuste, a função admite dois tipos

de avanço em qualquer um dos sentidos, sendo um deles um ajuste fino para um melhor posicionamento final.

A necessidade de posicionar correctamente na posição de origem deve-se ao facto de, por uma questão de rapidez do processo, o varrimento se efectuar numa zona restrita em torno da posição ($X=100$; $Y=0$) em coordenadas do robot. Aliás o utilizador é desde logo convidado a posicionar o novo modelo nessas coordenadas. Um desvio inicial da linha laser relativamente à referência poderia fazer com que o varrimento não cobrisse a totalidade do objecto ou mesmo o perdesse por completo, originando em qualquer dos casos valores errados.

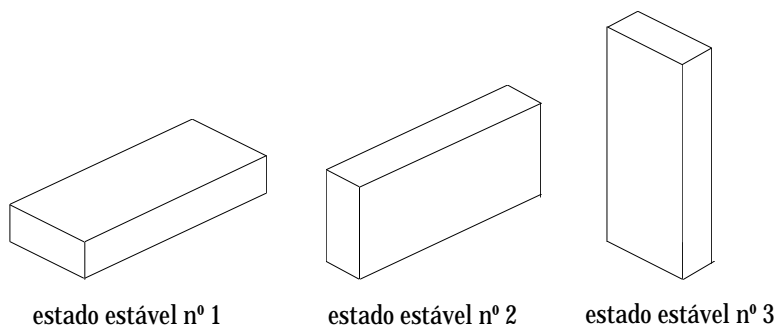


Fig. 10 - Representação dos vários estados estáveis de um poliedro.

A experiência apresentada a seguir corresponde a um caso real de um objecto com 3 estados estáveis (paralelepípedo):

comando

```
> 3Dmodels m <enter>
```

display

```
'3Dmodels.dat' has been found !
ptr = 26E1:0008

    Making and appending new model ...

choice made was 'make new model' !

===== Making a new 3Dmodel =====

Please supply the following data:

name = object 7
Nr of stable states = 3

place object in stable state nr -1- near (X,Y)=(100,0)
Laser line at home position ? (<enter>/n) n

Adjusting the Home Position ...

<enter> when ready...

y (min) = 186
```

```

shift (max) = 194      shift (min) = 158
                      y (max) = 346

'output3D.dat' will be used as filename for formatted output!

line 0
x_ref1 = 351, x_ref2 = 352
...
line 35
x_ref1 = 245, x_ref2 = 250

storing to disk ...
storing 'heights.dat' (binary format) ...
coming back ...
plot to monitor ... brightest pixel for h = 33 mm

Writing to disk blocks 2 to 6: 'range.img'... (2 seconds)
Zmax = 33.0
interval of range: [0,0]
Building Range Histograms ...
blob3> 1 second ...

Range Blobs already found !

Nr of 'childs' of background = 1
Total nr of Range Blobs = 1

max_hists = 4
print hist3D ? (y/n) n
smoothed histogram: print hist3D ? (y/n) n

place object in stable state nr -2- near (X,Y)=(100,0)
Laser line at home position ? (<enter>/n)

<enter> when ready...

                      y (min) = 186
shift (max) = 194      shift (min) = 158
                      y (max) = 346

'output3D.dat' will be used as filename for formatted output!

line 0
x_ref1 = 351, x_ref2 = 352
...
line 35
x_ref1 = 246, x_ref2 = 250

storing to disk ...
storing 'heights.dat' (binary format) ...
coming back ...
plot to monitor ... brightest pixel for h = 67 mm

Writing to disk blocks 2 to 6: 'range.img'... (2 seconds)

```

```

Zmax = 66.9
interval of range: [0,0]
Building Range Histograms ...
blob3> 1 second ...

Range Blobs already found !

Nr of 'childs' of background = 1
Total nr of Range Blobs = 1

max_hists = 2
print hist3D ? (y/n) n
smoothed histogram: print hist3D ? (y/n) n

place object in stable state nr -3- near (X,Y)=(100,0)
Laser line at home position ? (<enter>/n)

<enter> when ready...

                y (min) = 186
shift (max) = 194      shift (min) = 158
                y (max) = 346

'output3D.dat' will be used as filename for formatted output!

line 0
x_ref1 = 351, x_ref2 = 352
...
line 35
x_ref1 = 245, x_ref2 = 250

storing to disk ...
storing 'heights.dat' (binary format) ...
coming back ...
plot to monitor ... brightest pixel for h = 96 mm

Writing to disk blocks 2 to 6: 'range.img'... (2 seconds)
Zmax = 95.7
interval of range: [0,0]
Building Range Histograms ...
blob3> 1 second ...

Range Blobs already found !

Nr of 'childs' of background = 1
Total nr of Range Blobs = 1

max_hists = 2
print hist3D ? (y/n) n
smoothed histogram: print hist3D ? (y/n) n

Save model now ? (enter,Esc)
Nr of models in database has been increased from -5- to -6- !

Thank you for your kindness !

```

mais uma vez, uma ordem de listagem demonstra que o novo modelo foi incluído,

comando

```
> 3Dmodels l <enter>
```

display

```
'3Dmodels.dat' has been found !  
ptr = 26E1:0008  
  
Listing all model names and s.s. ...  
  
There are -6- models in Database!  
  
model nr - 1- :    object 1      nr of ss = 1  
model nr - 2- :      obj 2      nr of ss = 1  
model nr - 3- :    object 3      nr of ss = 1  
model nr - 4- :    object 5      nr of ss = 3  
model nr - 5- :    object 6      nr of ss = 3  
model nr - 6- :    object 7    nr of ss = 3  
  
Thank you for your kindness !
```

e registou valores perfeitamente aceitáveis:

comando

```
> 3Dmodels g 6 <enter>
```

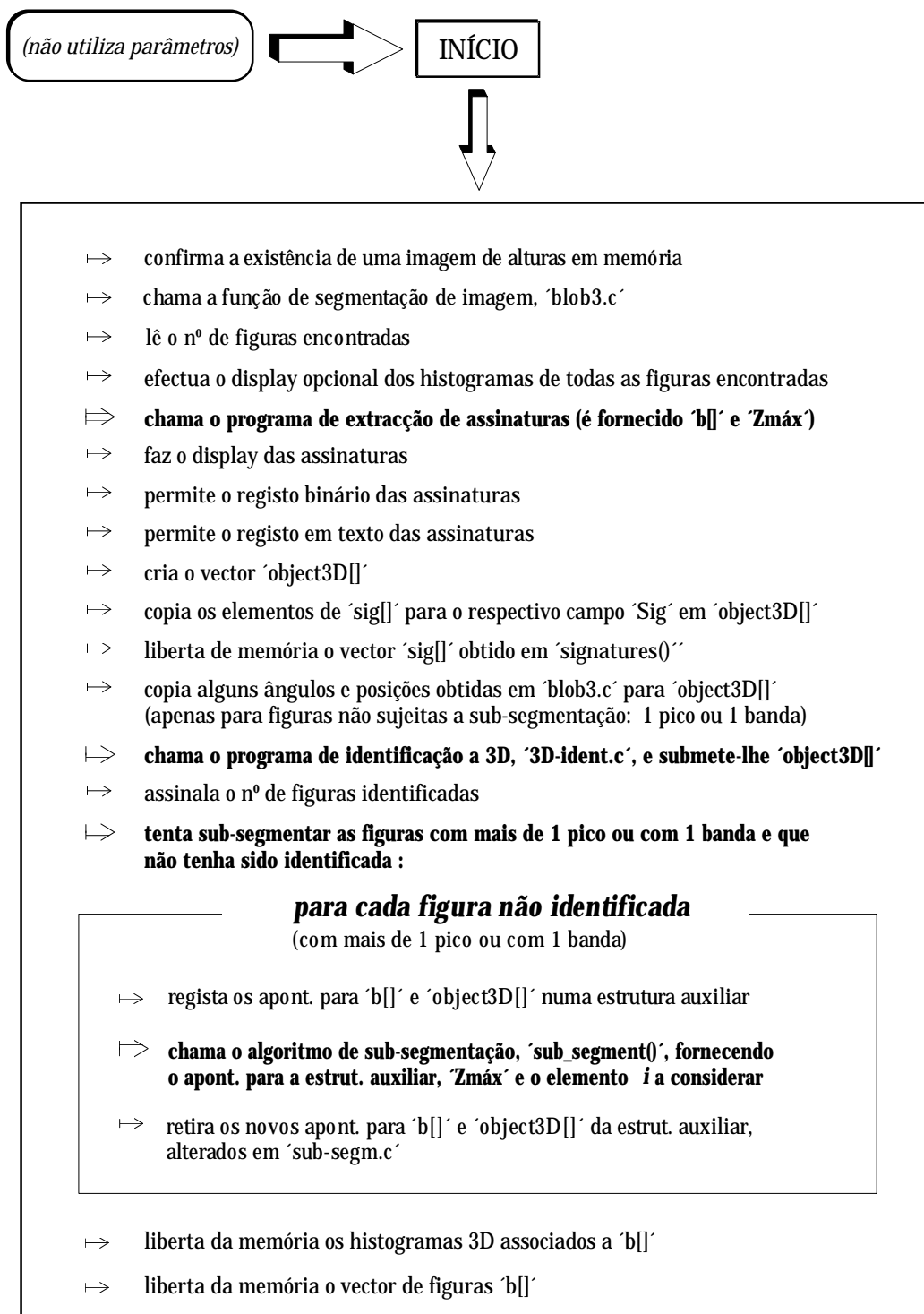
display

```
'3Dmodels.dat' has been found !  
ptr = 26E1:0008  
  
Reading model nr -6- ...  
displaying model info now...  
  
name = object 7  
nr. of s. states = 3  
  
State nr 1:  
  peak -1-:  Z[0] = 29, nr[0] =4127  
State nr 2:  
  peak -1-:  Z[0] = 63, nr[0] =2008  
State nr 3:  
  peak -1-:  Z[0] = 93, nr[0] = 755  
  
Thank you for your kindness !
```

Esta opção tem fornecido bons resultados o que vem confirmar o bom desempenho de todo o software 3D desenvolvido.

Anexo I

I.1 a) Descrição Esquemática do Programa Principal de Gestão do Agente Laser



objectos não considerados para sub-segmentação por terem sido identificados (só aqueles possuidores de mais do que 1 pico)

reavaliação da posição e ângulo considerando apenas a sup. plana superior (pico superior) fazendo nova chamada a 'blob3.c' (com fornecimento do intervalo de níveis de cinzento a considerar assim como as coordenadas do rectângulo delimitador) para a obtenção de valores mais favoráveis à manipulação robótica

para cada figura descrita atrás:

- ↳ regista o último Z da assinatura no campo Z da posição
- ↳ define um novo intervalo de níveis de cinzento referente ao plano superior
- ↳ define o rectângulo delimitador para o novo varrimento
- ↳ chama o programa de varrimento 'blob3.c'
- ↳ copia a nova posição e ângulo para o vector 'object3D[]'
- ↳ liberta de memória o vector da figura 'b[]' assim como o respectivo histograma

↳ **cálculo de θ e α para poliedros rodados no espaço**

- ↳ cálculo do novo ângulo de orientação α segundo a direcção dos ZZ crescentes
- ↳ cálculo do parâmetro local de conversão 'p_to_mm'
- ↳ avanço na direcção dos ZZ crescentes desde o cm até a o fim do objecto rodado
- ↳ avaliação da distância caminhada na horizontal com 'new_x', 'new_y' e 'p_to_mm'
- ↳ avaliação do desnível em Z associado àquela caminhada
- ↳ cálculo e registo do ângulo de rotação no espaço, θ
- ↳ registo e marcação em ecrã do melhor ponto para apreensão robótica
- ↳ leitura da matriz de conversão de coordenadas M
- ↳ conversão de (x,y) para (X,Y) reais em mm, a partir de Z, para todos os objectos
- ↳ registo de 'result3D', 'results.Las' e 'remove.Las'
- ↳ envio dos 3 ficheiros para o Turing usando o FTP

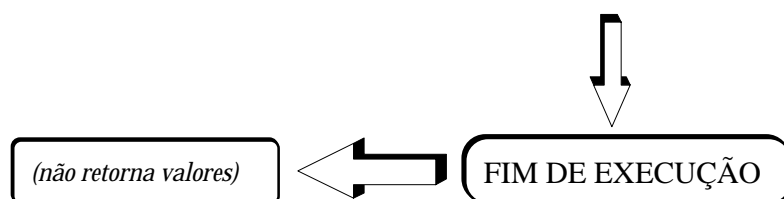


Fig. 1 - Descrição esquemática do programa de gestão do Agente Laser.

I.1 b) Esquema de Chamadas do Programa Principal de Gestão do Agente Laser

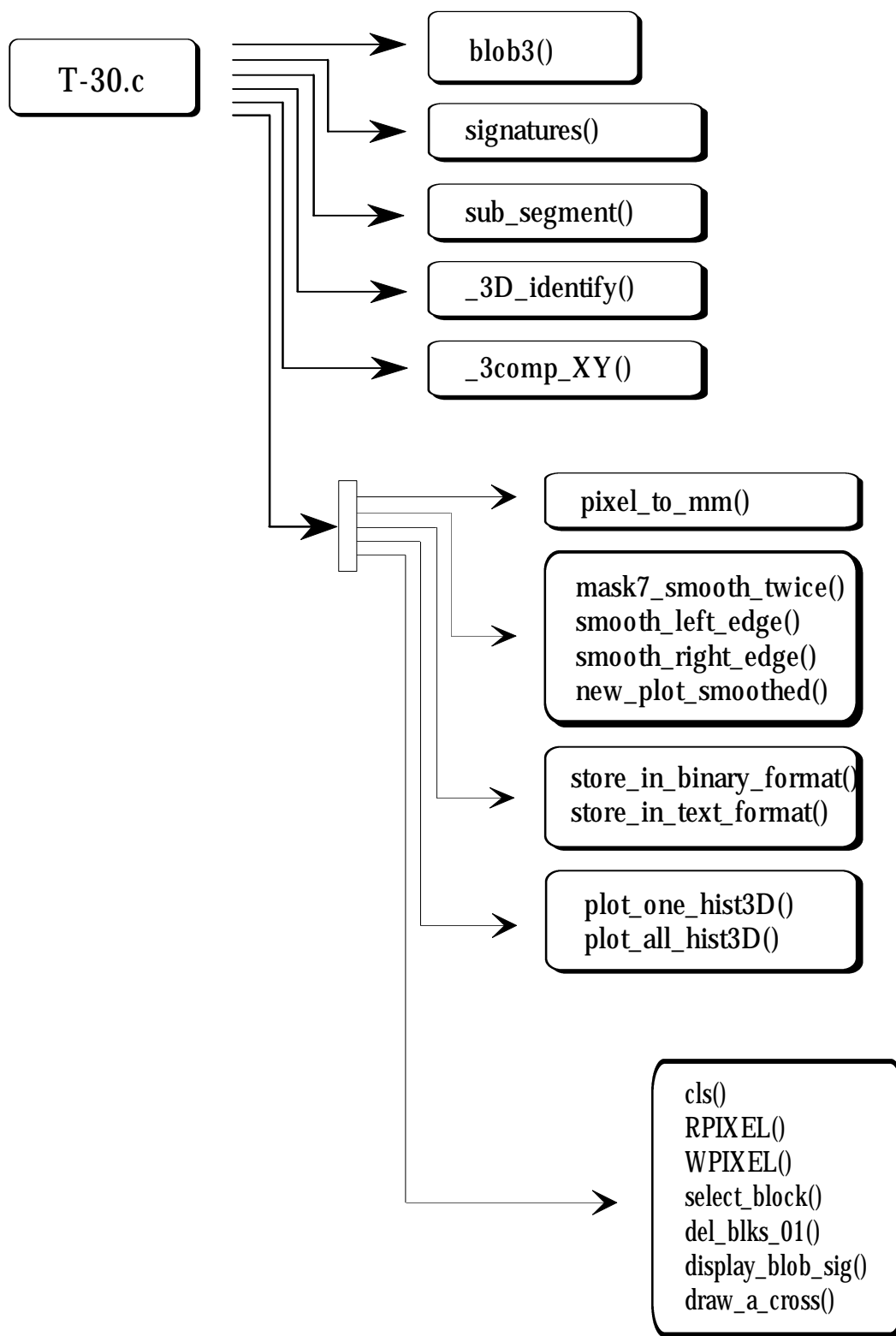


Fig. 2 - Esquema de chamadas do programa de gestão do Agente Laser.

I.2 a) Esquema de Chamadas do Programa de Segmentação a 3D

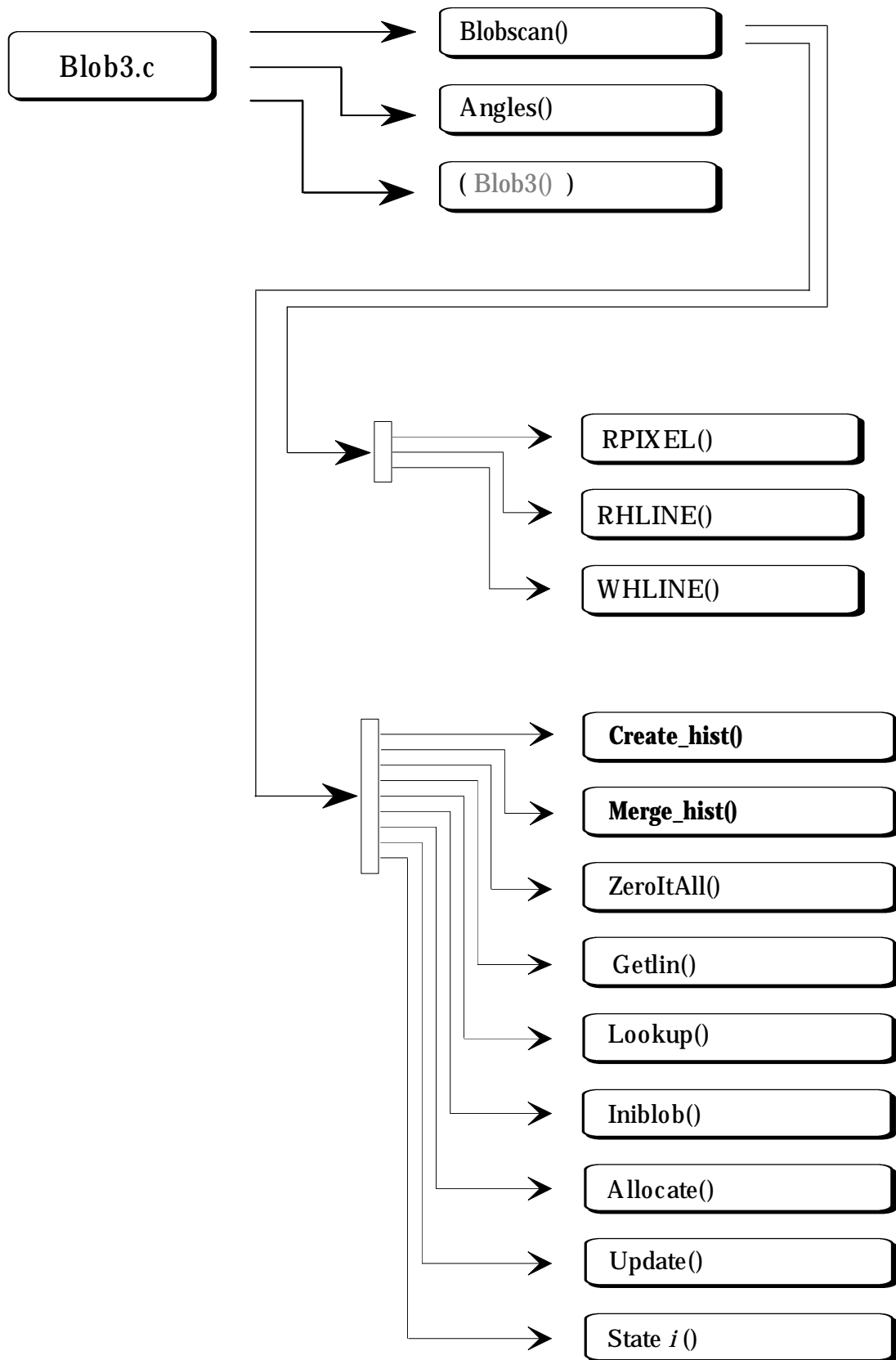


Fig. 3 - Esquema de chamadas do programa de segmentação a 3D.

I.2 b) O Significado das Funções no Programa de Segmentação a 3D

A tabela apresentada abaixo faz a descrição das várias funções que integram o programa de segmentação a 3D:

Blob3()	Programa principal
Blobscan()	Programa de varrimento da imagem e medida dos parâmetros geométricos
Angles()	Função de cálculo do ângulo de orientação no referencial da imagem, a partir dos momentos de segunda ordem (a conversão para o referencial do robot é feita dentro de 'Blob3.doc')
RPIXEL()	Função de leitura aleatória do conteúdo de pixeis da imagem
RHLINE()	Função de leitura de uma linha integral de pixeis da imagem
WHLINE()	Função de escrita de uma linha integral de pixeis na imagem
Create_hist()	Função de criação de um novo histograma 3D
Merge_hist()	Função responsável pela fusão de dois histogramas 3D quando o mesmo acontece aos respectivos "blobs".
ZeroItAll()	Inicialização de todas as variáveis globais
Getlin()	Função de aquisição de uma linha de pixeis com transformação destes em 0's ou 1's
Update()	Função de actualização dos parâmetros geométricos de uma figura ou "blob"
State <i>i</i> ()	Uma de 6 funções chamada sempre que se detecte uma transição na janela de 2×2 que varre a imagem; consoante a combinação de pixeis 0 ou 1 na janela, assim se identifica a situação e se chama a função adequada que altera o conteúdo de variáveis de controle ou efectua actualização de valores
Allocate()	Acréscimo de mais um "blob" com inicialização de alguns campo
Iniblob()	Inicialização dos campos dos novo "blob"; é chamado pela função anterior

Lookup()	Consulta do nº do "blob" válido a que o pixel pertence; esta verificação é necessária dada a constante fusão de "blobs" que invalida sempre um dos intervenientes; o vector <i>BP[]</i> mantém sempre informação dos "blobs" que absorveram outros, como se verá adiante
----------	--

I.3 b) Esquema de Chamadas do Programa de Sub-Segmentação

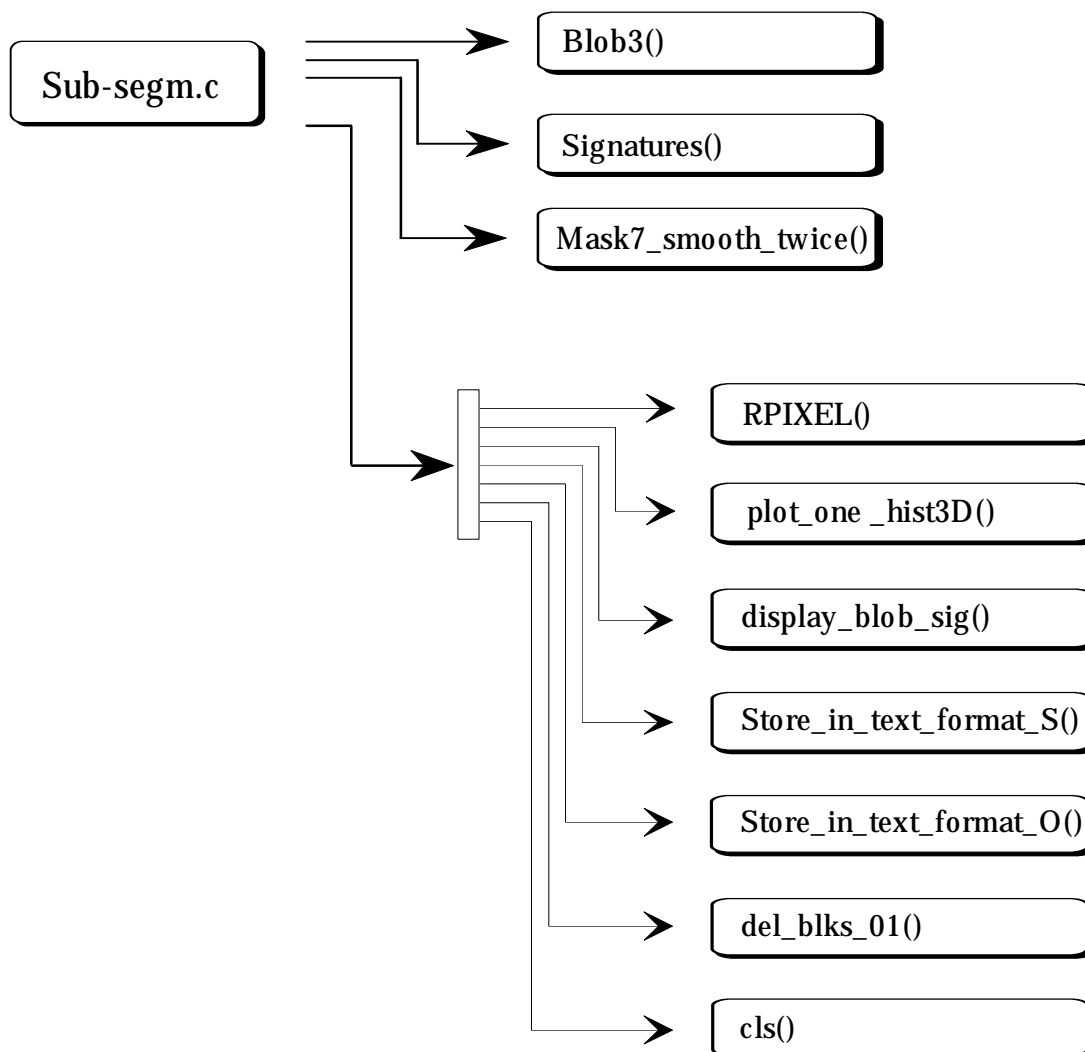


Fig. 5 - Esquema de chamadas do programa de sub-segmentação.

I.3 a) Descrição Esquemática do Programa de Sub-Segmentação

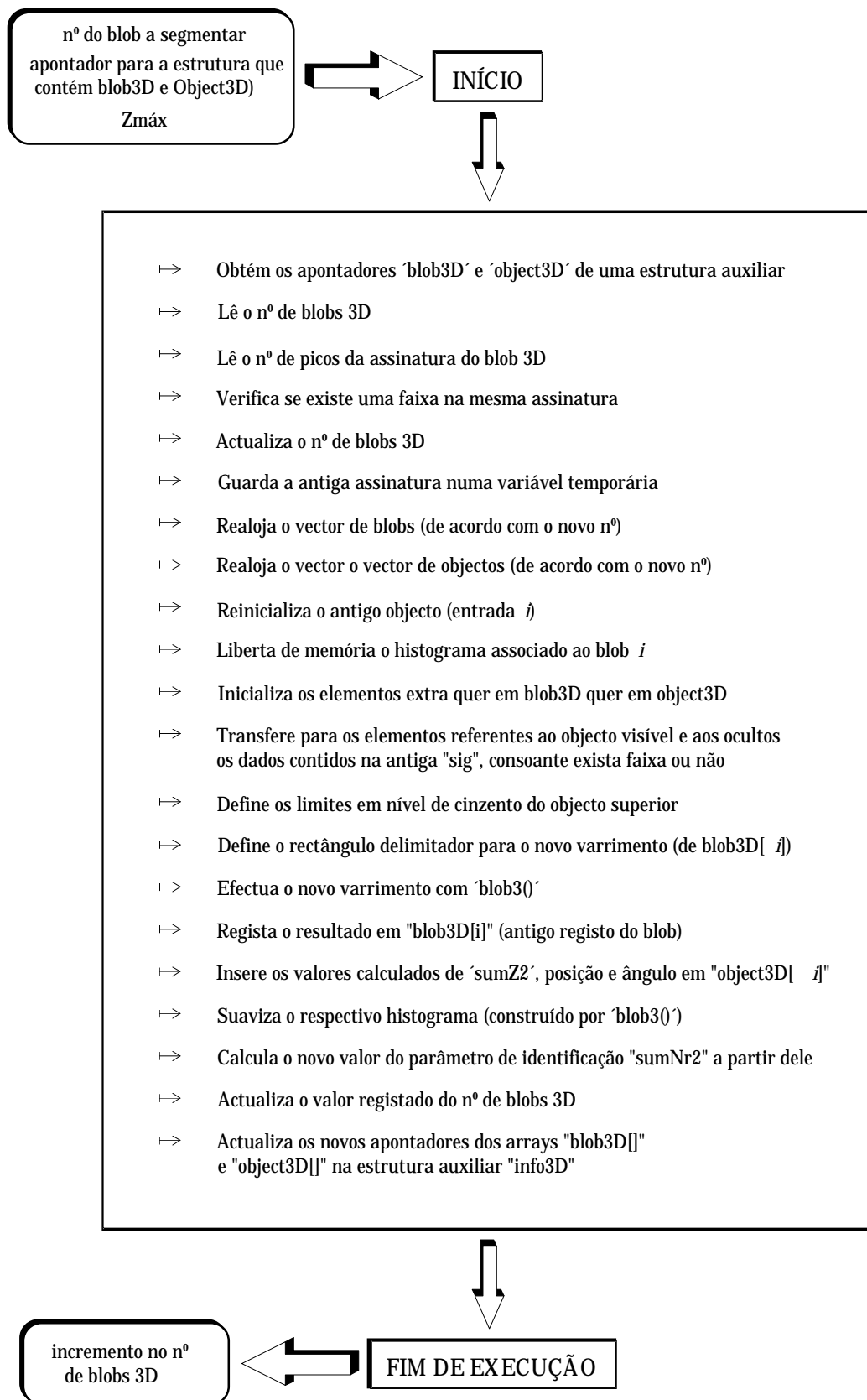


Fig. 4 - Descrição esquemática do programa de sub-segmentação.

Anexo J

J.1 Esquema de chamadas do Programa de Identificação a 3D

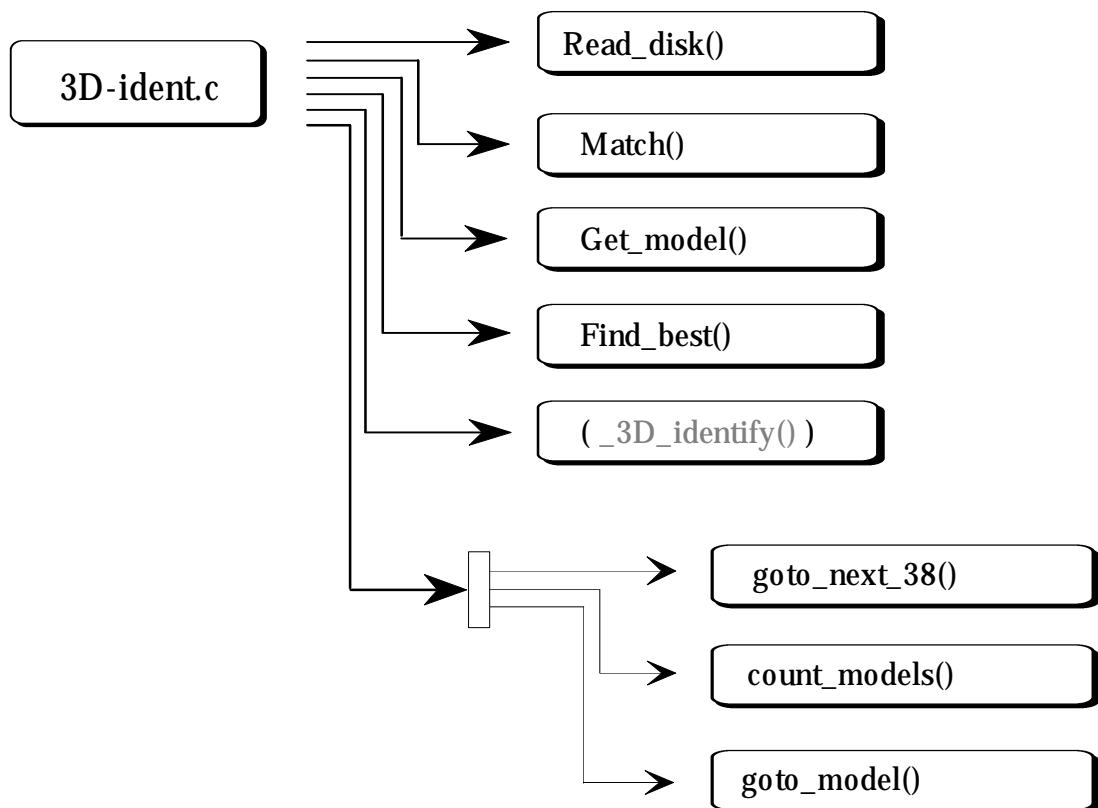


Fig. 1 - Esquema de chamadas do programa de identificação a 3D.

J.2 Descrição Esquemática da Função de Procura do Melhor Modelo

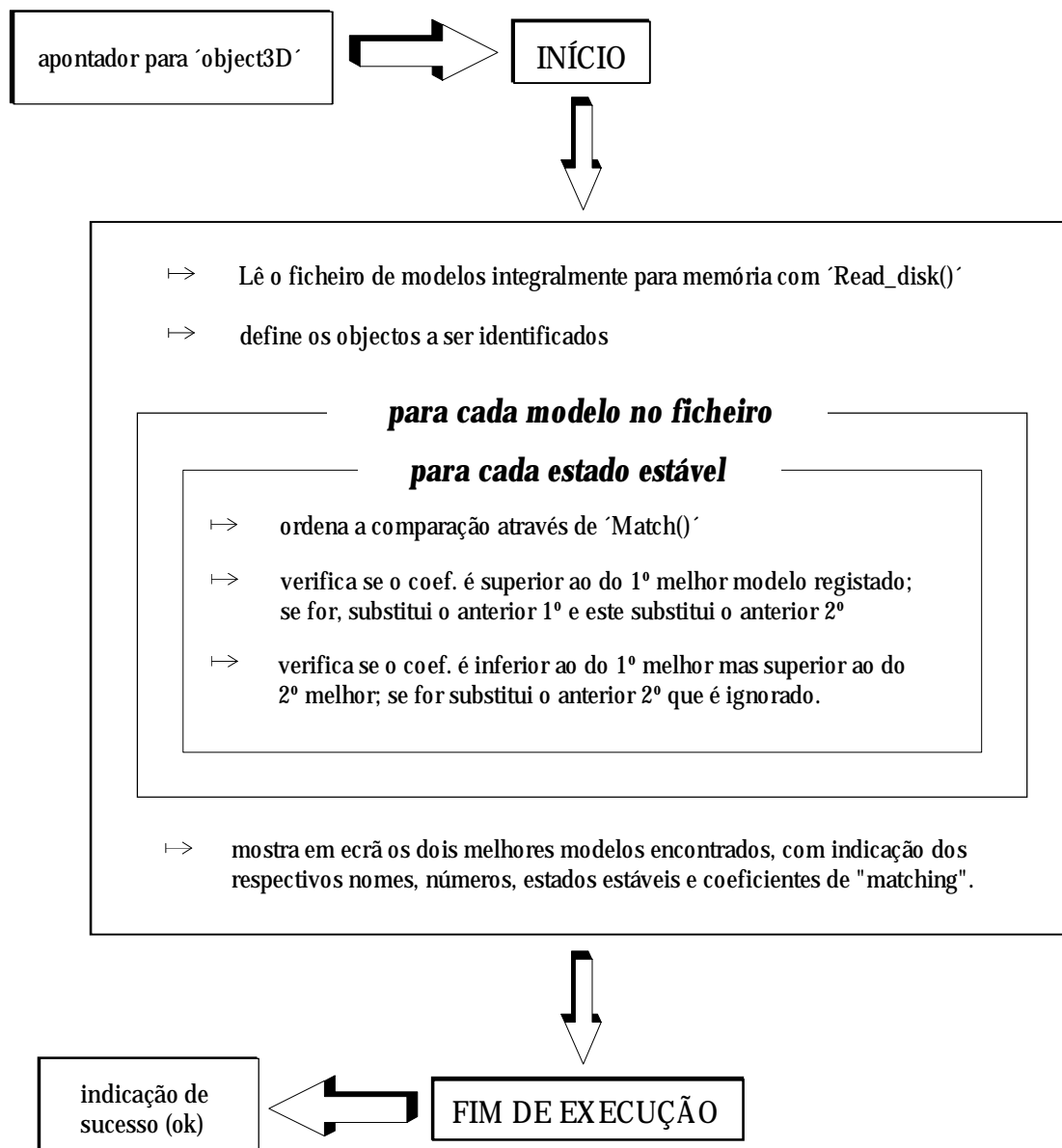


Fig. 2 - Descrição esquemática da função de procura do melhor modelo 3D.

Referências

REFERÊNCIAS

- [3Dscan96] Project 3DSCAN, Integration in Manufacturing, 3D Scanners Ltd., South Bank Technopark, London Road, London, SE1 6LN, U.K., 1996.
- [Andersen91] Andersen C.S., Sorensen J.J., "An Analysis of Three Depth Recovery Techniques", Acta da 7th SCIA, Aalborg, pp. 114-126, 1991.
- [Ballard82] Ballard D.H., Brown C.M., *Computer Vision*, Cap. 4, pp. 123-131, Prentice-Hall, 1982.
- [Ballard84] Ballard D.H., Parameter Nets, *Artificial Intelligence* 22, pp. 235-267, 1984.
- [Batista93] Batista J., Dias J., Araújo H., Traça A., "Monoplanar Calibration for Off-the-Shelf TV Cameras and Lenses Iterative Multi-Step Approach", Acta da 5th Portuguese Conference on Pattern Recognition (RecPad'93), pp. 53-62, 1993.
- [Bolles86] Bolles R.C., Horaud P., "3DPO: A three-dimensional part orientation system", *Int. Journal of Robotics Research*, Vol. 5, No. 3, pp. 3-20, 1986.
- [Bortolozzi91] Bortolozzi F., "Vision Trinoculaire: Une Solution Geometrique", Tese de Doutorado, UTC, Compiègne, França, 1991.
- [Boshra96] M. Boshra, H. Zhang, "An Efficient Pixel-Based Technique for Visual Verification of 3-D Object Hypotheses", Acta do IEEE International Conference on Robotics and Automation, Minneapolis, Minnesota, pp. 3472-3477, April 1996.
- [Breuel90] Breuel T.M., "Indexing for Visual Recognition from a Large Model Base", *A.I. Memo No. 1108*, Artificial Intelligence Laboratory, Institute of Technology, Massachusetts, 1990.
- [Brooks83] Brooks R.A., "Model-based three-dimensional interpretation of two-dimensional images", *IEEE Trans. Pattern Analysis and Machine Intelligence*, PAMI-5, no. 2, pp. 140-150, Março 1983.
- [Brown83] Brown C.M., "Inherent Bias and Noise in the Hough Transform", *IEEE Trans. on Pattern Anal. and Mach. Intell.*, Vol. 5, No. 5, pp. 493-505, Setembro 1983.

- [Caprile90] Caprile B., Torre V., "Using Vanishing Points for Camera Calibration", *Int. Journal of Computer Vision*, no. 4, pp. 127-140, 1990.
- [Clemens91] Clemens D.T., Jacobs D.W., "Space and Time Bounds on Indexing 3-D Models from 2-D Images", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(10), 1991.
- [Corazza91] Corazza A., "On the Hough Transform of a Curve: A Constructive Definition and Some Applications, Acta da 5th International Conference on Advanced Robotic, pp. 1752-1756s, 1991.
- [Cunningham81] Cunningham R., "Robotics Age", Vol. 3, No. 4, Julho-Agosto 1981.
- [Devy94] Devy M., Boumaza R., "Multi-Sensory Fusion and Model-Based Recognition of Complex Objects", *IEEE Int. Conf. on MFI'94*, pp. 345-352, 1994.
- [Dhome89] Dhome M., Richetin M., Lapreste J., Rives G., "Determination of the Attitude of 3-D Objects from a Single Perspective View", *IEEE Trans. on Pattern Anal. and Mach. Intell.*, 11(12), pp. 1265-1278, 1989.
- [Dias92] Dias J., Araújo H., Batista J., Almeida A., "Depth Recovery and Segmentation by Active Focus", Acta do 4th Portuguese Conference on Pattern Recognition (RecPad'92), pp.281-291, 1992.
- [Duda77] Duda R.O., Hart P.E., Use of the Hough Transform to Detect Lines and Curves in Pictures, Computer Methods in Image Analysis, *IEEE Press*, pp. 204-208, New York, 1977.
- [Fan89] Fan T., Medioni G., Nevatia R., "Recognizing 3-D Objects Using Surface Descriptions", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 11, No. 11, Novembro 1989.
- [Faugeras86] Faugeras O., Hebert M., "The representation recognition and locating of 3-D objects", *Int. Journal of Robotics Research*, Vol. 5, No. 3, pp. 40-52, 1984.
- [Faugeras87] Faugeras O., Toscani G., "The Calibration Problem for Stereo", Acta da *Int. Conf. of Computer Vision and Pattern Recognition*, 1987.
- [Fayek96] Fayek R.E., Wong A.K.C., "Fast 3D Object Modeling and Recognition in Range Images", *Vision Interface VI'96*, Toronto, Canada, pp. 33-40, Maio 1996.

- [Flynn92] Flynn P.J., Jain A.K., "3D Object Recognition Using Invariant Feature Indexing of Interpretation Tables", *Computer Vision, Graphics and Image Processing*, 55(2), pp. 119-129, 1992.
- [Fu87] Fu K.S., Gonzalez R.C., Lee C.S.G., "Robotics: Control, Sensing, Vision, and Intelligence", McGraw-Hill Book Company, 1987.
- [Grattoni91] P. Grattoni, G. Pettini, F. Pollastri, "Geometric Camera Calibration: a Comparison of Methods", Acta da *5th International Conference on Advanced Robotics (ICAR'91)*, Vol. 2, pp.1775-1779, 1991.
- [Grau96] Grau O., "A KnowledgeBased Scene Analysis System for the Generation of 3-D Models", *5th International Conference on Intelligent Systems*, Reno, E.U.A., Junho 1996.
- [Grimson84] Grimson W., Lozano-Pérez T., "Model-based Recognition and Localization from Sparse Range or Tactile Data", *Int. Journal of Robotics Research*, Vol. 3, No. 3, pp. 3-35, 1984.
- [Grimson90] Grimson W.E.L., Huttenlocher D.P., "On the Sensitivity of the Hough Transform for Object Recognition", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-12, No. 3, pp. 255-274, Março 1990.
- [Gruss92] Gruss A., Tada S., Kanade T., "A VLSI Smart Sensor for Fast Range Imaging", Acta do *1992 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Raleigh, NC, E.U.A., pages 349-358, Julho 1992.
- [Hattori92] Hattori T., Nakada M., Kataoka I., Kubo K., "High-Speed Image Processor for the Extraction of 3-D Depth Information from Image Sequence", Acta do *1992 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Raleigh, NC, E.U.A., pp. 341-348, Julho 1992.
- [Horaud84] Horaud P., Bolles R.C., "3DPO's Strategy for Matching Three-Dimensional Objects in Range Data", Acta da *Int. Conf. of Robotics*, Atlanta, E.U.A., pp. 78-85, Março 1984.
- [Horn84a] Horn B., "Extended Gaussian Images", em *Proc. IEEE*, Vol. 1, pp. 1656-1678, Dezembro 1984.
- [Horn84b] Horn B., Ikeuchi K., "The Mechanical Manipulation of Domly Oriented Parts", *Science America*, Vol. 251, No. 2, pp. 105-111, Agosto 1984.
- [Hough62] Hough P.V.C., "Method and Means for Recognizing Complex Patterns", patente U.S. 3069, 654, 18/Dez, 1962.

- [Ikeuchi81] Ikeuchi K., "Recognition of 3-D Objects using the Extended Gaussian Image", *Acta do 7th Int. Joint Conf. Artificial Intelligence*, Vancouver, B.C., Canadá, pp. 595-600, Agosto 1981.
- [Ikeuchi87] Ikeuchi K., "Precompiling a Geometrical Model into an Interpretation for Objects Recognition in Bin-picking Tasks", *Acta de DARPA Image Understanding Workshop*, pp. 321-339, Fevereiro 1987.
- [Illingworth88] Illingworth J., Kitler J., A Survey of the Hough Transform, *Computer Vision, Graphics and Image Processing*, Vol. 44, No. 1, pp. 87-116, 1988.
- [Jarvis83a] Jarvis R.A., "A Perspective on Range Finding Techniques for Computer Vision", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-5, No. 2, pp. 122-139, Março 1983.
- [Jarvis83b] Jarvis R.A., "A Laser Time-of-Flight Range Scanner for Robotic Vision", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-5, No. 5, pp. 505-512, Março 1983.
- [Kadono91] Kadono K., Asada M., Shirai Y., "Context-Constrained Matching of Hierarchical CAD-Based Models for Outdoor Scene Interpretation", *Acta do IEEE Workshop on Directions in Automated CAD-Based Vision*, pp. 186-195, 1991.
- [Korn87] Korn M.R., Dyer C.R., "3D Multiview Object Representations for Model-Based Object Recognition", *Pattern Recognition*, 20(1), pp. 91-103, 1987.
- [Lamdan88] Lamdan Y., Wolfson H.-J., "Geometric Hashing: A General and Efficient Model-Based Recognition Scheme", *Technical Report No. 368, Robotics Report No. 152*, New York University, Robotics Research Laboratory, Department of Computer Science, New York, 1988.
- [Lopes92] Lopes P, Oliveira E., Contributions to Increase Efficiency on the Hough Transform, *Acta do 2nd International Conference on Automation, Robotics and Computer Vision (ICARCV'92)*, pp. CV-2.5.1-2.5.5, Singapura, 1992.
- [Lopes93a] Lopes. P, Oliveira. E., Identification of Partially Occluded Objects using Object Chain Code, *Acta da IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'93)*, Yokohama, Japão, 1993.
- [Lopes93b] Lopes P. Oliveira E., "A Fast Identification Method for Partially Occluded Objects using a Modified Hough Transform and Segment

- Chain Code", Acta da *5th Portuguese Conference on Pattern Recognition (RecPad'93)*, pp. 88-96, Porto, Portugal, Maio 1993.
- [Lowe87] Lowe D.G., "Three-Dimensional Object Recognition from Single Two-Dimensional Images", *Artificial Intelligence*, 31(3), pp. 355-395, 1987.
- [Maitre86] Maitre H., "Contribution to the Prediction of Performances of the Hough Transform", *IEEE Trans. on Pattern Anal. and Mach. Intell.*, Vol. 8, No. 5, Setembro 1986.
- [Marszalec92] J.A. Marszalec, H.M. Keränen, "A Photoelectric Range Scanner Using an Array of LED Chips", Acta da *1992 IEEE International Conference on Robotics and Automation*, Nice, França, pp. 593-598, Maio 1992.
- [Martins81] Martins H.A., Birk J.R., Kelley R.B., "Camera Models Based on Data from Two Calibration Planes".
- [Nakazawa94] Nakazawa K., Oshita T., "Position Control of Vision Based Robot Hand – Fusion of 2-D Image and Discrete Range Image by Multi-Agent", *IEEE Int. Conf. on MFI'94*, pp. 353-359.
- [Nevatia77] Nevatia R., Binford T., "Description and Recognition of Complex-Curved Objects", *Artificial Intelligence*, Vol. 8, pp. 77-98, 1977.
- [Nitzan77] Nitzan D., Brain A.E., Duda R.O., "The Measurement and Use of Registered Reflectance and Range Data in Scene Analysis", Acta do *IEEE*, Vol. 65, pp. 206-220, Fevereiro 1977.
- [Nitzan88] Nitzan D., "Three-Dimensional Vision Structure for Robot Applications", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 10, No. 3, Maio 1988.
- [Noble69] Applied Linear Algebra, B. Noble, Prentice-Hall, Englewood Cliffs, N. J., 1988.
- [Oshima83] Oshima M., Shirai Y., "Object recognition using three-dimensional information", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 3, No. 4, pp. 350-361, Julho 1983.
- [Parvin92] Parvin B., Medioni G., "B-rep from Unregistered Multiple Range Images", Acta do *1992 IEEE International Conference on Robotics and Automation*, Nice, França, pp. 1602-1606, Maio 1992.
- [PcVision89] Manual de utilização da placa de aquisição de imagem, "PcVision plus Frame Grabber User's Manual", Imaging Technology Inc., Maio de 1989.

- [Pollastri89] Pollastri F., "Metodo per la Misura della Posizione del Punto Focale in un Sistema di Riprese di Immagini", Patente No. 67782-A/89, Setembro 1989.
- [Ramos89] Ramos C., Oliveira E., "Detection of Straight Lines, Circles and Ellipses in an Image using the Hough Transform", Acta do *Portuguese Conference on Pattern Recognition (RecPad'89)*, pp. II.12-17, Porto, 1989.
- [Rioux89] Rioux, M., Blais, F., Beraldin, J., Boulanger, P., "Range Imaging Sensors Development at NRC Laboratories", Acta do *Workshop on Interpretation of 3-D Scenes*, pp. 154-160, IEEE Press, Austin, TX, Novembro de 1989.
- [Rosenfeld69] Rosenfeld A., *Picture Processing by Computer*, Academic Press, New York, 1969.
- [Roux94] Roux M., McKweon D.M., "Feature Matching for Building Extraction from Multiple Views", *IEEE Computer Vision and Pattern Recognition*, pp. 46-60, 1994.
- [Shapiro79] Shapiro S.D., Iannino A., "Geometric Construction for Predicting Hough Transform Performance", *IEEE Trans. on Pattern Anal. and Mach. Intell.*, Vol. 1, No. 3, pp. 310-317, Julho 1979.
- [Shirai83] Shirai Y., "Three-Dimensional Computer Vision", Springer-Verlag, 1983.
- [Siegel92] Siegel M.W., Leary M.L., "Range from Focus Pixel-by-Pixel", Acta da *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Raleigh, NC, E.U.A., pp. 2035-2040, Julho 1992.
- [Silva94] Silva, Jorge A., "Aquisição e Processamento de Informação Tridimensional", Tese de Doutorado publicada na Faculdade de Engenharia da Universidade do Porto, 1994.
- [Stein91] Stein F., Medioni G., "Structural Hashing: Efficient Three Dimensional Object Recognition", Acta da *Image Understanding Workshop*, 1991.
- [Sklansky78] Sklansky J., "On the Hough Technique for Curve Detection", *IEEE Trans. Comp.*, Vol. 27, No. 100, pp. 734-736, Outubro 1978.
- [Strang88] *Linear Algebra and its Applications*, Gilbert Strang, 3rd edition, Harcourt Brace Jovanovich Publishers, 1988.

- [Svalbe89] Svalbe I.D., "Natural Representation for Straight Lines and the Hough Transform on Discrete Arrays", *IEEE Trans. on Pattern Anal. and Mach. Intell.*, Vol. 11, No. 9, pp. 941-950, Setembro 1989.
- [Tsai86] Tsai R.Y., "An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision", *IEEE Conf. on Computer Vision and Pattern Recognition*, Miami, pp.68-75, 1986.
- [Tsai87] Tsai R.Y., "A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology using Off-the-Shelf TV Cameras and Lenses", *IEEE Journal of Robotics and Automation*, Vol. RA-3, No. 4, Miami, Agosto 1987.
- [Tsai89] Tsai R.Y., "Synopsis of Recent Progress on Camera Calibration for 3D Machine Vision", *The Robotic Review*, O.Khatib, J.J. Craig, T. Lozano-Perez Editors, pp.147-159, The MIT Press, 1989.
- [Umeda94] Umeda K., Arai T., "Industrial Vision System by Fusing Range Image and Intensity Image", *IEEE Int. Conf. on MFI'94*, pp. 337-344.
- [Umeda96] Umeda K., Ikushima K., Arai T., "Fusion of Range Image and Intensity Image for 3D Shape Recognition", *Acta do IEEE International Conference on Robotics and Automation*, Minneapolis, Minnesota, pp. 680-685, Abril 1996.
- [Will71] Will, P.M., Pennington, K.S., "Grid Coding: A Preprocessing Technique for Robot and Machine Vision", *Artificial Intelligence 2*, pp. 319-329, 1971.
- [VanVeen81] Van Veen T.M., Groen F.C.A., "Discretization Errors in the Hough Transform", *Pattern Recognition*, Vol. 14, pp. 137-145, 1981.
- [Yuen89] Yuen H.K., Illingworth J., Kittler J., "Detecting Partially Occluded Ellipses Using the Hough Transform", *Image and Vision Computing*, Vol. 7, No. 1, 1989.
- [Yuta91] Yuta S., Suzuki S., Saito Y., Iida S., "Implementation of an Active Optical Range Sensor using Laser Slit In-Door Intelligent Mobile Robot", *IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS'91)*, pp. 415-420, Novembro 1991.

