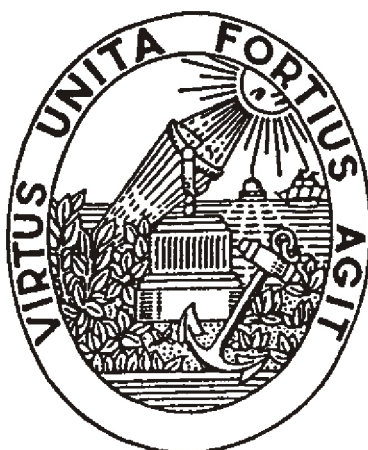


João Manuel Ribeiro da Silva Tavares

***Obtenção de Estrutura Tridimensional
a Partir de Movimento de Câmara***



Faculdade de Engenharia da Universidade do Porto

Junho de 1995

***Obtenção de Estrutura Tridimensional
a Partir de Movimento de Câmara***

Dissertação submetida ao Departamento de Engenharia Electrotécnica
e de Computadores para satisfação parcial dos requisitos do:

***Mestrado em Engenharia Electrotécnica
e de Computadores***

Perfil de Informática Industrial

Por:

João Manuel Ribeiro da Silva Tavares

Licenciado em Engenharia Mecânica pela Faculdade
de Engenharia da Universidade do Porto (1992)

Orientador:

A. Jorge Padilha

Prof. Auxiliar do Departamento de Engenharia Electrotécnica e de
Computadores da Faculdade de Engenharia da Universidade do Porto

J. TAVARES

Obtenção de Estrutura Tridimensional a Partir de Movimento de Câmara

1995

Agradecimentos

Ao Prof. A. Jorge Padilha por todo o apoio prestado ao longo desta dissertação; nomeadamente, pela sua orientação e total disponibilidade sempre demonstrada.

Ao Eng. Jorge Alves por todo o apoio prestado ao longo desta dissertação; em particular, nos domínios de Calibração de Câmaras e Extração de Coordenadas 3D.

Ao Eng. Miguel Velhote por todo o apoio prestado ao longo desta dissertação; principalmente, na disponibilização do texto sobre Detecção de Orlas de Intensidade do qual foi adaptada a introdução apresentada nesta publicação; assim como, pela disponibilidade de utilização de funções em linguagem C por si desenvolvidas.

À Junta Nacional de Investigação Científica e Tecnológica pela Bolsa de Mestrado atribuída ao abrigo do Programa Ciência com a ref. BM / 3258 / 92 - RM com a qual foi possível frequentar este Curso de Mestrado a tempo inteiro.

Ao Instituto de Engenharia Biomédica pelos recursos utilizados ao longo desta dissertação.

A todos aqueles que tornaram esta publicação possível ...

Sumário

A obtenção da estrutura tridimensional de cenas tem um papel deveras importante na visão por computador, sendo imprescindível em todos os sistemas para os quais é absolutamente indispensável a determinação das coordenadas 3D das entidades que constituem a cena. Como exemplos de tais sistemas podem ser referidos veículos e *robots* guiados autonomamente, seguimento e identificação de objectos, sistemas de inspecção automática, etc.

A abordagem global seguida e implementada ao longo desta dissertação é constituída pelas fases de calibração da câmara utilizada, de determinação das entidades a considerar em cada imagem da sequência, de seguimento das mesmas ao longo da sequência dada e, finalmente, de obtenção de coordenadas 3D de pontos particulares de cada entidade considerada, assim como de uma aproximação da profundidade dos restantes pontos.

Na fase da determinação das entidades a considerar em cada imagem da sequência, optou-se por considerar estas como sendo segmentos de recta. Para a determinação destas entidades, empregou-se em primeiro lugar um detector de orlas de intensidade. Após a detecção, é executado o seguimento das várias linhas determinadas e efectuada a sua aproximação poligonal, utilizando faixas dinâmicas, ficando cada imagem reduzida a um conjunto de segmentos de recta. Admitindo como provável que segmentos de recta próximos e de direcções similares representem uma mesma entidade no mundo 3D, optou-se por fundir tais segmentos de recta. Deste modo, o conjunto original de entidades representativas de uma dada imagem da sequência em estudo reduz-se a um conjunto com um número de elementos significativamente menor.

Determinadas as entidades a considerar, estas necessitam de ser devidamente parametrizadas; na abordagem seguida, a parametrização é baseada na posição do ponto médio, na direcção e no comprimento de cada segmento de recta.

Na fase seguinte, as entidades consideradas necessitam de ser seguidas ao longo da sequência. Para tal, utiliza-se um filtro de Kalman independente, com um modelo de aceleração localmente constante, para cada variável de parametrização, de forma a estimar cada uma destas características na imagem seguinte a partir das medidas verificadas nas anteriores. Como medida de emparelhamento, utilizam-se distâncias normalizadas de Mahalanobis ou, quando desta forma o mesmo não é obtido com êxito, restrições geométricas.

Estando devidamente realizado o emparelhamento das entidades entre imagens sucessivas, torna-se então possível extrair as coordenadas 3D dos pontos iniciais e finais de cada entidade, através da resolução do respectivo sistema de projecção estéreo; após a determinação da verdadeira localização dos pontos inicial e final do respectivo segmento de recta, visível numa dada imagem, sobre o segmento visível na imagem anterior a esta e com o qual foi emparelhado. Desta forma, é ultrapassado o possível problema de um mesmo segmento de recta no espaço 3D ser visível de modo diferente nas duas imagens consideradas.

No seguimento é utilizado um modelo 2D de entidades cujos elementos estão continuamente actualizados, não existindo, assim, a possibilidade de os recursos utilizados crescerem de forma exagerada e não controlada.

Na dissertação é apresentada, detalhadamente, cada uma destas fases, assim como exemplos de resultados obtidos em situações reais, algumas conclusões finais e perspectivas de desenvolvimento futuro da metodologia seguida.

A par da dissertação foi ainda realizada uma publicação que é basicamente um manual do utilizador de todas as implementações desenvolvidas.

Summary

The acquisition of the three dimensional structure of a scene plays a major role in computer vision, where the knowledge of the 3D co-ordinates of specific scene components is often required. Examples of such instances occur in autonomously guided vehicles and robots, in the tracking and identification of moving objects, in automatic visual inspection, and so on.

The general approach adopted in this thesis for capturing the three dimensional structure of a scene from a monocular image sequence includes the camera calibration, the detection of the selected entities in each frame of the sequence, the tracking of those entities along the sequence, and, finally, the computation of the 3D co-ordinates of specific points in the entities and the estimation of the depth co-ordinate of the other points.

In the entity detection phase, straight line segments were considered. The segments in each image of the sequence are detected by using a good intensity edge detector. Line following is performed on the grey level outputs of the edge detector, considering both edge magnitude and direction, while a polygonal approximation, using dynamic strips, is concurrently achieved. Moreover, it is assumed that 2D segments which are very close and have similar directions belong to the same 3D entity, so a line merging step is also included in the process. As a result, the significance of each line segment in the sequence is enhanced, while the number of segments is reduced.

The detected 2D entities are then parameterised; the line segment parameters chosen are the midpoint position, the segment direction, and its length.

In the next phase the line segments are tracked along the image sequence. For that purpose, an independent Kalman filter for each line parameter is used, considering a locally constant acceleration model, so as to estimate each line parameter in the next image of the sequence, given the measurements made on the current and previous images. To measure the degree of match between predicted and measured values, the Mahalanobis distance between pairs of parameters is used; however, when matching does not succeed, a second measure, based on geometric constraints, is performed to attempt the matching.

When the entity matching along the sequence is completed, the 3D co-ordinates of each segment's endpoints are computed, by solving the appropriate stereo projection equations; prior to this step, the correct position of the endpoints in one image is determined on the corresponding line segment in the other image, by using epipolar lines.

For the segment tracking process a 2D entity model is used where its elements are continually updated, thus preventing the uncontrolled growth of the resources used.

Each of the above steps is described in detail in this thesis, together with results obtained for real image sequences. A user manual of the implemented methods has been published in a separate document.

Índice

Capítulo I INTRODUÇÃO À DISSERTAÇÃO PROPOSTA	1
1 - Introdução ao processamento de imagem e à visão por computador	3
2 - Introdução ao tema da dissertação	4
3 - Introdução à abordagem global concebida	6
4 - Estrutura organizativa da dissertação	9
5 - Contribuições inovadoras alcançadas ao longo desta dissertação	10
Capítulo II CALIBRAÇÃO DE CÂMARAS PARA UTILIZAÇÃO EM VISÃO TRIDIMENSIONAL	13
Introdução	15
1 - Apresentação de alguns métodos de calibração de câmaras	16
1.1 - Método de calibração considerando projecção perspectiva, um modelo ideal para a câmara, e pontos de calibração não coplanares	17
1.1.1 - Algumas observações	18
1.2 - Método de calibração considerando projecção perspectiva, distorção radial da lente (um único termo) e utilizando pontos de calibração não coplanares	19
1.2.1 - Introdução à abordagem seguida	19
1.2.2 - O modelo da câmara utilizado	19
1.2.2.1 - Transformação do sistema de coordenadas mundo para o sistema câmara	19
1.2.2.2 - Equações relacionando as coordenadas 3D mundo com as coordenadas 2D imagem	22
1.2.3 - A motivação da técnica de calibração utilizada	23
1.2.4 - Calibração utilizando um conjunto de pontos não coplanares	24
1.2.5 - Algumas observações	26
1.3 - Método de calibração considerando a distorção radial da lente (dois termos) e utilizando pontos de calibração não coplanares	26
1.4 - Método de calibração considerando a distorção radial da lente (um único termo) e utilizando pontos de calibração coplanares	27
1.4.1 - Calibração de uma câmara utilizando um conjunto de pontos coplanares	27
1.4.2 - Algumas observações	31
1.5 - Método de calibração considerando a distorção radial da lente (um único termo), um referencial de coordenadas na memória <i>frame</i> não necessariamente paralelo ao referencial imagem e utilizando pontos de calibração coplanares	32
1.5.1 - Introdução	32
1.5.2 - Modelo utilizado para a câmara	33
1.5.3 - Determinação da matriz de rotação	34
1.5.4 - Determinação das translações segundo os eixos x e y e do factor de incerteza horizontal s_x	37
1.5.5 - Determinação da translação segundo o eixo z , da distância focal f e do coeficiente de distorção radial k_f	38
1.5.6 - Determinação dos factores de escala de imagem, das coordenadas do centro da imagem na memória <i>frame</i> e dos desvios ortogonais do referencial imagem	39
1.5.7 - Calibração iterativa	40
1.5.5 - Algumas observações	42
1.6 - Método de calibração concebido e implementado no âmbito desta dissertação	42
1.6.1 - Determinação do factor de incerteza horizontal s_x	43
1.6.2 - Determinação das coordenadas do centro da imagem na memória <i>frame</i>	44
1.6.2.1 - Influência de uma determinação incorrecta do centro da imagem na restrição do alinhamento radial	44
1.6.2.2 - Método para a determinação das coordenadas correctas do centro da imagem na memória <i>frame</i>	46
1.6.2.2.1 - Outros métodos para a determinação das coordenadas correctas do centro da imagem na memória <i>frame</i>	47
1.6.3 - Método de calibração de todos os parâmetros intrínsecos e extrínsecos de uma câmara exceptuando o factor de incerteza horizontal s_x	48
1.6.3.1 - Algumas observações	49

2 - Determinação das coordenadas dos pontos de calibração na memória <i>frame</i>	49
2.1 - Alguns métodos existentes para a determinação das coordenadas dos pontos de calibração na memória <i>frame</i>	50
2.2 - Método para a determinação das coordenadas dos pontos de calibração na memória <i>frame</i> concebido e implementado no âmbito desta dissertação	52
3 - Implementações desenvolvidas*	53
3.1 - Calibração do factor de incerteza horizontal de uma câmara: <i>findsx</i>	55
3.2 - Calibração de uma dada câmara: <i>calcamera</i>	56
3.3 - Determinação das coordenadas na memória <i>frame</i> dos pontos de calibração: <i>calpoint</i>	58
3.4 - Simulador de uma dada câmara: <i>simcamera</i>	60
4 - Alguns resultados experimentais obtidos pelas implementações desenvolvidas	62
4.1 - Utilizando a implementação <i>calpoint</i> para a determinação das coordenadas na memória <i>frame</i> dos pontos de calibração considerados	62
4.2 - Utilizando a implementação <i>simcamera</i> para a simulação de câmaras de geometria interna conhecida	66
4.3 - Utilizando a implementação <i>calcamera</i> para calibração de câmaras	70
5 - Algumas conclusões finais	75
Capítulo III	INTRODUÇÃO À DETECÇÃO DE ORLAS DE INTENSIDADE
	SEGUIMENTO DE LINHAS E APROXIMAÇÃO POLIGONAL UTILIZANDO FAIXAS
	FUSÃO DE SEGMENTOS DE RECTA PRÓXIMOS E DE DIRECÇÕES SIMILARES
	77
Introdução	79
1 - Introdução à detecção de orlas de intensidade	79
1.1 - Diferenciação e operadores afins	81
1.1.1 - Gradiente de uma imagem	81
1.1.2 - Laplaciano de uma imagem	83
1.2 - Filtros para detecção de orlas de intensidade	84
1.2.1 - Critérios de desempenho	86
1.2.2 - O filtro utilizado no trabalho desta dissertação: Filtro de Deriche	86
1.3 - Operações complementares	87
1.3.1 - Supressão de não-máximos	87
1.3.2 - Limiarização	87
1.3.3 - Validação das passagens por zero	88
2 - Seguimento de linhas e aproximação poligonal utilizando faixas	88
2.1 - Métodos de aproximação poligonal de linhas utilizando faixas	88
2.1.1 - Método de aproximação poligonal de linhas utilizando faixas estáticas (Roberge)	89
2.1.2 - Método de aproximação poligonal de linhas utilizando faixas dinâmicas (Leung e Yang)	90
2.2 - Um método para o seguimento de linhas e conseqüente aproximação poligonal utilizando faixas dinâmicas	93
3 - Fusão de segmentos de recta próximos e de direcções similares	95
3.1 - Métodos para fusão de segmentos de recta próximos e de direcções similares	95
4 - Implementações desenvolvidas*	100
4.1 - Seguimento de linhas e conseqüente aproximação poligonal utilizando faixas dinâmicas: <i>dstrip</i>	101
4.2 - Desenho numa imagem dos segmentos de recta definidos num ficheiro: <i>showlines</i>	102
4.3 - Fusão de segmentos de recta próximos e de direcções similares: <i>linelink</i>	104
5 - Alguns resultados experimentais obtidos pelas implementações desenvolvidas	106
5.1 - Utilizando a implementação <i>dstrip</i> para o seguimento de linhas e conseqüente aproximação poligonal utilizando faixas dinâmicas	106
5.2 - Utilizando a implementação <i>showlines</i> para o desenho numa imagem dos segmentos de recta definidos num ficheiro	109

* Esta secção foi introduzida por razões de completude, mas não é essencial à compreensão das secções seguintes.

5.3 - Utilizando a implementação <i>linelink</i> para a fusão de segmentos de recta próximos e de direcções similares _____	109
6 - Algumas conclusões finais _____	112
Capítulo IV <i>SEGUIMENTO DE SEGMENTOS DE RECTA EM IMAGENS SUCESSIVAS E OBTENÇÃO DE INFORMAÇÃO TRIDIMENSIONAL</i> _____	115
Introdução _____	117
1 - Seguimento de segmentos de recta em imagens sucessivas _____	117
1.2 - Parametrisação utilizada para as entidades _____	119
1.3 - Filosofia utilizada para o seguimento das entidades em imagens sucessivas _____	119
1.3.1 - Filtro de Kalman _____	120
1.3.1.1 - Elementos do filtro de Kalman utilizado para a posição do ponto médio das entidades _____	121
1.3.1.2 - Elementos do filtro de Kalman utilizado para a direcção das entidades _____	123
1.3.1.3 - Elementos do filtro de Kalman utilizado para o comprimento das entidades _____	124
1.3.2 - Distância normalizada de Mahalanobis _____	125
1.3.3 - Restrições geométricas _____	126
1.3.4 - Fase de medição e emparelhamento _____	126
2 - Determinação das coordenadas 3D dos pontos de cada entidade previamente emparelhada em duas imagens distintas _____	131
2.1 - Princípio da triangulação estereoscópica _____	132
2.2 - Determinação da verdadeira localização dos pontos inicial e final de um segmento de recta, visíveis numa dada imagem, sobre o segmento de recta visível na imagem anterior a esta _____	133
3 - Implementações desenvolvidas* _____	136
3.1 - Determinação da área visível por uma dada câmara sobre um determinado plano: <i>simspace</i> _____	136
3.2 - Simulador de uma dada câmara móvel para segmentos de recta definidos no espaço tridimensional: <i>simoca</i> _____	138
3.3 - Obtenção de informação tridimensional a partir do movimento de uma câmara: <i>deep</i> _____	140
3.4 - Extracção de coordenadas 3D de pontos a partir das suas coordenadas em duas imagens distintas: <i>compdeep</i> _____	145
4 - Alguns resultados experimentais obtidos pelas implementações desenvolvidas _____	147
4.1 - Utilizando a implementação <i>simspace</i> para a determinação da área visível por uma dada câmara sobre um dado plano _____	147
4.2 - Utilizando a implementação <i>simoca</i> para a simulação de uma câmara móvel, para segmentos de recta definidos no espaço tridimensional _____	148
4.3 - Utilizando a implementação <i>deep</i> para a obtenção de informação tridimensional a partir do movimento de uma câmara _____	149
4.4 - Utilizando a implementação <i>compdeep</i> para a extracção de coordenadas 3D de pontos a partir das suas coordenadas 2D em duas imagens distintas _____	156
5 - Algumas conclusões finais _____	157
Capítulo V <i>ALGUNS RESULTADOS EXPERIMENTAIS OBTIDOS EM SITUAÇÕES REAIS</i> _____	159
Introdução _____	161
1 - Resultados experimentais obtidos para a calibração da câmara _____	161
2 - Resultados experimentais obtidos para a determinação de informação tridimensional _____	168
2.1 - Situação real de uma câmara com movimento de translação pura _____	168
2.2 - Situação real de uma câmara com movimento de rotação pura _____	175
3 - Algumas conclusões finais _____	184

* Esta secção foi introduzida por razões de completude, mas não é essencial à compreensão das secções seguintes.

Capítulo VI CONCLUSÕES FINAIS E PERSPECTIVAS DE DESENVOLVIMENTO FUTURO	187
1 - Algumas conclusões finais	189
2 - Perspectivas de desenvolvimento futuro	190
Anexo TRANSFORMAÇÕES GEOMÉTRICAS EM 2D E EM 3D	193
1 - Transformações geométricas em 2D	195
1.1 - Representação matricial das transformações geométricas em 2D e coordenadas homogéneas	197
1.2 - Composição de transformações geométricas em 2D	201
2 - Representação matricial de transformações geométricas em 3D	202
2.1 - Composição de transformações em 3D	205
2.2 - Transformações para mudança de sistema de coordenadas	209
2.3 - Outras transformações geométricas em 3D	212
3 - Implementação*	216
3.1 - Alguns resultados experimentais	217
4 - Conclusão	217
Bibliografia	219

* Esta secção foi introduzida por razões de completude.

Capítulo I

INTRODUÇÃO

1 - Introdução ao processamento de imagem e à visão por computador

Uma elevadíssima percentagem dos seres vivos têm no seu sistema de visão o elemento sensorial mais importante para a sua sobrevivência e para as suas condições de vida. Esta importância do sistema de visão prende-se com a riqueza de informação que este faculto, não só em termos quantitativos, mas também qualitativos. Tais informações permitem, por exemplo, a detecção e o seguimento de certos alvos (predadores, alimento, etc.), a determinação de obstáculos na sua trajectória, em suma, informações sobre o ambiente que rodeia cada ser.

Com tamanha importância do sistema de visão não é surpreendente que o homem, e nomeadamente a sua comunidade científica, tenha nos últimos tempos realizado elevados esforços no sentido de prover sistemas automáticos, isto é sistemas computadorizados, que sejam capazes de executar funções do sistema de visão que são normalmente encontradas nos sistemas equivalentes dos seres vivos, e em especial no seu próprio. Esta tentativa de implementar certas funções do sistema de visão humana em sistemas automáticos pode ser realizada quer ao nível de *software*, quer ao nível de *hardware*. Surge, assim, uma área do desenvolvimento científico que é designada por processamento de imagem e visão por computador ou visão artificial.

O processamento de imagem e a visão por computador são normalmente divididos em quatro áreas de actuação:

- a) *melhoramento de imagens* - consiste basicamente na tentativa de melhorar e realçar subjectivamente certas características de uma dada imagem;
- b) *restauração de imagens* - consiste basicamente na tentativa de restaurar imagens que tenham sido degradadas na sua qualidade por um qualquer processo, como por exemplo distorção geométrica, movimento, etc.;
- c) *compressão de imagens* - consiste basicamente na tentativa de representar uma imagem original de forma mais simples e portanto mais leve, sem contudo perder informação necessária;
- d) *análise de imagens* - consiste basicamente na tentativa de descrever ou interpretar uma dada imagem ou sequência de imagens; isto é, na tentativa de entender uma imagem ou conjunto de imagens.

As três primeiras áreas costumam ser agrupadas na categoria do processamento de imagem; a última costuma ser atribuída à visão por computador e aparece geralmente associada à inteligência artificial.

Naturalmente que surgem várias situações em visão por computador que todas as áreas anteriores aparecem perfeitamente combinadas e integradas. Um exemplo desta combinação pode ser, por exemplo, um sistema que procure analisar o movimento de certos objectos a partir de uma sequência de imagens. Este sistema deverá incluir, quase obrigatoriamente, funções de melhoramento das imagens originais, de restauração das imagens originais, corrompidas por algum processo (como por exemplo ruído, más condições de iluminação, etc.), de análise das imagens originais e, porventura, pode ser utilizada compressão de imagens (por exemplo, para comprimi-las e assim poupar espaço de armazenamento, ou para diminuir tempos de transmissão).

Na actualidade, surgem cada vez mais aplicações do processamento de imagem e da visão por computador. Como exemplos de tais aplicações podem ser referidos os seguintes:

- *inspecção industrial* - na indústria, a qualidade do produto final tem cada vez mais um papel de importância primordial devido à actual competição feroz existente nos mercados. Como, geralmente, as funções de inspecção visual humana são bastante rotineiras, cansativas, morosas, e consequentemente originam frequentemente falhas e erros, surge a necessidade de as automatizar por sistemas computadorizados (utilizando, por exemplo, *robots* e manipuladores em tais tarefas). Evidentemente que é necessário prover estes sistemas de “visão”; surge, assim, uma área importante da visão por computador em que o objectivo é o

controlo dimensional de componentes, o controlo da qualidade superficial ou a verificação da integridade dos mesmos.

- *no guiamento de veículos autónomos* - cada vez mais se pretende substituir operários a cumprir funções pesadas e perigosas para a sua integridade física. Surgem assim veículos, *robots* ou manipuladores, que, possuindo sistemas de condução autónoma, se podem mover em ambientes hostis para o homem e transportando diferentes tipo de materiais e produtos. Se estes sistemas autónomos dispuserem de informação visual à cerca do ambiente que os rodeia, serão mais “*inteligentes*” e poderão guiar-se de forma mais correcta e segura, pois podem seguir a melhor trajectória possível e de forma mais rápida.
- *compressão de imagens* - quando se pretende armazenar um elevado número de imagens, torna-se essencial diminuir o volume da respectiva informação. Tal redução pode também ser necessária na transmissão de imagens, em que a largura de banda é inevitavelmente reduzida. Técnicas de compressão de imagem desempenham um papel fulcral em inúmeros sistemas de arquivo e comunicação de imagens.
- *aplicações médicas* - na medicina, existem bastantes imagens de diagnóstico obtidas por diferentes processos e técnicas (como por exemplo por raios X, termografia, endoscopia, etc.). Tais imagens necessitam de ser processadas no sentido de remover ruído, melhorar algumas características e analisá-las. A análise não é geralmente pretendida com um sentido perfeitamente autónomo mas como um auxiliar importante ao diagnóstico efectuado pelos especialistas. Não é, assim, surpreendente encontrar um elevado número de aplicações de processamento de imagem e de visão por computador, em medicina.
- *recuperação de imagens degradadas* - certas imagens são obtidas com uma inevitável deterioração; tal pode ser devido às más condições de iluminação, influência de campos eléctricos e/ou magnéticos, às elevadas distâncias de transmissão, etc. Nestas situações, é necessário realizar uma melhoria da qualidade das imagens. Por curiosidade, refira-se que uma das primeiras aplicações do processamento de imagem se deve com a recuperação das imagens enviadas para a Terra por um sonda espacial em 1960. A deterioração ficava-se a dever a elevadas restrições acerca do peso do sistema de visão [Lim, 1990], implicando assim que o sistema de imagem a bordo da sonda fosse de reduzida qualidade.
- *na meteorologia* - pela análise do movimento das nuvens, sistemas de visão por computador podem auxiliar na previsão do estado do tempo.
- *em sistemas de tráfego automóvel* - cada vez mais pretende-se dotar os sistemas de gestão de tráfego automóvel, actualmente existentes, com sistemas de visão por computador. Tal incorporação tem como objectivo tornar a gestão mais flexível, eficiente e rápida.
- *na agricultura* - na análise do crescimento e grau de maturação das plantações, a visão por computador, baseada em imagens de detecção remota, surge cada vez mais como um sistema bastante útil na mesma análise e controlo.

Os exemplos anteriores, apesar de serem em número reduzido, evidenciam perfeitamente a utilidade do processamento de imagens e da visão por computador no dia a dia da humanidade.

O tema da presente dissertação insere-se no domínio da visão por computador. Na próxima secção deste capítulo, é realizada a apresentação deste tema.

2 - Introdução ao tema da dissertação

Das informações que o sistema de visão proporciona ao Homem uma, que indiscutivelmente tem um papel de elevada importância, prende-se com a informação tridimensional sobre o ambiente que o rodeia. Só com esta informação tridimensional o homem pode saber se pode continuar a seguir uma mesma trajectória sem chocar com nenhum obstáculo, a que distância se encontra de um dado objecto ou ser, se o mesmo se está a aproximar ou a afastar de si, como pegar num certo objecto, etc.

Assumindo este tipo de informação um papel tão importante para o homem, é natural que este tente dotar os sistemas automáticos, por si desenvolvidos, com capacidades mais ou menos elaboradas de forma que possam dispor de tal informação. Deste modo, uma das áreas da visão por computador que tem recebido nos últimos tempos vastas contribuições da comunidade científica, é a obtenção de estrutura, ou informação, tridimensional acerca das cenas.

Existem actualmente várias técnicas para tornar possível esta obtenção de informação tridimensional utilizando visão por computador. Normalmente, as técnicas existentes são divididas em duas categorias¹:

- a) *activas* - as técnicas associadas a esta categoria envolvem a projecção controlada de energia (como por exemplo luz, ultra-sons, etc.) sobre a cena a considerar. A energia reflectida é detectada por sensores que, directa ou indirectamente, fornecem informação tridimensional.
- b) *passivas* - as técnicas associadas a esta categoria não envolvem a projecção de energia como as da categoria anterior. Desta forma, usam a iluminação ambiente existente sendo portanto técnicas de uso mais geral. Contudo, a utilização destas técnicas é geralmente mais difícil.

Como exemplos de técnicas activas podem ser referidos os seguintes:

- *estereoscopia activa* - consiste em projectar um padrão de luz (como, por exemplo, um feixe, um plano ou um conjunto de planos de luz) sobre a cena em questão e obter uma imagem de intensidades com a câmara afastada da fonte de luz utilizada. De seguida, deve-se estabelecer as correspondências entre os pontos do padrão de luz projectado e os obtidos na imagem. Finalmente, é possível calcular as coordenadas 3D usando as equações de projecção do ponto 3D em questão, para a fonte de luz e para a câmara utilizadas. Tais equações formam um sistema linear² sobredimensionado que pode ser resolvido em ordem às coordenadas 3D do respectivo ponto.
- *interferometria* - consiste em projectar sobre a cena riscas de sombra igualmente espaçadas. De seguida, deve-se adquirir uma imagem da cena com uma câmara deslocada lateralmente em relação ao projector e em frente da qual está colocada uma grelha idêntica à usada para projectar as riscas. A imagem obtida apresenta os contornos (franjas) dos pontos que estão à mesma distância, mas não dá indicação acerca do sinal da variação de distância entre linhas de contorno adjacentes. Pode-se obter tal informação, deslocando os objectos num sentido conhecido ou usando uma segunda grelha. Esta técnica é usada essencialmente para medir distâncias relativas.
- *detecção de ecos* - consiste na projecção de energia (luz ou ultra-sons) sobre a cena e na medida do tempo que decorre entre a emissão e a recepção dessa energia, após reflexão na cena.

De um modo geral, as técnicas activas são aptas para utilização em cenas sem pontos característicos e quando comparadas com as técnicas passivas, das quais serão apresentados de seguida alguns exemplos, a extracção de pontos característicos e o estabelecimento de correspondências, vêm simplificados. Do mesmo modo, as mesmas técnicas, apresentam algumas dificuldades: quando se usam padrões não codificados ou a cena apresenta superfícies com descontinuidades, torna-se difícil estabelecer as correspondências; quando é utilizado um feixe ou um plano de luz, existe a necessidade de varrer a cena, provocando assim uma certa lentidão do processo e a possível ocorrência de imprecisões mecânicas; quando são utilizados vários planos de luz, o emparelhamento torna-se mais difícil principalmente em cenas com descontinuidades; surgem problemas na utilização destas técnicas nas partes ocultas da cena em questão.

Como exemplos de técnicas passivas para a obtenção de informação tridimensional de uma cena podem ser referidos os seguinte:

¹ Ver, por exemplo [Silva, 1994].

² Normalmente designado por sistema de projecção estéreo. O método utilizado na obtenção deste sistema, é normalmente referido por método estereoscópico.

- *utilizando duas câmaras (estereoscopia passiva)* - consiste em obter uma imagem por cada uma das duas câmaras que deverão estar posicionadas de forma afastada uma da outra. Estabelecer as correspondências entre pontos característicos das duas imagens anteriormente referidas. Determinar as coordenadas 3D usando as equações de projecção do ponto 3D em questão para cada uma das duas câmaras utilizadas, e o conhecimento da geometria do sistema.
- *utilizando uma única câmara* - é semelhante ao exemplo anterior; contudo, neste caso as duas imagens são obtidas pela mesma câmara em duas posições distintas.
- *forma com base em X* - procura-se extrair a informação de distância a partir de uma ou mais imagens monoculares de intensidade, obtidas com uma câmara fixa. Estas técnicas podem ser baseadas (X) no brilho, na perspectiva, na textura, na ocultação, ou noutras características que de forma indirecta traduzem a geometria da cena.

De um modo geral, estas técnicas passivas apresentam como principal vantagem o facto de serem de aplicabilidade geral, mesmo em ambientes naturais não controlados. Do mesmo modo, as mesmas técnicas, apresentam algumas desvantagens: dificuldade de aplicação, em caso de cenas sem pontos característicos; dificuldades no caso de existirem pontos que são vistos numa imagem mas não na outra; lentidão³ do processo na obtenção de informação 3D, ou quando se pretende informação densa.

O presente trabalho enquadra-se nas técnicas passivas de obtenção de informação tridimensional, pretendendo-se estudar e implementar um conjunto de ferramentas que formem uma solução global para a obtenção de estrutura tridimensional a partir do movimento conhecido de uma única câmara. Assume-se como perfeitamente conhecido o movimento da câmara utilizada; tal pode ser obtido a partir da informação disponibilizada por transdutores de posição montados na plataforma sobre a qual assenta a câmara considerada.

A aplicação-alvo do trabalho consistia no fornecimento de informações tridimensionais para o sistema de guiamento autónomo de um veículo acerca do ambiente, tipicamente industrial, no qual este estaria inserido. Como em ambientes industriais, tipicamente existem em grande número segmentos de recta - bem como linhas curvas que podem ser aproximadas, sem grande perda de informação, por segmentos de recta - originadas por arestas direitas de estruturas normalmente encontradas em tais ambientes, optou-se por considerar segmentos de recta como as entidades características elementares a analisar.

Note-se que o principal objectivo deste trabalho não foi desenvolver e implementar um sistema para a aplicação-alvo mas, apenas, o de estudar e implementar as ferramentas que tornem tal sistema possível.

3 - Introdução à abordagem global concebida

Quando se pretende obter informação tridimensional de uma cena, a partir de uma sequência de imagens obtidas por uma mesma câmara com movimento conhecido, torna-se indispensável o prévio conhecimento do modelo da câmara utilizada. Assim, é necessário o conhecimento prévio de como pontos no espaço tridimensional são transformados em pontos no plano imagem da câmara, para cada posição e orientação ao longo do respectivo movimento. Esta determinação do modelo da câmara - isto é, a determinação da geometria interna e das características ópticas (parâmetros intrínsecos) e/ou a orientação e posição 3D da câmara relativamente a um certo sistema de coordenadas mundo (parâmetros extrínsecos) - é designada por calibração da câmara. A calibração geralmente é realizada apenas uma vez e constitui a primeira fase da abordagem seguida nesta dissertação.

Após a calibração da câmara, é então possível obter-se a sequência de imagens desejada. Em cada imagem, surge a necessidade de determinar as entidades que irão ser consideradas. Na abordagem seguida optou-se por considerar as entidades, como já foi referido na secção anterior, como sendo segmentos de recta. Para a sua determinação decidiu-se empregar um detector de orlas de intensidade

³ Obviamente que esta dificuldade pode ser ultrapassada pela utilização de *hardware* dedicado.

(como, por exemplo, os detectores de Sobel, Roberts, Laplaciano, Laplaciano do Gaussiano, Canny, Deriche, ISEF, etc.), após o que é executado o seguimento das várias linhas determinadas e a consequente aproximação poligonal, utilizando faixas dinâmicas. Como, provavelmente, devido ao tipo de ambiente considerado, segmentos de recta que sejam próximos e de direcções similares representam uma mesma entidade no mundo 3D, optou-se por fundir segmentos de recta nessas condições. Deste modo, o conjunto de segmentos de recta original, representativo de uma dada imagem da sequência em estudo, reduz-se a um conjunto com um menor número de elementos. Contudo, com este procedimento, dois segmentos de recta que no mundo 3D correspondem a duas entidades distintas podem ser fundidos como se correspondessem a uma mesma entidade no mundo 3D (devido à disposição relativa dos segmentos no plano 2D imagem), o que origina uma fusão não desejável; no entanto, tal fusão será transitória, pois quando as duas entidades passarem a ser distintas no plano 2D imagem (devido ao movimento da câmara) a fusão já não será possível. Nestes casos uma das entidades é considerada, pelo processo de seguimento adoptado, como correspondente à entidade que era considerada até então e a outra é considerada como uma nova entidade. Com a fusão, em vez de se considerar um elevado número de entidades considera-se um número mais reduzido; no entanto, as entidades resultantes serão de características mais fortes e, assim, mais representativas. Como exemplos de vantagens da diminuição do número de entidades a considerar, podem ser citados: o tempo total de computação necessário é menor; entidades de reduzida importância são absorvidas por entidades de maior importância ou desprezadas; o seguimento das entidades ao longo da sequência em estudo fica facilitado, pois o número de possíveis candidatas ao emparelhamento em duas imagens sucessivas é menor.

Após a determinação das entidades a considerar em cada imagem da sequência, estas necessitam de ser devidamente parametrizadas; a parametrização escolhida é baseada na posição do ponto médio, na direcção e no comprimento, dos segmentos de recta.

Na fase seguinte, as entidades consideradas necessitam de ser seguidas ao longo da sequência, isto é, deve-se associar cada segmento de recta numa dada imagem com um segmento na imagem seguinte, de modo a corresponder adequadamente à mesma entidade no espaço tridimensional. Na abordagem adoptada utiliza-se um filtro de Kalman independente para cada variável de parametrização, de forma a estimar cada uma destas características, na imagem seguinte, a partir das verificadas nas imagens anteriormente consideradas. Desta forma, utilizam-se três filtros de Kalman independentes, sendo associado a cada um deles um modelo cinemático de aceleração constante localmente. A utilização de três filtros de Kalman independentes é possível pelo facto de as variáveis de parametrização não apresentarem qualquer relação entre si. Como medida de emparelhamento de um dado segmento de recta numa imagem com um segmento na imagem sucessiva utilizam-se distâncias normalizadas de Mahalanobis ou, quando desta forma o emparelhamento não é obtido com êxito, restrições geométricas.

Estando o emparelhamento das várias entidades consideradas em duas imagens sucessivas devidamente realizado, torna-se então possível extrair as coordenadas 3D dos pontos de cada entidade. Esta determinação é realizada apenas para os pontos extremos de cada entidade, através da resolução do sistema de projecção estéreo, após a determinação da verdadeira localização dos pontos inicial e final do segmento visível na segunda imagem sobre o segmento visível na primeira. Tal determinação é realizada utilizando-se linhas epipolares ou, quando deste modo não é possível, analisando as posições dos referidos pontos nas duas imagens a considerar. Desta forma, é ultrapassado o possível problema de um mesmo segmento de recta no espaço 3D ser visível de modo diferente nas duas imagens consideradas, não existindo, assim, uma correspondência verdadeira entre os pontos extremos dos segmentos de recta visíveis nas duas imagens. Determinadas as coordenadas 3D dos pontos inicial e final de cada segmento de recta é obtida uma aproximação da profundidade dos restantes pontos utilizando-se interpolação linear.

No seguimento das entidades ao longo da sequência é utilizado um modelo 2D de entidades; assim, sempre que uma nova entidade se torna visível, ela é inserida neste modelo com um factor de confiança de valor médio; sempre que uma entidade é novamente visível na imagem seguinte, este factor é aumentado de uma unidade até ao seu valor máximo admissível; se, pelo contrário, não é visível na imagem seguinte, (isto é, não é possível o seu emparelhamento devido, por exemplo, a um desaparecimento temporário ou definitivo, ou a um movimento da mesma entidade no plano 2D

imagem diferente do que estava a ser considerado até então pelos filtros de Kalman) este factor é diminuído em uma unidade até ao seu valor mínimo admissível, caso em que é removida do modelo 2D. Desta forma dispõe-se de um modelo 2D cujos elementos estão continuamente actualizados, não existindo, a possibilidade de os recursos utilizados crescerem de forma exagerada e não controlada.

Na Fig. 1 está representado o diagrama geral da abordagem global adoptada.

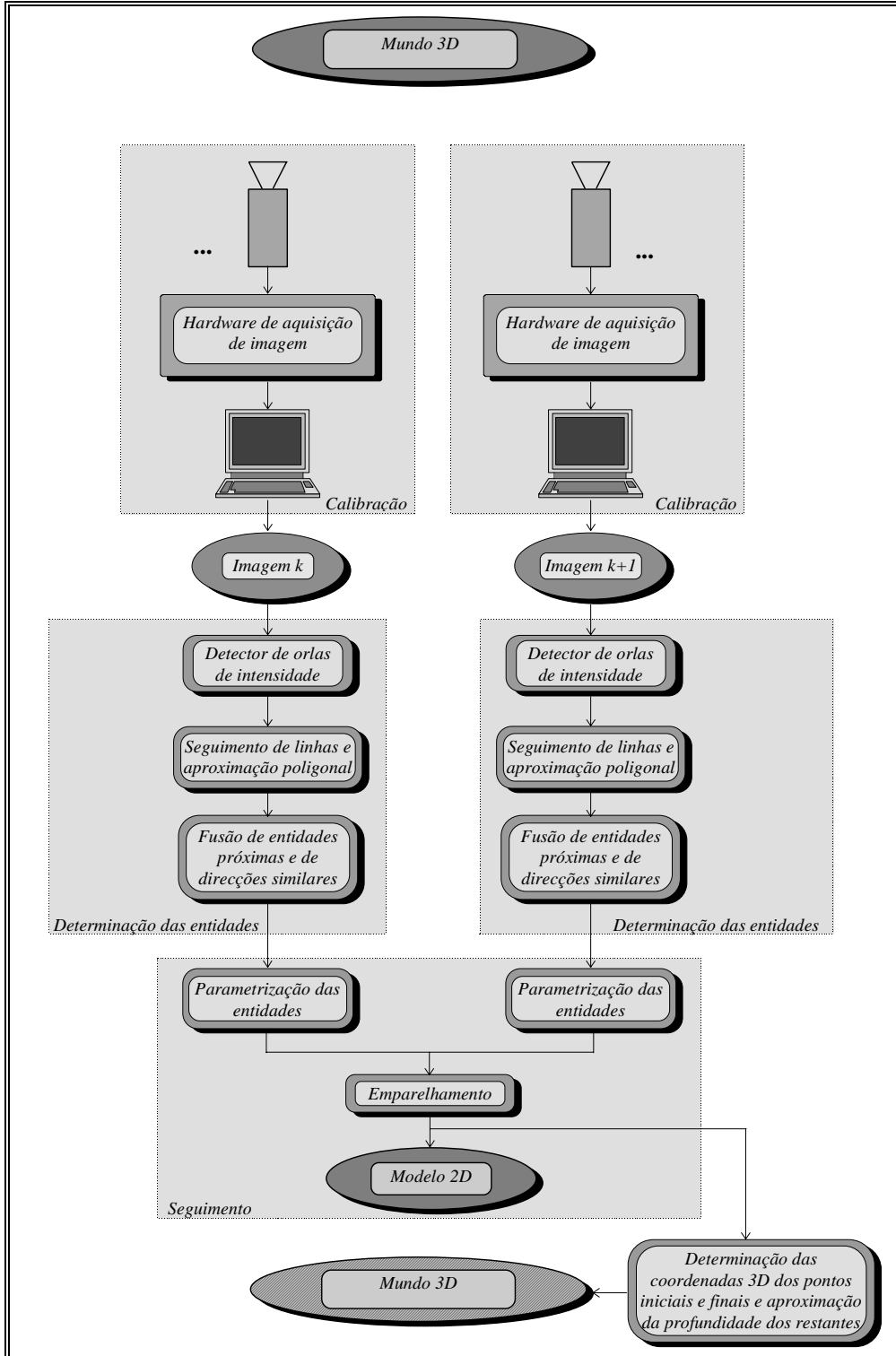


Fig. 1 - Diagrama representativo da abordagem global seguida nesta dissertação.

4 - Estrutura organizativa da dissertação

Pretendeu-se estruturar a dissertação em secções que se apresentem de forma completamente autónoma e independente; assim, as secções podem, por si só, ser consideradas e utilizadas independentemente em outros domínios da visão por computador.

Apresentam-se de seguida, de forma resumida, os restantes cinco capítulos e o único anexo desta publicação:

• *Capítulo II* || *Calibração de Câmaras para Utilização em Visão Tridimensional*

Neste capítulo é abordado o problema da calibração de câmaras nos vários domínios da visão por computador, nomeadamente no domínio da visão tridimensional. Este problema consiste basicamente na determinação da matriz matemática que transforma pontos de coordenadas 3D num sistema mundo em pontos de coordenadas 2D no sistema da memória *frame*. Vários métodos de calibração são apresentados, incluindo o concebido e implementado no âmbito desta dissertação.

Quando se pretende proceder a uma calibração, surge a necessidade de determinar as coordenadas 2D na memória *frame* dos pontos de calibração pretendidos; neste capítulo são apresentados vários métodos para esta determinação.

É também apresentado um método para geração automática das coordenadas 3D num sistema mundo e 2D na memória *frame* de um conjunto de pontos de calibração coplanares, para uma câmara de que se conhece a geometria interna, bem como a sua posição e orientação relativamente ao plano de calibração.

São ainda apresentadas as implementações desenvolvidas, assim como alguns resultados experimentais obtidos.

• *Capítulo III* || *Introdução à Detecção de Orlas de Intensidade* *Seguimento de Linhas e Aproximação Poligonal Utilizando Faixas* *Fusão de Segmentos de Recta Próximos e de Direcções Similares*

Neste capítulo é induzida uma solução global para a determinação numa dada imagem das entidades que irão ser objecto do seguimento ao longo da sequência; isto é, a determinação dos segmentos de recta mais significativos presentes em cada imagem.

A abordagem usada consiste em classificar os *pixels* que constituem a imagem original em termos de direcção e amplitude do gradiente - utilizando para o efeito um detector de orlas de intensidade -, realizar de seguida um seguimento das linhas presentes na mesma imagem e a subsequente aproximação poligonal e, finalmente, realizar fusões de segmentos de recta que provavelmente pertencerão a uma mesma entidade.

Assim, neste capítulo é apresentada uma introdução à detecção de orlas de intensidade, métodos de aproximação poligonal utilizando faixas, um método para o seguimento de linhas (baseado na direcção e em dois níveis de amplitude) e consequente aproximação poligonal utilizando faixas dinâmicas e métodos de fusão de segmentos de recta que sejam próximos e de direcção similares.

São ainda apresentadas as implementações desenvolvidas, assim como alguns resultados experimentais obtidos.

• *Capítulo IV* || *Seguimento de Segmentos de Recta em Imagens Sucessivas e Obtenção de Informação Tridimensional*

Neste capítulo é explanada uma abordagem global para o seguimento de segmentos de recta ao longo da sequência e para a consequente obtenção de informação tridimensional dos seus pontos extremos.

Para o seguimento é apresentado um método que utiliza uma parametrização das entidades baseada na posição do ponto médio, na direcção e no comprimento, três filtros de Kalman independentes (um para cada característica de parametrização) para a estimação das características das entidades ao longo da sequência e, como medida do emparelhamento, distâncias normalizadas de Mahalanobis ou, quando tal não é conseguido, restrições geométricas.

Para a obtenção de informação tridimensional, é apresentado neste capítulo um método que utiliza o princípio da triangulação estereoscópica e que determina, previamente, a verdadeira localização dos pontos inicial e final de um segmento de recta visível numa dada imagem sobre o segmento de recta visível na anterior a esta e com o qual foi emparelhado.

São ainda apresentadas as implementações desenvolvidas, assim como alguns resultados experimentais.

• **Capítulo V** || *Alguns Resultados Experimentais Obtidos em Situações Reais*

Após a apresentação da abordagem global utilizada nesta dissertação para a obtenção de estrutura tridimensional a partir do movimento de uma câmara, neste capítulo são apresentados dois exemplos de tal obtenção. No primeiro exemplo considerado, a câmara tem um movimento de translação pura e, no segundo, tem um movimento de rotação pura.

São assim, apresentados resultados experimentais obtidos para a calibração da câmara, para a determinação das entidades consideradas em cada imagem das sequências, e para a obtenção de informação tridimensional a partir do movimento da câmara.

• **Capítulo VI** || *Conclusões Finais e Perspectivas de Desenvolvimento Futuro*

Neste capítulo são apresentadas algumas conclusões finais sobre a abordagem global adoptada e sobre os resultados experimentais obtidos, bem como algumas perspectivas de desenvolvimento futuro.

• **Anexo** || *Transformações Geométricas em 2D e em 3D*

Neste anexo são introduzidas as transformações geométricas básicas em 2D e 3D utilizadas em vários domínios como, por exemplo, em computação gráfica e em visão por computador. O anexo encontra-se dividido em quatro secções: na primeira, são apresentadas as transformações geométricas em 2D; na segunda, em 3D; na terceira, é referida de forma muito sucinta uma implementação, desenvolvida no âmbito desta dissertação, capaz de aplicar as diversas transformações a pontos em 2D e em 3D, assim como alguns exemplos de resultados obtidos; na quarta e última secção, são apresentadas algumas conclusões.

A utilização de coordenadas homogéneas é devidamente justificada, assim como a utilização de matrizes para a representação das diversas transformações geométricas.

Informação complementar sobre a realização dos métodos descritos, de forma sumária, nos diversos capítulos⁴, pode ser obtida noutra publicação⁵, contendo todos os detalhes de implementação.

5 - Contribuições inovadoras alcançadas ao longo desta dissertação

São a seguir referenciadas as principais contribuições, com carácter inovador, obtidas ao longo do trabalho realizado no domínio desta dissertação:

• *No domínio da calibração de câmaras para utilização em visão tridimensional:*

- a) O método concebido e implementado para determinação das coordenadas, na memória *frame*, dos pontos de calibração considerados, é completamente inovador. Trata-se de um método onde é incluído seguimento baseado em direcção e amplitude, e aproximação linear para determinar as equações que definem as verdadeiras fronteiras das figuras geométricas consideradas, e que oferece resultados de boa qualidade, com precisão ao nível do *subpixel*.
- b) O método para determinação, de forma automática, das coordenadas 3D e 2D na memória *frame* de um conjunto de pontos coplanares, além de inovador, é uma solução óptima para a

⁴ As secções correspondentes, assinaladas por meio de um asterisco, foram introduzidas apenas por razões de completude, não sendo essenciais à compreensão das restantes secções dos respectivos capítulos.

⁵ [Tavares, 1995]

geração de dados, de forma artificial e perfeitamente controlada, para serem utilizados como teste de um qualquer método de calibração de câmaras sem ser necessário dispor do *hardware* e *setup* prático.

- c) O método de calibração do factor de incerteza horizontal, não sendo totalmente inovador, contém algumas alterações relativamente ao original.
 - d) O método de calibração de uma dada câmara, com excepção do factor de incerteza horizontal, não sendo completamente inovador pois é baseado em dois métodos independentes já existentes, é uma abordagem que unifica tais métodos e funciona de forma iterativa, fornecendo assim resultados de qualidade superior e constituindo uma boa solução global para a calibração de uma câmara.
- *No domínio da determinação das entidades a considerar em cada imagem:*
 - a) O método concebido e implementado no âmbito desta dissertação para o seguimento de linhas e consequente aproximação poligonal utilizando faixas dinâmicas, não é completamente inovador pois a parte referente à aproximação é conhecida; contudo, o método como um todo é inovador, pois considera o seguimento das linhas baseado em dois níveis de amplitude e na direcção, e integra automaticamente a aproximação poligonal de cada linha seguida, utilizando o método de aproximação poligonal baseado em faixas dinâmicas. A solução seguida para a actualização dos ângulos de cada faixa de aproximação também é diferente da versão original.
 - b) O método para a fusão de segmentos próximos e de direcções similares é de concepção original, constituindo uma boa solução para situações nas quais é útil reduzir-se o número de segmentos de recta originais, e nas quais é lícito considerar-se que segmentos muito próximos correspondem a distintos fragmentos de uma única entidade.
 - *No domínio do seguimento das várias entidades ao longo da sequência:*
 - a) O método para o seguimento das várias entidades ao longo da sequência apresenta algumas inovações, nomeadamente nas fase de medição e de emparelhamento; em particular, na utilização em conjunto de distâncias normalizadas de Mahalanobis e de restrições geométricas como medida do emparelhamento.
 - b) No método para a obtenção de informação tridimensional, a determinação prévia da verdadeira localização dos pontos inicial e final de um segmento de recta visível numa dada imagem sobre o segmento visível e emparelhado na imagem anterior constitui uma solução globalmente inovadora.

Capítulo II

CALIBRAÇÃO DE CÂMARAS PARA UTILIZAÇÃO EM VISÃO TRIDIMENSIONAL

Neste capítulo é abordado o problema da calibração de câmaras nos vários domínios da visão por computador, nomeadamente no domínio da visão tridimensional. Este problema consiste basicamente na determinação da matriz matemática que transforma pontos de coordenadas 3D num sistema mundo em pontos de coordenadas 2D no sistema da memória *frame*. Vários métodos de calibração são apresentados, incluindo o concebido e implementado no âmbito desta dissertação.

Quando se pretende proceder a uma calibração, surge a necessidade de determinar as coordenadas 2D na memória *frame* dos pontos de calibração pretendidos; neste capítulo são apresentados vários métodos para esta determinação.

É também apresentado um método para geração automática das coordenadas 3D num sistema mundo e 2D na memória *frame* de um conjunto de pontos de calibração coplanares, para uma câmara de que se conhece a geometria interna, bem como a sua posição e orientação relativamente ao plano de calibração.

São ainda apresentadas as implementações desenvolvidas, assim como alguns resultados experimentais obtidos.

Introdução

Nos vários domínios da visão por computador é, frequentemente, necessário proceder à calibração de uma câmara, nomeadamente, no caso da visão tridimensional. Entende-se por calibração de uma câmara o processo para a determinação da relação matemática que, para a respectiva câmara, transforma pontos 3D em pontos 2D, Fig. 1.

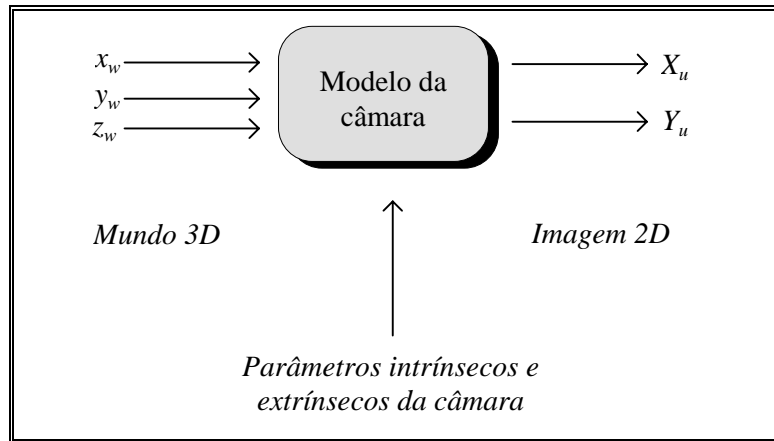


Fig. 1 - Transformação de um ponto no sistema de coordenadas 3D mundo para o sistema 2D imagem.

Deste modo, a calibração de uma dada câmara, no contexto de visão tridimensional, é o processo de determinação da geometria interna e das características ópticas (parâmetros intrínsecos), e/ou a orientação e posição 3D da câmara relativamente a um certo sistema de coordenadas mundo (parâmetros extrínsecos). Este processo pode ter os seguintes propósitos:

- Inferir informação 3D a partir das coordenadas imagem na memória *frame*. Por exemplo:
 - a) Informação 3D relativamente à localização de um objecto, alvo ou característica. Para simplificação, se o objecto é uma característica pontual (por exemplo, um ponto marcado num componente mecânico por um feixe laser ou um vértice de um componente electrónico numa placa de circuito impresso), a calibração da câmara fornece, a partir das coordenadas imagem na memória *frame*, uma solução para a determinação de uma recta de projecção no espaço 3D sobre a qual o referido ponto objecto deverá encontrar-se. Com duas imagens adquiridas, por duas câmaras ou por apenas uma câmara mas em duas localizações distintas, a posição do ponto objecto pode ser determinada pela intersecção das duas rectas de projecção respectivas - triangulação estereoscópica. As aplicações incluem medição de componentes mecânicos, montagem automática de componentes mecânicos ou electrónicos, seguimento, calibração de *robots* e análise de trajectórias. Nestas aplicações, a calibração dos parâmetros intrínsecos e extrínsecos da câmara é, geralmente, realizada uma única vez.
 - b) Informação 3D quanto à posição e orientação de uma dada câmara móvel (por exemplo, uma câmara montada num *robot* ou num veículo de navegação autónoma) relativamente a um sistema de coordenadas mundo. As aplicações incluem calibração de *robots*, com a câmara montada no próprio *robot*, e navegação de *robots*.
- Inferir as coordenadas 2D imagem na memória *frame* a partir de informação 3D. Nas aplicações baseadas em modelos e utilizando visão por computador (como por exemplo, aplicações de inspecção ou de montagem), uma hipótese do estado do mundo pode ser comprovada ou confirmada pela verificação de que as coordenadas imagem do objecto estão de acordo com a hipótese. Para esta verificação ser realizável, é necessário ter os parâmetros intrínsecos e extrínsecos do modelo da câmara calibrados, de tal modo que as coordenadas 2D imagem possam ser devidamente estimadas a partir da hipotética localização 3D do objecto.

As características que um sistema de calibração de câmaras deverá reunir são:

- *Autonomia*: O procedimento de calibração não deverá requerer a intervenção de operadores para fornecer ao sistema valores iniciais para certos parâmetros ou para escolher manualmente certos parâmetros do sistema.
- *Precisão*: Muitas aplicações, como a inspecção de componentes mecânicos, a montagem ou calibração de braços robóticos, requerem uma precisão bastante elevada. A técnica de calibração deverá ter a capacidade de atingir a precisão requerida. Esta exigência implica que o modelamento teórico do processo de aquisição de imagem seja preciso; deverá, por isso, incluir a distorção da lente e projecção perspectiva em vez da projecção paralela.
- *Razoável eficiência*: O processo completo de calibração de uma câmara não deverá incluir um procedimento não linear de elevada dimensão, isto é, com mais do que cinco equações não lineares. Para aplicações em que o objectivo é inferir informação 3D relativamente à posição e orientação de uma dada câmara móvel em relação a um sistema de coordenadas mundo, a velocidade de processamento da implementação torna-se crítica, devendo, por isso, a técnica de calibração ser adequada a uma implementação em tempo real.
- *Versatilidade*: O método de calibração deverá operar uniformemente e autonomamente para uma vasta gama de precisões requeridas, condições de *setups* ópticos existentes e aplicações.
- *Necessidade apenas de câmaras e lentes de utilização comum e habitualmente disponíveis*: A grande maioria dos métodos de calibração de câmaras, na área fotogramétrica, requerem câmaras e equipamento de processamento especiais e profissionais. Tais requisitos são proibitivos em automação, pois são demasiado morosos e trabalhosos na sua implementação. A vantagem da utilização de câmaras de estado sólido ou *vidicon* e de lentes de uso e disponibilidade corrente, são:
 - a) *versatilidade*: câmaras de estado sólido e lentes podem ser utilizadas nas mais diversas aplicações de automação;
 - b) *disponibilidade*: como as câmaras de estado sólido e as lentes de uso corrente são comuns nas mais diversas aplicações, elas são facilmente disponíveis, não existindo assim a necessidade de serem encomendadas aos fabricantes;
 - c) *familiaridade*: a sua operação é muito fácil, assim como a interface com computadores e sua instalação.

Este capítulo encontra-se dividido em cinco secções. Na primeira secção apresentam-se os fundamentos teóricos de alguns métodos de calibração existentes, assim como o que foi concebido realizado no âmbito desta dissertação; na segunda, são apresentados alguns métodos existentes para determinação das coordenadas imagem dos pontos de calibração, bem como o que foi adoptado neste trabalho; na terceira, são referidas sucintamente as várias implementações desenvolvidas; na quarta, são expostos alguns resultados experimentais obtidos; e, na quinta e última secção, são apresentadas algumas conclusões finais.

1 - Apresentação de alguns métodos de calibração de câmaras

Nesta secção são revistos seis métodos para calibração de câmaras. Assim, em primeiro lugar, é apresentado um método que utiliza um conjunto de pontos de calibração não coplanares, projecção perspectiva e um modelo ideal para a câmara. A seguir, é apresentado um método proposto em [Tsai, 1987] que utiliza um conjunto de pontos não coplanares, projecção perspectiva e um modelamento teórico para a câmara no qual é considerada a distorção radial da lente (modelada por um único termo) e um factor de incerteza horizontal (devido ao varrimento da câmara TV ou a erros de sincronismo na aquisição). Em terceiro lugar, é apresentado um método proposto em [Wu, 1988], em tudo idêntico ao anterior, mas no qual são considerados dois termos na modelação da distorção radial

da lente. Em quarto lugar, é apresentado um método proposto em [Tsai, 1987], idêntico ao segundo método apresentado, mas utilizando um conjunto de pontos coplanares. A seguir, é apresentado um método proposto em [Batista, 1993] que utiliza um conjunto de pontos de calibração coplanares e que, no modelamento teórico para a câmara, considera a distorção radial da lente (modelada por um único termo), dois factores de escala de imagem, um factor de incerteza horizontal (tal como no segundo e no quarto método), e a possibilidade de o sistema de coordenadas na memória *frame* não ser paralelo ao sistema no plano imagem. Em sexto e último lugar, é apresentado o método concebido e implementado no âmbito desta dissertação. Trata-se de um método cujo modelamento teórico é o mesmo do modelamento utilizado no segundo e no quarto método e que usa para a calibração um conjunto de pontos coplanares.

1.1 - Método de calibração considerando projecção perspectiva, um modelo ideal para a câmara, e pontos de calibração não coplanares

Este método de calibração de câmaras utiliza projecção perspectiva, um modelo perfeito para a câmara, isto é o modelo *pinhole*, e um conjunto de pontos de calibração não coplanares. A Fig. 2 representa a geometria básica do modelo da câmara.

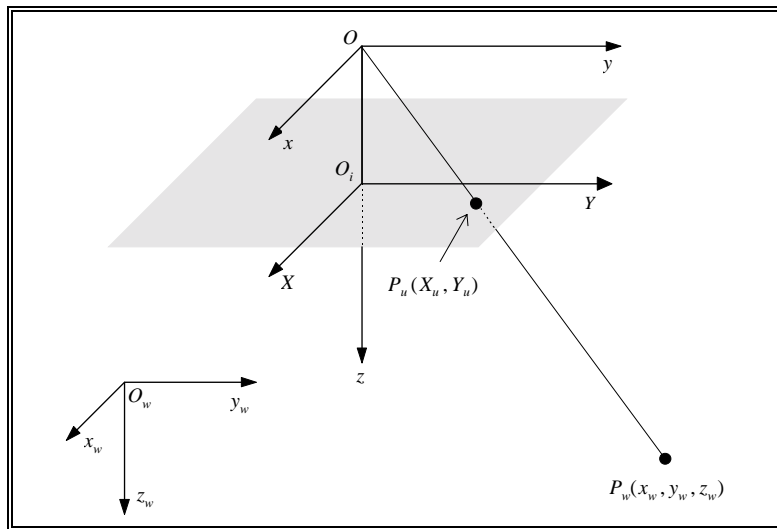


Fig. 2 - Geometria da câmara com transformação perspectiva.

Assim, nesta figura:

- $P_w(x_w, y_w, z_w)$ representa as coordenadas 3D do ponto P no sistema mundo;
- (x, y, z) representa o sistema de coordenadas câmara, o qual é centrado no ponto O (o centro óptico) e com o eixo z coincidente com o eixo óptico;
- (X, Y) representa o sistema de coordenadas imagem centrado em O_i , intersecção do eixo óptico z e o plano imagem anterior ao plano câmara, e com os eixos X e Y paralelos, respectivamente, aos eixos x e y do sistema câmara;
- $P_u(X_u, Y_u)$ representa as coordenadas 2D imagem do ponto P .

Utilizando este modelo para a câmara, um ponto P num sistema de coordenadas 3D mundo é transformado num ponto no sistema 2D imagem por uma dada transformação geométrica¹ M . Esta transformação geométrica corresponde à seguinte composição de transformações:

- a) rotação 3D do sistema de coordenadas mundo para o sistema câmara;

¹ Ver o anexo desta publicação: *Transformações Geométricas em 2D e em 3D*.

- b) translação 3D do sistema de coordenadas mundo para o sistema câmara;
- c) projecção perspectiva do sistema de coordenadas câmara para o sistema imagem;
- d) escalonamento do sistema de coordenadas imagem para o sistema na memória *frame*.

Deste modo, utilizando-se coordenadas homogéneas, para um qualquer ponto P tem-se $P_u = MP_w$ ou

$$\begin{bmatrix} wX_u \\ wY_u \\ w \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}. \quad (Eq. 1)$$

Para um dado conjunto de pontos não coplanares, cuja correspondência entre as coordenadas no sistema mundo e as coordenadas no sistema imagem é conhecida, a Eq. 1 permite a determinação dos elementos a_{ij} da transformação M , isto é, a calibração da câmara. No entanto, a Eq. 1 apresenta uma não unicidade dos elementos a_{ij} ; para esta ambiguidade ser desfeita, é necessário normalizar M ; assim, é normalmente atribuído ao elemento a_{34} o valor 1; outras normalizações são obviamente possíveis. Resolvendo a Eq. 1 obtém-se:

$$wX_u = a_{11}x_w + a_{12}y_w + a_{13}z_w + a_{14}, wY_u = a_{21}x_w + a_{22}y_w + a_{23}z_w + a_{24}, w = a_{31}x_w + a_{32}y_w + a_{33}z_w + a_{34}.$$

Substituindo o valor de w nas duas primeiras equações obtém-se:

$$a_{11}x_w + a_{12}y_w + a_{13}z_w + a_{14} - a_{31}x_wX_u - a_{32}y_wX_u - a_{33}z_wX_u - a_{34}X_u = 0,$$

$$a_{21}x_w + a_{22}y_w + a_{23}z_w + a_{24} - a_{31}x_wY_u - a_{32}y_wY_u - a_{33}z_wY_u - a_{34}Y_u = 0.$$

Escrevendo estas duas equações para todos os pontos não coplanares i considerados para a calibração, obtém-se o sistema de equações:

$$QB = 0, \quad (Eq. 2)$$

em que $B = [a_{11} \ a_{12} \ a_{13} \ a_{14} \ a_{21} \ a_{22} \ a_{23} \ a_{24} \ a_{31} \ a_{32} \ a_{33} \ a_{34}]^T$ e os vectores linha da matriz Q são:

$$q_{k-1} = [x_{w_i} \ y_{w_i} \ z_{w_i} \ 1 \ 0 \ 0 \ 0 \ 0 \ -x_{w_i}X_{u_i} \ -y_{w_i}X_{u_i} \ -z_{w_i}X_{u_i} \ -X_{u_i}]$$

$$q_k = [0 \ 0 \ 0 \ 0 \ x_{w_i} \ y_{w_i} \ z_{w_i} \ 1 \ -x_{w_i}Y_{u_i} \ -y_{w_i}Y_{u_i} \ -z_{w_i}Y_{u_i} \ -Y_{u_i}] \text{ onde } k \text{ é um índice par.}$$

Para se resolver o sistema de equações lineares sobredeterminado da Eq. 2, deve-se primeiro normalizá-lo como já anteriormente foi referido (fazendo por exemplo $a_{34} = 1$), de seguida passar a última coluna da Eq. 2 para o segundo membro da mesma e, por último, determinar as soluções pretendidas - utilizando por exemplo o método de regressão dos mínimos quadrados.

Saliente-se que, para utilizar este método de calibração, são necessários pelo menos seis pontos de calibração não coplanares; no entanto, para melhorar a precisão dos resultados obtidos, convém usar mais pontos de calibração.

1.1.1 - Algumas observações

Este método tem como grandes vantagens a sua simplicidade e a utilização de apenas equações lineares. Contudo, apresenta como grandes desvantagens: o facto de não entrar em conta com a distorção da lente, considerar que as coordenadas da imagem do centro óptico na memória *frame* coincidem com as coordenadas do centro desta - havendo até quem refira que, por estas duas desvantagens, este método não é adequado para utilização no domínio da visão tridimensional [Tsai, 1987]; considerar que todo o processo de aquisição, por exemplo em termos de sincronismo, é perfeito; os elementos da transformação M não são linearmente independentes entre si, pois para sete graus de liberdade - três ângulos de rotação, três translações e a distância entre o centro de projecção

e o plano de projecção (distância focal efectiva) - correspondem doze incógnitas no sistema da Eq. 2; e a necessidade de utilizar um conjunto de pontos de calibração não coplanares, que pode não corresponder à solução mais interessante e desejável em certas aplicações.

1.2 - Método de calibração considerando projecção perspectiva, distorção radial da lente (um único termo) e utilizando pontos de calibração não coplanares

Neste ponto é apresentado um método de calibração de câmaras proposto em [Tsai, 1987]. Neste método, é considerado, no modelamento teórico da câmara, projecção perspectiva e a distorção radial da lente (modelada por um único termo). A calibração é realizada utilizando, para o efeito, um conjunto de pontos de calibração não coplanares. Este método de calibração é a base de muitos outros métodos existentes, para os quais são utilizados pontos de calibração não coplanares.

1.2.1 - Introdução à abordagem seguida

A abordagem seguida neste método de calibração baseia-se na procura de uma restrição real ou equação, para redução da dimensão do espaço de parâmetros desconhecidos, que seja apenas função de um subconjunto dos parâmetros de calibração. Na realidade esta restrição existe e é designada por restrição do alinhamento radial [Tsai, 1987]. Esta restrição (ou as equações resultantes desta restrição física) é apenas função da rotação e da translação relativa, exceptuando a componente da translação segundo z , entre a câmara e os pontos de calibração. Além do mais, apesar da restrição ser uma função não linear dos parâmetros de calibração anteriormente referidos (designados por parâmetros do grupo I), existe uma solução fácil e eficiente para os calcular. Os parâmetros de calibração restantes (designados por parâmetros do grupo II) são calculados utilizando as equações normais de projecção. Uma boa aproximação inicial para as soluções dos parâmetros do grupo II pode ser obtida ignorando a distorção da lente e utilizando simples equações lineares com duas incógnitas. Os valores exactos para os parâmetros de calibração do grupo II podem ser calculados com uma ou mais iterações, minimizando o erro das equações de projecção perspectiva.

1.2.2 - O modelo da câmara utilizado

Este ponto descreve o modelo da câmara considerado neste método de calibração, define os parâmetros de calibração e apresenta o princípio da restrição do alinhamento radial, que é a base deste método. O modelo da câmara, por si só, é basicamente idêntico ao utilizado em inúmeros métodos de calibração existentes.

1.2.2.1 - Transformação do sistema de coordenadas mundo para o sistema câmara

A Fig. 3 representa a geometria básica do modelo da câmara considerada neste método. Assim, nesta figura:

- (x_w, y_w, z_w) representa as coordenadas 3D do ponto P no sistema mundo;
- (x, y, z) representa as coordenadas 3D do ponto P no sistema câmara, o qual é centrado no ponto O (o centro óptico) e com o eixo z coincidente com o eixo óptico;
- (X, Y) representa o sistema de coordenadas imagem centrado em O_i (intercepção do eixo óptico z e o plano imagem) e com os eixos X e Y paralelos, respectivamente, aos eixos x e y , do sistema câmara;
- f é a distância entre o plano imagem e o centro óptico, isto é a distância focal efectiva;
- (X_u, Y_u) representa as coordenadas imagem de P se for utilizado um modelo *pinhole* perfeito para a câmara, isto é as coordenadas imagem sem distorção;

- (X_d, Y_d) representa as coordenadas imagem reais, que diferem de (X_u, Y_u) devido à distorção da lente.

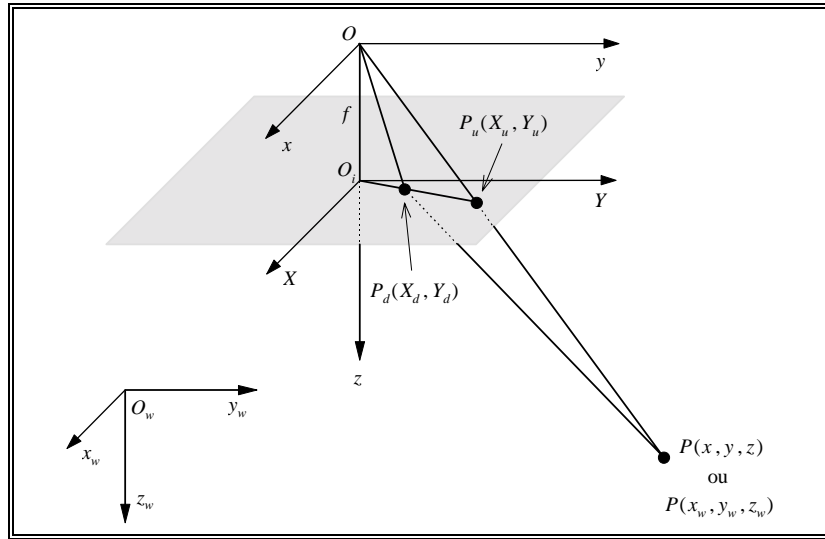


Fig. 3 - Geometria da câmara com projecção perspectiva e distorção radial da lente.

Contudo, como a unidade para as coordenadas utilizadas no computador (X_f, Y_f) é o número de *pixels* para a imagem discreta na memória *frame*, parâmetros adicionais necessitam de ser especificados e calibrados para relacionamento do sistema de coordenadas imagem com o sistema na memória *frame*. A transformação global de (x_w, y_w, z_w) para (X_f, Y_f) é representada na Fig. 4 por quatro passos. O quarto passo (a determinação do factor de incerteza horizontal s_x), é específico para aplicações industriais de visão por computador onde se utilizam câmaras TV, nomeadamente, quando são utilizadas câmaras CCD ou CID.

Apresenta-se de seguida, na forma analítica, a transformação para cada um dos quatro passos representados na Fig. 4.

Passo 1: Transformação corpo rígido das coordenadas (x_w, y_w, z_w) no sistema mundo para as coordenadas (x, y, z) no sistema câmara:

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = R \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} + T, \quad (Eq. 3)$$

onde R é a matriz rotação 3D $R = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix}$ e T é o vector translação 3D $T = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$.

Os parâmetros a calibrar são a matriz de rotação 3D R e o vector de translação 3D T .

Note-se que a transformação de corpo rígido de um sistema de coordenadas cartesiano (x_w, y_w, z_w) para outro sistema (x, y, z) é única se a transformação for definida como uma rotação 3D em torno da origem seguida por uma translação 3D.

Passo 2: Transformação das coordenadas 3D câmara (x, y, z) para as coordenadas imagem ideal (X_u, Y_u) , utilizando projecção perspectiva com geometria *pinhole* para a câmara:

$$X_u = f \frac{x}{z}, \quad Y_u = f \frac{y}{z}. \quad (Eqs. 4a e 4b)$$

O parâmetro a calibrar é a distância focal efectiva f .

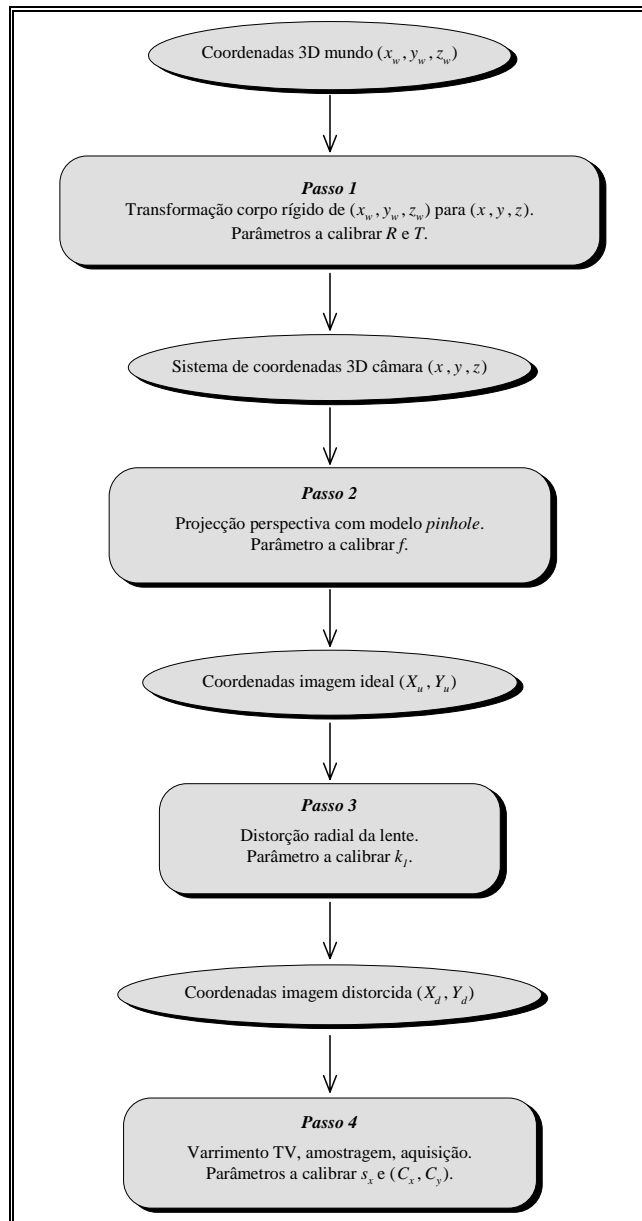


Fig. 4 - Transformação das coordenadas 3D mundo para as coordenadas imagem distorcida na memória frame. De notar que as coordenadas do centro da imagem na memória frame (C_x, C_y) não são obtidas, neste método, por calibração.

Passo 3: A distorção radial da lente é:

$$X_d + D_x = X_u, \quad Y_d + D_y = Y_u, \quad (\text{Eqs. 5a e 5b})$$

onde (X_d, Y_d) são as coordenadas imagem real ou distorcida no plano imagem, $D_x = X_d(k_1 r^2 + k_2 r^4 + \dots)$ e $D_y = Y_d(k_1 r^2 + k_2 r^4 + \dots)$ com $r^2 = X_u^2 + Y_u^2$.

Os parâmetros a serem calibrados são os coeficientes de distorção k_i . Existem dois tipos de distorção de uma lente: radial e tangencial. Para cada tipo de distorção, é necessária uma série infinita; contudo, experiências reportadas demonstram que, em aplicações industriais de visão por computador, apenas é necessário considerar a distorção radial e um só termo da respectiva série. Qualquer modelamento mais elaborado para a distorção da lente não só será desnecessário como provocará instabilidades numéricas [Tsai, 1987].

Passo 4: Transformação das coordenadas imagem real (X_d, Y_d) para as coordenadas imagem na memória *frame* (X_f, Y_f):

$$X_f = d_x^{-1} X_d + C_x, \quad Y_f = d_y^{-1} Y_d + C_y, \quad d_x' = s_x d_x, \quad (\text{Eqs. 6a, 6b e 6c})$$

onde:

- (X_f, Y_f) são as coordenadas discretas do ponto a considerar na memória *frame*;
- (C_x, C_y) são as coordenadas discretas do centro óptico na memória *frame*;
- s_x é o factor de incerteza horizontal;
- d_x é a distância entre centros dos elementos sensores vizinhos segundo a direcção x ;
- d_y é a distância entre centros dos sensores CCD vizinhos segundo a direcção y .

O parâmetro a calibrar é o factor de incerteza horizontal s_x .

Normalmente, os fabricantes de câmaras CCD fornecem informação relativamente a d_x e a d_y , com precisão de submicron; contudo, um parâmetro de incerteza horizontal necessita de ser introduzido. Esta necessidade é provocada por vários factores, como má sincronização temporal entre o *hardware* de aquisição de imagem e o de varrimento da câmara, ou de imprecisões no sincronismo de varrimento TV. Deste modo, foi introduzido um factor de incerteza horizontal s_x na Eq. 6c, sendo assim mais um parâmetro a ser calibrado.

1.2.2.2 - Equações relacionando as coordenadas 3D mundo com as coordenadas 2D imagem

Combinando os três últimos passos representados na Fig. 4, as coordenadas 2D imagem real (X_d, Y_d) estão relacionadas com as coordenadas 3D do ponto P no sistema câmara (x, y, z), pelas seguintes equações:

$$X_d(1+k_1r^2) = f \frac{x}{z}, \quad Y_d(1+k_1r^2) = f \frac{y}{z}. \quad (\text{Eqs. 7a e 7b})$$

Substituindo a Eq. 3 nas Eqs. 7a e 7b obtêm-se:

$$X_d(1+k_1r^2) = f \frac{r_1x_w + r_2y_w + r_3z_w + t_x}{r_7x_w + r_8y_w + r_9z_w + t_z}, \quad Y_d(1+k_1r^2) = f \frac{r_4x_w + r_5y_w + r_6z_w + t_y}{r_7x_w + r_8y_w + r_9z_w + t_z}. \quad (\text{Eqs. 8a e 8b})$$

Os parâmetros utilizados na transformação global, representada na Fig. 4, podem ser divididos nas duas classes seguintes:

I) Parâmetros extrínsecos: Os parâmetros utilizados no passo 1, para a transformação do sistema de coordenadas 3D mundo para o sistema 3D câmara, são designados por parâmetros extrínsecos. Existem seis parâmetros extrínsecos: os ângulos de rotação ϕ , θ e ψ , e as três componentes do vector de translação T . A matriz de rotação R pode ser representada em função de ϕ , θ e ψ , como:

$$R = R_{(x,\phi)} R_{(y,\theta)} R_{(z,\psi)}$$

$$= \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \cos \psi \sin \phi \sin \theta - \cos \phi \sin \psi & \sin \psi \sin \phi \sin \theta + \cos \phi \cos \psi & \sin \phi \cos \theta \\ \cos \psi \sin \theta \cos \phi + \sin \phi \sin \psi & -\sin \phi \cos \psi + \sin \psi \sin \theta \cos \phi & \cos \theta \cos \phi \end{bmatrix}. \quad (\text{Eq. 9})$$

II) Parâmetros intrínsecos: Os parâmetros utilizados nos passos 2 a 4, para a transformação das coordenadas 3D no sistema câmara para as coordenadas no sistema na memória *frame*, são designados por parâmetros intrínsecos. Existem cinco parâmetros intrínsecos:

- distância focal efectiva ou seja, a distância entre o centro de projecção e o plano imagem f ;
- coeficiente de distorção radial da lente k_1 ;
- factor de incerteza horizontal s_x ;
- coordenadas, em *pixels*, do centro óptico da imagem na memória *frame* (C_x, C_y).

1.2.3 - A motivação da técnica de calibração utilizada

Na base da técnica utilizada neste método de calibração estão as seguintes quatro observações:

Observação I: Como se assume que a distorção imposta pela lente é radial, independentemente do seu valor a direcção do vector $\overline{O_i P_d}$ - definido no plano imagem pela origem O_i e pela imagem do ponto P , (X_d, Y_d) - permanece constante e radialmente alinhada com o vector $\overline{P_{oz} P}$ - definido pelo ponto P_{oz} sobre o eixo óptico (cuja coordenada z é a mesma do ponto P) e pelo ponto P -, Fig. 5.

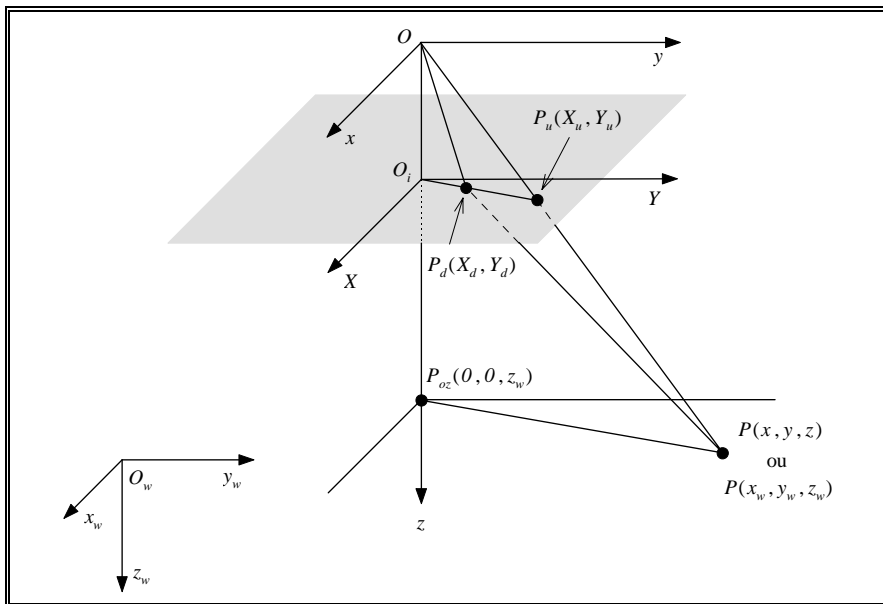


Fig. 5 - Ilustração da restrição do alinhamento radial. A distorção radial não afecta a direcção do vector $\overline{O_i P_d}$, o que origina: $\overline{O_i P_u} // \overline{O_i P_d} // \overline{P_{oz} P}$.

Observação II: A distância focal efectiva f também não afecta a direcção do vector $\overline{O_i P_d}$, pois f influencia as coordenadas imagem distorcida X_d e Y_d do mesmo modo.

Observação III: Uma vez que o sistema de coordenadas mundo é rodado e transladado em x e y no passo 1 de tal forma que $\overline{O_i P_d}$ é paralelo a $\overline{P_{oz} P}$ para cada ponto considerado, então a translação em z não influencia a direcção de $\overline{O_i P_d}$, uma vez que, de acordo com as Eqs. 4a e 4b, z influencia X_u e Y_u do mesmo modo, de tal forma que $\overline{O_i P_u} // \overline{O_i P_d}$.

Observação IV: A restrição $\overline{O_i P_d}$ é paralelo a $\overline{P_{oz} P}$ para cada ponto considerado, independentemente do coeficiente de distorção radial k_1 , da distância focal efectiva f e da componente z do vector 3D de translação T , é suficiente para a determinação da matriz de rotação e das

componentes do vector de translação t_x e t_y , integrantes da transformação de coordenadas do sistema mundo para o sistema câmara.

Entre as quatro observações, as três primeiras são facilmente verificadas, enquanto a última requer alguma intuição geométrica e imaginação para a sua verificação. De facto, como se verá mais adiante, a restrição do alinhamento radial não é só suficiente para a determinação única dos parâmetros extrínsecos, excepto t_z , e do parâmetro intrínseco s_x , mas também o seu cálculo envolve apenas a solução de equações lineares com cinco incógnitas, o que permite que o cálculo possa ser efectuado com elevada velocidade e de forma automática, pois valores iniciais, que são normalmente requeridos para optimizações não lineares, não são necessários.

1.2.4 - Calibração utilizando um conjunto de pontos não coplanares

A Fig. 6 ilustra um exemplo do *setup* para a calibração de uma câmara utilizando um conjunto de pontos não coplanares. No caso apresentado, o plano de calibração tem desenhados vários quadrados cujos vértices são os pontos de calibração. Para se obter um conjunto de pontos de calibração não coplanares, obtém-se, para duas posições distintas do plano de calibração relativamente à câmara, o respectivo subconjunto de pontos de calibração.

Neste método, a determinação dos parâmetros intrínsecos e extrínsecos da câmara em questão está dividida em dois estágios: no primeiro, são determinados os três ângulos da rotação 3D, as componentes segundo x e y da translação 3D e o factor de incerteza horizontal s_x ; no segundo, são determinados a componente segundo z da translação 3D, o factor de distorção radial da lente k_l , e a distância focal efectiva f .

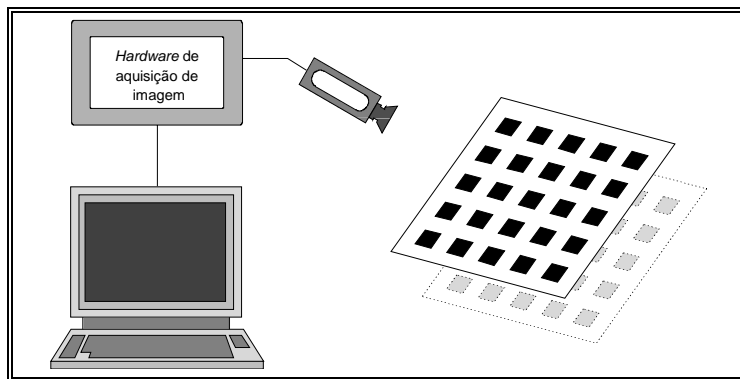


Fig. 6 - Diagrama esquemático do *setup* experimental para calibração de uma câmara utilizando um conjunto de pontos não coplanares.

1) **Estágio 1** - Cálculo da rotação 3D, das translações segundo os eixos x e y , e do factor de incerteza horizontal s_x .

a) Para cada ponto de calibração, calcular as coordenadas (X_d, Y_d) da imagem distorcida.

Procedimento:

- 1) Detectar a localização de cada ponto de calibração i na memória *frame*, designar as respectivas coordenadas por (X_f, Y_f) .
- 2) Obter d'_x e d'_y utilizando os dados fornecidos pelo fabricante da câmara e a Eq. 6c com $s_x = 1$.
- 3) Fazer (C_x, C_y) igual às coordenadas do centro da memória *frame*.
- 4) Calcular (X_{d_i}, Y_{d_i}) utilizando as Eqs. 6a e 6b: $X_{d_i} = d'_x(X_f + C_x)$, $Y_{d_i} = d'_y(Y_f + C_y)$, com $i = 1, \dots, N$, sendo N o número total de pontos de calibração considerados.

b) Calcular as sete incógnitas $t_y^{-1}s_x r_1, t_y^{-1}s_x r_2, t_y^{-1}s_x r_3, t_y^{-1}s_x t_x, t_y^{-1}r_4, t_y^{-1}r_5$ e $t_y^{-1}r_6$.

Procedimento: Para cada ponto de calibração i , com $(x_{w_i}, y_{w_i}, z_{w_i})$ e (X_{d_i}, Y_{d_i}) representando as coordenadas 3D no sistema mundo e as coordenadas imagem distorcida calculadas anteriormente, resolver o seguinte sistema de equações lineares com $t_y^{-1}s_x r_1, t_y^{-1}s_x r_2, t_y^{-1}s_x r_3, t_y^{-1}s_x t_x, t_y^{-1}r_4, t_y^{-1}r_5$ e $t_y^{-1}r_6$, como incógnitas:

$$\begin{bmatrix} Y_{d_i} x_{w_i} & Y_{d_i} y_{w_i} & Y_{d_i} z_{w_i} & Y_{d_i} & -X_{d_i} x_{w_i} & -X_{d_i} y_{w_i} & -X_{d_i} z_{w_i} \end{bmatrix} \begin{bmatrix} t_y^{-1}s_x r_1 \\ t_y^{-1}s_x r_2 \\ t_y^{-1}s_x r_3 \\ t_y^{-1}s_x t_x \\ t_y^{-1}r_4 \\ t_y^{-1}r_5 \\ t_y^{-1}r_6 \end{bmatrix} = X_{d_i}$$

Com N (o número de pontos de calibração) muito maior do que sete, um sistema de equações linear sobredeterminado pode ser resolvido em ordem às incógnitas $t_y^{-1}s_x r_1, t_y^{-1}s_x r_2, t_y^{-1}s_x r_3, t_y^{-1}s_x t_x, t_y^{-1}r_4, t_y^{-1}r_5$ e $t_y^{-1}r_6$, utilizando, por exemplo, o método de regressão dos mínimos quadrados².

c) Cálculo de $r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8, r_9, t_x$ e t_y , a partir de $t_y^{-1}s_x r_1, t_y^{-1}s_x r_2, t_y^{-1}s_x r_3, t_y^{-1}s_x t_x, t_y^{-1}r_4, t_y^{-1}r_5$ e $t_y^{-1}r_6$.

1) Calcular $|t_y|$ a partir de $t_y^{-1}s_x r_1, t_y^{-1}s_x r_2, t_y^{-1}s_x r_3, t_y^{-1}s_x t_x, t_y^{-1}r_4, t_y^{-1}r_5$ e $t_y^{-1}r_6$.

Procedimento: Considerando o conjunto a_i com $i = 1, \dots, 7$, definido como $a_1 = t_y^{-1}s_x r_1, a_2 = t_y^{-1}s_x r_2, a_3 = t_y^{-1}s_x r_3, a_4 = t_y^{-1}s_x t_x, a_5 = t_y^{-1}r_4, a_6 = t_y^{-1}r_5$ e $a_7 = t_y^{-1}r_6$, $|t_y|$ é definido como $|t_y| = (a_5^2 + a_6^2 + a_7^2)^{1/2}$.

2) Determinação do sinal de t_y .

Procedimento:

- i) Escolher um ponto de calibração i cuja imagem na memória *frame* (X_{d_i}, Y_{d_i}) seja afastada da imagem do centro óptico na mesma memória; considerar as suas coordenadas no sistema mundo $(x_{w_i}, y_{w_i}, z_{w_i})$.
- ii) Fazer o sinal de t_y igual a $+1$.
- iii) Calcular $r_1 = (t_y^{-1}s_x r_1)t_y, r_2 = (t_y^{-1}s_x r_2)t_y, r_3 = (t_y^{-1}s_x r_3)t_y, r_4 = (t_y^{-1}r_4)t_y, r_5 = (t_y^{-1}r_5)t_y, r_6 = (t_y^{-1}r_6)t_y, t_x = (t_y^{-1}s_x t_x)t_y, x_i = r_1 x_{w_i} + r_2 y_{w_i} + r_3 z_{w_i} + t_x$ e $y_i = r_4 x_{w_i} + r_5 y_{w_i} + r_6 z_{w_i} + t_y$.
- iv) Se x_i e X_{d_i} têm o mesmo sinal assim como y_i e Y_{d_i} , então o sinal de t_y é igual a $+1$, caso contrário é igual a -1 .

3) Determinação do factor de incerteza horizontal s_x .

Procedimento: Definir s_x como $s_x = (a_1^2 + a_2^2 + a_3^2)^{1/2} |t_y|$.

² Ver, por exemplo [Chapra, 1988].

4) Cálculo da matriz de rotação 3D R e de t_x .

Procedimento:

i) Calcular $r_1, r_2, r_3, r_4, r_5, r_6$, e t_x , utilizando as equações $r_1 = a_1 t_y / s_x, r_2 = a_2 t_y / s_x, r_3 = a_3 t_y / s_x, r_4 = a_5 t_y, r_5 = a_6 t_y, r_6 = a_7 t_y$ e $t_x = a_4 t_y / s_x$.

ii) Calcular r_7, r_8 e r_9 , utilizando a propriedade ortonormal da matriz de rotação 3D R e a regra da mão direita: $r_7 = r_2 r_6 - r_3 r_5, r_8 = r_3 r_5 - r_1 r_6$ e $r_9 = r_1 r_5 - r_2 r_4$.

II) **Estágio 2** - Calcular a distância focal efectiva f , o coeficiente de distorção radial k_1 , e a translação segundo z , t_z .

a) Calcular uma aproximação para f e para t_z , ignorando a distorção da lente.

Procedimento: Para cada ponto de calibração i , estabelecer a seguinte equação linear com f e t_z como incógnitas:

$$\begin{bmatrix} y_i - Y_{d_i} \\ t_z \end{bmatrix} = w_i Y_{d_i} \text{ onde } y_i = r_4 x_{w_i} + r_5 y_{w_i} + r_6 z_{w_i} + t_y \text{ e } w_i = r_7 x_{w_i} + r_8 y_{w_i} + r_9 z_{w_i}.$$

Com vários pontos de calibração, obtém-se um sistema de equações lineares sobredeterminado que pode ser resolvido para a determinação das incógnitas f e t_z , por exemplo, utilizando o método de regressão dos mínimos quadrados.

b) Cálculo da solução exacta para f, t_z e k_1 .

Procedimento: Resolver a Eq. 8b, com f, t_z e k_1 como incógnitas, por um método de optimização *standard*, como, por exemplo, o método Levenberg-Marquardt³. Utilizar como solução inicial a aproximação para f e para t_z encontrada anteriormente, e zero para k_1 .

1.2.5 - Algumas observações

Este método tem como grandes vantagens a inclusão no modelamento utilizado para a câmara da distorção radial da lente e de um factor de incerteza horizontal relacionado com má sincronização temporal entre o *hardware* de aquisição de imagem e o de varrimento da câmara ou de imprecisões no varrimento TV, e a utilização de equações lineares e não lineares de pequena dimensão. Contudo, apresenta como desvantagens o facto de considerar que as coordenadas da imagem do centro óptico na memória *frame* do computador coincidem com as coordenadas do centro desta, e a necessidade de utilizar um conjunto de pontos de calibração não coplanares, o que pode não ser a solução mais interessante e desejável em certas aplicações.

1.3 - Método de calibração considerando a distorção radial da lente (dois termos) e utilizando pontos de calibração não coplanares

Em [Wu, 1988] é proposto um método de calibração de câmaras em tudo similar ao proposto em [Tsai, 1987], a não ser no facto da distorção radial da lente ser modelada por dois termos. Assim, as Eqs. 7a e 7b, que relacionam as coordenadas 2D imagem real (X_d, Y_d) com as coordenadas 3D do ponto P no sistema câmara (x, y, z) , apresentadas anteriormente, são substituídas neste método por:

$$X_d(1 + k_1 r^2 + k_2 r^4) = f \frac{x}{z} \text{ e } Y_d(1 + k_1 r^2 + k_2 r^4) = f \frac{y}{z}, \text{ respectivamente.}$$

Saliente-se, novamente, que experiências reportadas demonstram que, em aplicações industriais de visão por computador, apenas é necessário considerar a distorção radial e um único termo da

³ Ver, por exemplo [Press, 1992].

respectiva série. Qualquer modelamento mais elaborado para a distorção da lente não só será desnecessário como provocará instabilidades numéricas [Tsai, 1987].

1.4 - Método de calibração considerando a distorção radial da lente (um único termo) e utilizando pontos de calibração coplanares

Neste ponto é apresentado um outro método de calibração de câmaras proposto em [Tsai, 1987]. Este método é similar ao apresentado no ponto 1.2, no entanto, utiliza pontos coplanares em vez de não coplanares. Deste modo a abordagem seguida, o modelo utilizado para a câmara e a motivação da técnica de calibração utilizada, são comuns ao referido método. Como este método de calibração é a base do que foi implementado e utilizado, assim como de muitos métodos actualmente existentes que utilizam pontos de calibração coplanares, é apresentado com algum detalhe.

1.4.1 - Calibração de uma câmara utilizando um conjunto de pontos coplanares

A Fig. 7 ilustra um exemplo de *setup* para a calibração de uma câmara utilizando um conjunto de pontos coplanares. No caso apresentado, o plano de calibração tem desenhados vários quadrados cujos vértices são os pontos de calibração. Como os pontos de calibração estão sobre um mesmo plano, o sistema de coordenadas (x_w, y_w, z_w) pode ser escolhido de forma que a coordenada z_w de todos os pontos seja igual a zero e a sua origem não seja próxima do centro de visão ou do eixo y do sistema de coordenadas da câmara. Como o sistema de coordenadas (x_w, y_w, z_w) é definido pelo utilizador e a sua origem é arbitrária, não constitui problema impor essas condições. O propósito desta última condição é garantir que t_y (a translação segundo o eixo y) não seja exactamente igual a zero, de tal modo que a descrição do procedimento de cálculo seja mais compacta e simples. No caso de t_y ser igual a zero, é muito fácil modificar o algoritmo que irá ser apresentado; contudo, tal não deve ser necessário pois pode ser evitado.

Neste método de calibração, a determinação dos parâmetros intrínsecos e extrínsecos da câmara em questão está também dividida em dois estágios; assim, no primeiro, são determinados os três ângulos da rotação 3D e as componentes segundo x e y da translação 3D; no segundo, são determinados a componente segundo z da translação 3D, o factor de distorção radial da lente k_1 , e a distância focal efectiva f .

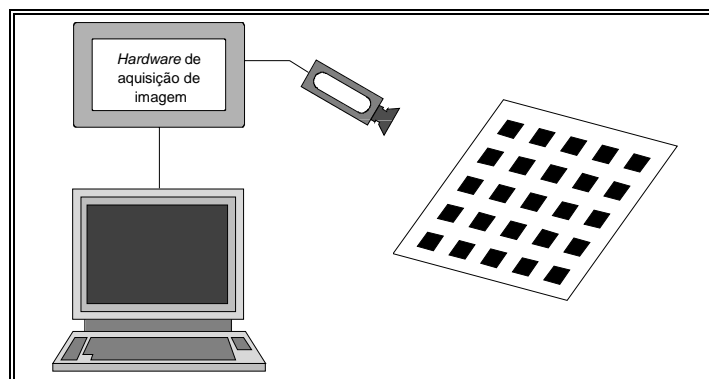


Fig. 7 - Diagrama esquemático do *setup* experimental para calibração de uma câmara utilizando um conjunto de pontos coplanares.

I) **Estágio 1** - Cálculo da rotação 3D e das translações segundo os eixos x e y .

a) Determinar o factor de incerteza horizontal s_x .

Procedimento:

Neste método de calibração, não é possível determinar o factor de incerteza horizontal s_x pelo processo de calibração; assim, define-se s_x como $s_x = N_{cx}/N_{fx}$, onde N_{cx} é o número de elementos sensores na direcção x e N_{fx} é o número de elementos imagem numa linha discretizada pelo computador.

b) Calcular as coordenadas da imagem distorcida (X_d, Y_d) .

Procedimento:

- 1) Detectar a localização de cada ponto de calibração i na memória *frame*, designar as respectivas coordenadas por (X_{f_i}, Y_{f_i}) .
- 2) Obter d'_x e d'_y , utilizando os dados respectivos fornecidos pelo fabricante da câmara e a Eq. 6c com s_x definido anteriormente.
- 3) Fazer (C_x, C_y) igual às coordenadas do centro da memória *frame*.
- 4) Calcular (X_{d_i}, Y_{d_i}) utilizando as Eqs. 6a e 6b: $X_{d_i} = d'_x(X_{f_i} + C_x)$, $Y_{d_i} = d'_y(Y_{f_i} + C_y)$, com $i = 1, \dots, N$, sendo N o número total de pontos de calibração utilizados.

c) Calcular as cinco incógnitas $t_y^{-1}r_1, t_y^{-1}r_2, t_y^{-1}t_x, t_y^{-1}r_4$ e $t_y^{-1}r_5$.

Procedimento: Para cada ponto de calibração i com $(x_{w_i}, y_{w_i}, z_{w_i})$ e (X_{d_i}, Y_{d_i}) representando as coordenadas 3D no sistema mundo e as coordenadas imagem distorcida calculadas anteriormente, resolver o seguinte sistema de equações lineares com $t_y^{-1}r_1, t_y^{-1}r_2, t_y^{-1}t_x, t_y^{-1}r_4$ e $t_y^{-1}r_5$, como incógnitas:

$$\begin{bmatrix} Y_{d_i}x_{w_i} & Y_{d_i}y_{w_i} & Y_{d_i} & -X_{d_i}x_{w_i} & -X_{d_i}y_{w_i} \end{bmatrix} \begin{bmatrix} t_y^{-1}r_1 \\ t_y^{-1}r_2 \\ t_y^{-1}t_x \\ t_y^{-1}r_4 \\ t_y^{-1}r_5 \end{bmatrix} = X_{d_i} \quad (\text{Eq. 10})$$

Com N , o número de pontos de calibração, muito maior do que cinco, um sistema de equações linear sobredeterminado pode ser resolvido em ordem às cinco incógnitas $t_y^{-1}r_1, t_y^{-1}r_2, t_y^{-1}t_x, t_y^{-1}r_4$ e $t_y^{-1}r_5$ utilizando, por exemplo, o método de regressão dos mínimos quadrados.

Justificação e verificação da existência de uma solução única: A Eq. 10 deriva simplesmente da restrição do alinhamento radial $\overline{O_i P_d} // \overline{P_{oz} P}$, representado na Fig. 5 e descrito no ponto 1.2. Obviamente, a Eq. 10 tem uma solução única, se e só se, a matriz dos coeficientes tiver todas as colunas linearmente independentes. Facilmente se verifica que, de facto, todas as colunas da matriz dos coeficientes são linearmente independentes para N muito maior do que cinco.

d) Cálculo de $r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8, r_9, t_x$ e t_y , a partir de $t_y^{-1}r_1, t_y^{-1}r_2, t_y^{-1}t_x, t_y^{-1}r_4$ e $t_y^{-1}r_5$.

1) Calcular $|t_y|$ a partir de $t_y^{-1}r_1, t_y^{-1}r_2, t_y^{-1}t_x, t_y^{-1}r_4$ e $t_y^{-1}r_5$.

Procedimento: Seja C uma submatriz 2×2 da matriz de rotação 3D R ; por exemplo, C é definida como:

$$C = \begin{bmatrix} r'_1 & r'_2 \\ r'_4 & r'_5 \end{bmatrix} = \begin{bmatrix} r_1/t_y & r_2/t_y \\ r_4/t_y & r_5/t_y \end{bmatrix}. \quad (Eq. 11)$$

Se o determinante de C é diferente de zero, então:

$$t_y^2 = \frac{S_r - [S_r^2 - 4(r'_1 r'_5 - r'_4 r'_2)^2]^{1/2}}{2(r'_1 r'_5 - r'_4 r'_2)^2}, \quad (Eq. 12)$$

onde $S_r = r'_1{}^2 + r'_2{}^2 + r'_4{}^2 + r'_5{}^2$, senão:

$$t_y^2 = (r'_i{}^2 + r'_j{}^2)^{-1}, \quad (Eq. 13)$$

onde r'_i e r'_j , são os elementos da linha ou da coluna da matriz C que não apresenta nenhum elemento igual a zero. Esta última situação normalmente nunca será verificada.

Justificação e verificação da existência de uma solução única: Todos os elementos na Eq. 11 para a matriz C são únicos; além do mais, C é na verdade a submatriz 2×2 superior esquerda da matriz ortonormal R (do primeiro tipo⁴) escalada de $1/t_y$. O próximo lema impõe uma restrição para o escalonamento de uma submatriz 2×2 de uma matriz 3×3 ortonormal, preservando a ortogonalidade.

Lema I: Não existem duas matrizes 3×3 ortonormais cujas submatrizes 2×2 respectivas difiram de um factor de escala diferente de ± 1 . Do mesmo modo, se a submatriz 2×2 de uma matriz 3×3 ortonormal é conhecida, excepto o factor de escala, então o factor de escala é único a menos do sinal.

Note-se que a matriz C (a submatriz 2×2 de R), é determinada em \mathbf{c} e existem duas ou mais soluções para $|t_y|$; então da Eq. 11 existirão dois ou mais factores de escala r_1, r_2, r_4 e r_5 , o que contradiz o **lema I**. Deste modo, o **lema I** claramente estabelece a existência de uma solução única para $|t_y|$.

2) Determinação do sinal de t_y .

Procedimento:

i) Escolher um ponto i no sistema de coordenadas mundo $(x_{w_i}, y_{w_i}, z_{w_i})$ cujas coordenadas imagem na memória *frame* (X_f, Y_f) são afastadas das coordenadas imagem do centro óptico, na mesma memória, (C_x, C_y) .

ii) Fazer o sinal de t_y igual a $+1$.

iii) Calcular $r_1 = (t_y^{-1} r_1) t_y$, $r_2 = (t_y^{-1} r_2) t_y$, $r_4 = (t_y^{-1} r_4) t_y$, $r_5 = (t_y^{-1} r_5) t_y$, $t_x = (t_y^{-1} t_x) t_y$, $x_i = r_1 x_{w_i} + r_2 y_{w_i} + t_x$ e $y_i = r_4 x_{w_i} + r_5 y_{w_i} + t_y$.

iv) Se x_i e X_{d_i} têm o mesmo sinal assim como y_i e Y_{d_i} , então o sinal de t_y é igual a $+1$, caso contrário é igual a -1 .

Justificação e verificação da existência de uma solução única: Apesar da determinação de uma solução única para t_y^2 ou $|t_y|$, t_y pode ainda assumir o sinal $+$ ou $-$. Note-se que calculando $t_y^{-1} r_1, t_y^{-1} r_2, t_y^{-1} t_x, t_y^{-1} r_4$ e $t_y^{-1} r_5$, utilizando a Eq. 10, invertendo o sinal de t_y troca-se o sinal de r_1, r_2, r_4, r_5 e de t_x . Relembrando que a equação

⁴ Por definição, uma matriz ortonormal do primeiro tipo tem determinante $+1$, ao contrário de uma matriz ortonormal do segundo tipo cujo determinante é -1 .

linear utilizada para calcular $t_y^{-1}r_1$, $t_y^{-1}r_2$, $t_y^{-1}t_x$, $t_y^{-1}r_4$ e $t_y^{-1}r_5$ (Eq. 10), foi derivada da restrição do alinhamento radial $\overline{O_i P_d} // \overline{P_{oz} P}$, ou $(X_d, Y_d) // (x, y)$ onde $x = r_1 x_{w_i} + r_2 y_{w_i} + t_x$ e $y = r_4 x_{w_i} + r_5 y_{w_i} + t_y$, a inversão do sinal de t_y implica que (x, y) se torne em $(-x, -y)$, permanecendo paralelo a (X_d, Y_d) , apesar de ter direcção oposta. Contudo, segundo as Eqs. 4a e 4b, não só $(X_d, Y_d) // (X_u, Y_u) // (x, y)$, mas também x_i e X_{d_i} , y_i e Y_{d_i} , têm o mesmo sinal, uma vez que f e z são ambos positivos; a observação da geometria simples ilustrada na Fig. 5 origina a mesma conclusão. Deste modo, apenas um dos sinais para t_y é válido, e pode ser determinado utilizando o procedimento anteriormente descrito.

3) Cálculo da matriz de rotação 3D R , ou seja $r_1, r_2, r_3, r_4, r_5, r_6, r_7, r_8$ e r_9 .

Procedimento:

i) Calcular $r_1 = (t_y^{-1}r_1)t_y$, $r_2 = (t_y^{-1}r_2)t_y$, $r_4 = (t_y^{-1}r_4)t_y$, $r_5 = (t_y^{-1}r_5)t_y$ e $t_x = (t_y^{-1}t_x)t_y$.

ii) Calcular a matriz R pela seguinte equação:

$$R = \begin{bmatrix} r_1 & r_2 & (1 - r_1^2 - r_2^2)^{1/2} \\ r_4 & r_5 & s(1 - r_4^2 - r_5^2)^{1/2} \\ r_7 & r_8 & r_9 \end{bmatrix}, \quad (\text{Eq. 14a})$$

onde $s = -\text{sinal}(r_1 r_4 + r_2 r_5)$. A função $\text{sinal}()$ significa o sinal do seu argumento. Os elementos da matriz de rotação 3D R : r_7, r_8 e r_9 , são determinados usando o produto vectorial das duas primeiras linhas, utilizando a propriedade ortonormal da matriz R .

iii) Calcular uma aproximação para a distância focal efectiva f utilizando a Eq. 15; se esta aproximação é menor do que zero, então:

$$R = \begin{bmatrix} r_1 & r_2 & -(1 - r_1^2 - r_2^2)^{1/2} \\ r_4 & r_5 & -s(1 - r_4^2 - r_5^2)^{1/2} \\ -r_7 & -r_8 & r_9 \end{bmatrix}. \quad (\text{Eq. 14b})$$

Justificação e verificação da existência de uma solução única: Como as soluções únicas para $t_y^{-1}r_1$, $t_y^{-1}r_2$, $t_y^{-1}r_4$ e $t_y^{-1}r_5$, são determinados em c) e a solução única para t_y é determinada em d), obviamente pode-se determinar soluções únicas para r_1, r_2, r_4 e r_5 . Note-se que r_1 e r_2 são os elementos da submatriz 2×2 superior da matriz R . O problema que se debate é como proceder à determinação das soluções únicas para os restantes elementos da matriz R . Uma solução para este problema é fornecida pelo lema seguinte:

Lema II: Dada uma submatriz 2×2 de uma matriz 3×3 ortonormal do primeiro tipo, existem exactamente duas soluções para a matriz ortonormal. Estas soluções são dadas pelas Eqs. 14a e 14b.

A explicação de que o procedimento descrito anteriormente para a escolha entre as Eqs. 14a e 14b corresponde à solução correcta e única, é a seguinte:

Nas Eqs. 14a e 14b, apenas as primeiras duas linhas são dadas explicitamente em termos das quantidades previamente determinadas r_1, r_2, r_4 e r_5 . Com base na propriedade ortonormal da matriz R e na regra da mão direita (por exemplo, o determinante de R é $+1$ e não -1) r_7, r_8 e r_9 , são facilmente calculados unicamente a partir das duas primeiras linhas. Apenas uma solução dada pelas Eqs. 14a e 14b é válida. Esta afirmação é baseada no facto de que, invertendo o sinal de z para todos os

pontos no sistema de coordenadas 3D câmara, por exemplo $(x, y, z) \rightarrow (x, y, -z)$, todos os pontos se mantêm coplanares. Contudo, como t_z ainda não é determinado no estágio 1, ainda não se pode determinar a coordenada z (pois $z = r_7x_w + r_8y_w + t_z$). Das Eqs. 4a e 4b, observa-se que invertendo o sinal de z também o sinal de f é invertido. Deste modo, a maneira mais simples de seleccionar a solução correcta entre as soluções dadas pelas Eqs. 14a e 14b é utilizando a equação linear apresentada no estágio 2 para calcular uma aproximação para f e t_z , Eq. 15. A solução errada origina uma aproximação para f negativa, enquanto a solução correcta para R origina uma aproximação para f positiva. Note-se que não é necessário entrar com a distorção para a escolha da solução correcta para a matriz R , pois apenas o sinal de f é necessário.

II) Estágio 2 - Calcular a distância efectiva f , coeficiente de distorção radial k_l e a translação segundo o eixo z , t_z .

a) Calcular uma aproximação para f e para t_z ignorando a distorção da lente.

Procedimento: Para cada ponto de calibração i , estabelecer a seguinte equação linear com f e t_z como incógnitas:

$$\begin{bmatrix} y_i - Y_{d_i} \\ f \\ t_z \end{bmatrix} = w_i Y_{d_i}, \text{ onde } y_i = r_4x_{w_i} + r_5y_{w_i} + t_y \text{ e } w_i = r_7x_{w_i} + r_8y_{w_i}. \quad (\text{Eq. 15})$$

Com vários pontos de calibração, obtém-se um sistema de equações linear sobredeterminado que pode ser resolvido para a determinação das incógnitas f e t_z . O plano de calibração não pode ser exactamente paralelo ao plano imagem pois, de outro modo, a Eq. 15 tornar-se-ia linearmente dependente.

Justificação: A Eq. 15 é derivada da Eq. 8b fazendo k_l igual a zero. Como R , t_x e t_y já foram determinados e y e w estão definidos, então a Eq. 15 é uma equação linear com f e t_z como incógnitas. Note-se que apesar da Eq. 8a poder originar uma equação similar, é redundante. Utilizando a Eq. 15 e um conjunto de pontos de calibração coplanares, um sistema de equações linear sobredeterminado pode ser estabelecido para a determinação de uma aproximação para f e para t_z . Esta determinação é facilmente obtida, por exemplo utilizando o método dos mínimos quadrados.

b) Cálculo da solução exacta para f , t_z e k_l .

Procedimento: Resolver a Eq. 8b com f , t_z e k_l como incógnitas, por um método de optimização *standard* como, por exemplo, o método Levenberg-Marquardt. Utilizar como solução inicial a aproximação para f e para t_z encontrada pela resolução do sistema linear de equações anterior (Eq. 15) e zero para k_l .

Justificação e verificação da existência de uma solução única: Como R , t_x e t_y foram determinados previamente, a Eq. 8b torna-se numa equação não linear com f , t_z e k_l como incógnitas. Normalmente, caso se utilize o método Levenberg-Marquardt, um número reduzido de iterações é suficiente.

1.4.2 - Algumas observações

Este método tem como grandes vantagens: a inclusão no modelamento utilizado para a câmara da distorção radial da lente e de um factor de incerteza horizontal (como no método apresentado no ponto 1.2); a utilização de equações lineares e não lineares de pequena dimensão e de pontos coplanares. Contudo, apresenta como desvantagens: o facto de considerar que as coordenadas da imagem do centro óptico na memória *frame* coincidem com as coordenadas do centro desta; a

impossibilidade do factor de incerteza horizontal s_x ser determinado pelo processo de calibração (a atribuição que é realizada a este parâmetro é uma aproximação de fraca qualidade [Lenz, 1988]); a necessidade do plano de calibração não ser paralelo ao plano câmara - o plano de calibração deve, pelo menos, apresentar um ângulo de 30° em relação ao eixo óptico [Batista, 1993] - constitui uma dificuldade na utilização deste método, em certas aplicações.

1.5 - Método de calibração considerando a distorção radial da lente (um único termo), um referencial de coordenadas na memória *frame* não necessariamente paralelo ao referencial imagem e utilizando pontos de calibração coplanares

Neste ponto, é apresentado um método de calibração de câmaras proposto em [Batista, 1993]. Este método é baseado no apresentado no ponto **1.4**; no entanto, neste método já é possível determinar por calibração o factor de incerteza horizontal s_x , assim como as coordenadas na memória *frame* do centro óptico. É também considerado que o referencial na memória *frame* pode não ser paralelo ao referencial imagem, assim como os seus eixos podem não ser perpendiculares entre si.

1.5.1 - Introdução

Neste método, a determinação dos parâmetros de calibração pode ser dividida em quatro passos: no primeiro, é realizada a determinação da matriz de rotação 3D; no segundo, é realizada a determinação das translações segundo as direcções x e y ; no terceiro, é realizada a determinação do coeficiente de distorção radial, da translação segundo a direcção z e da distância focal efectiva; no quarto e último passo, é realizada a determinação dos factores de escala de imagem, coordenadas na memória *frame* do centro óptico, e os desvios ortogonais do referencial na mesma memória.

A maior diferença entre este método de calibração e o apresentado no ponto **1.4**, prende-se com a determinação da matriz de rotação 3D. O princípio que é utilizado é baseado no facto de que existe informação suficiente, na projecção perspectiva 2D de um rectângulo de dimensões desconhecidas no espaço 3D, para a determinação dos ângulos de orientação da câmara em questão. Contudo, para a determinação da matriz de rotação 3D, é necessário o conhecimento das coordenadas na memória *frame* dos vértices do respectivo rectângulo; assim, é indispensável o prévio conhecimento dos factores de escala de imagem e das coordenadas, na memória *frame*, do centro óptico. Este problema é resolvido considerando que, num estágio mais avançado, serão determinados com maior precisão os factores de escala de imagem e as coordenadas na memória *frame* do centro óptico. Deste modo, no primeiro estágio é considerado para a câmara um modelo *pinhole* sem distorção (tal como acontece nos métodos apresentados nos pontos **1.2** e **1.4**), e que os referenciais no plano câmara e na memória *frame* são perfeitamente ortogonais.

Com a matriz de rotação 3D já determinada, é então possível determinar o vector de translação 3D e alguns parâmetros intrínsecos, utilizando a relação que define o mesmo cociente entre as coordenadas imagem ideal (X_u, Y_u) e as coordenadas no referencial câmara (x, y) para obter as translações segundo os eixos x e y , seguida da relação utilizada nos métodos apresentados nos pontos **1.2** e **1.4** para obter a translação segundo o eixo z e o factor de distorção radial k_l . Neste método, o procedimento para a determinação da distância focal efectiva f é diferente do utilizado nos métodos apresentados nos pontos **1.2** e **1.4**, basicamente pelo facto de que, utilizando as mesmas equações, a restrição de que o plano da câmara não fosse paralelo ao plano de calibração mantinha-se. Deste modo, neste método é assumido um valor inicial para a distância focal efectiva, sendo este valor actualizado utilizando a lei para as lentes de Newton e a transformação perspectiva determinada no plano que inclui a linha de projecção e o eixo óptico da lente.

Com o conhecimento destes parâmetros (R, T, f, k_l) , os factores de escala de imagem, as coordenadas do centro óptico na memória *frame*, e os desvios ortogonais podem ser obtidos utilizando a transformação perspectiva e as transformações entre as coordenadas imagem distorcida e as coordenadas na memória *frame*.

1.5.2 - Modelo utilizado para a câmara

Neste método, a transformação global das coordenadas do sistema 3D mundo para as coordenadas na memória *frame*, Fig. 8, é decomposta nos passos seguintes:

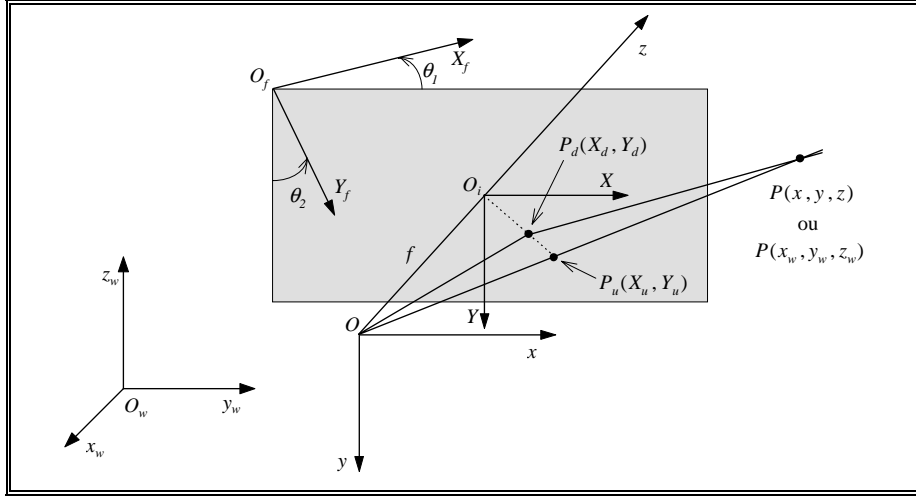


Fig. 8 - Ilustração do modelo utilizado para a câmara.

1) Transformação de corpo rígido do sistema de coordenadas mundo (x_w, y_w, z_w) para o sistema 3D câmara (x, y, z) , Eq. 3.

2) Transformação das coordenadas 3D câmara (x, y, z) para as coordenadas imagem ideal (X_u, Y_u) (isto é, sem distorção), utilizando projecção perspectiva com geometria *pinhole* para a câmara, Eqs. 4a e 4b.

3) Transformação das coordenadas ideais para as coordenadas reais utilizando apenas um coeficiente de distorção radial da lente:

$$X_d + D_x = X_u, Y_d + D_y = Y_u, \quad (\text{Eqs. 16a e 16b})$$

onde (X_d, Y_d) são as coordenadas imagem real ou distorcida no plano imagem, $D_x = X_d k_1 r^2$ e $D_y = Y_d k_1 r^2$, com $r^2 = X_u^2 + Y_u^2$ e k_1 o coeficiente de distorção radial.

4) Transformação das coordenadas reais (X_d, Y_d) para as coordenadas na memória *frame* (X_f, Y_f) :

$$X_f = a_x X_d + b_x Y_d + C_x, Y_f = a_y X_d + b_y Y_d + C_y, \quad (\text{Eqs. 17a e 17b})$$

onde (C_x, C_y) são as coordenadas discretas do centro óptico na memória *frame*, e os restantes parâmetros estão relacionados directamente com os factores de escala de imagem (k_x, k_y) e os desvios ortogonais do sistema de coordenadas na memória *frame* (θ_1, θ_2) .

Assumindo a geometria entre os eixos do referencial no plano imagem e os eixos do referencial na memória *frame* apresentada na Fig. 8, a transformação entre estes dois sistemas de coordenadas pode ser representada por:

$$X_f = \frac{k_x \cos(\theta_1)}{\cos(\theta_1 - \theta_2)} X_d + \frac{k_x \sin(\theta_1)}{\cos(\theta_1 - \theta_2)} Y_d + C_x, Y_f = \frac{-k_y \sin(\theta_2)}{\cos(\theta_1 - \theta_2)} X_d + \frac{k_y \cos(\theta_2)}{\cos(\theta_1 - \theta_2)} Y_d + C_y, \quad (\text{Eqs. 18a e 18b})$$

onde $\theta_1 = \arctg \frac{b_x}{a_x}$, $\theta_2 = -\arctg \frac{a_y}{b_y}$, $k_x = \sqrt{(a_x^2 + b_x^2) \cos^2(\theta_1 - \theta_2)}$ e $k_y = \sqrt{(a_y^2 + b_y^2) \cos^2(\theta_1 - \theta_2)}$.

Deste modo os parâmetros a determinar pelo processo de calibração são a matriz de rotação 3D R , o vector de translação 3D T , a distância focal efectiva f , o coeficiente de distorção radial da lente k_1 , as coordenadas do centro óptico na memória *frame*, C_x e C_y , e os parâmetros a_x, b_x, a_y e b_y .

1.5.3 - Determinação da matriz de rotação

Para a determinação da matriz de rotação 3D, assume-se neste método a existência de dois sistemas de coordenadas tridimensionais, Fig. 9, ambos com a respectiva origem sobre o centro óptico da lente da câmara e definidos do seguinte modo:

- 1) o sistema de coordenadas mundo é definido como ortogonal e directo com a sua origem sobre o centro óptico da lente da câmara;
- 2) o sistema de coordenadas câmara é também definido como ortogonal e directo com a sua origem sobre o centro óptico da lente da câmara, e com o eixo z coincidente com o eixo óptico da mesma.

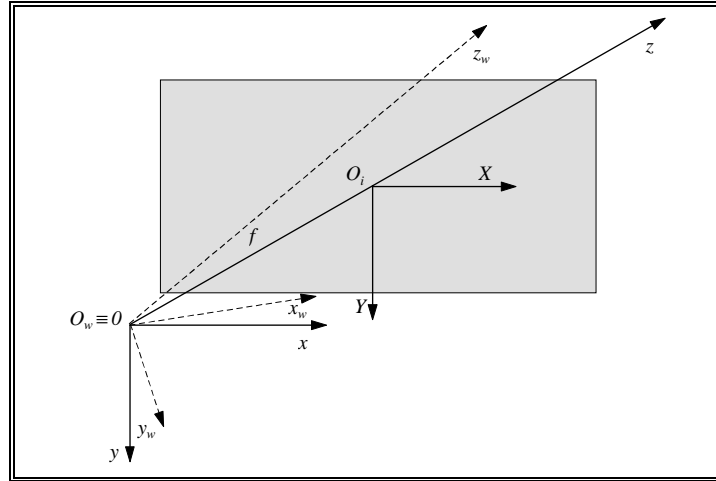


Fig. 9 - Sistemas de coordenadas 3D câmara e mundo, utilizados na determinação da matriz de rotação.

O plano imagem está localizado à frente do centro óptico de uma distância f , e é ortogonal ao eixo óptico da lente. Os eixos X e Y do sistema de coordenadas imagem são paralelos aos eixos x e y do sistema câmara, respectivamente.

A matriz de rotação define a transformação entre os sistemas de coordenadas mundo e câmara através da relação $P_{\text{mundo}} = R^{-1} P_{\text{câmara}}$, onde R^{-1} é uma matriz de rotação ortogonal resultante da inversão da matriz de rotação R definida na Eq. 9.

Considerando (x_w, y_w, z_w) as coordenadas no referencial mundo de um ponto cujas coordenadas no referencial câmara são (X_u, Y_u, f) , elas são obtidas do seguinte modo:

$$\begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} = R^{-1} \begin{bmatrix} X_u \\ Y_u \\ f \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & r_3 \\ r_4 & r_5 & r_6 \\ r_7 & r_8 & r_9 \end{bmatrix}^T \begin{bmatrix} X_u \\ Y_u \\ f \end{bmatrix}. \quad (\text{Eq. 19})$$

Como o centro da lente coincide com a origem de ambos os sistemas de coordenadas, uma linha passante pelo centro da lente e pelo ponto de coordenadas (x_w, y_w, z_w) no sistema mundo consiste em todos os múltiplos de (x_w, y_w, z_w) . Do mesmo modo, a linha cuja projecção perspectiva no plano imagem é (X_u, Y_u) consiste em todos os pontos:

$$\begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} \parallel \begin{bmatrix} X_u \\ Y_u \\ f \end{bmatrix} = \lambda \begin{bmatrix} X_u \\ Y_u \\ f \end{bmatrix} \text{ para alguma constante } \lambda. \quad (\text{Eq. 20})$$

Com o intuito de simplificar a relação anterior, é considerado que o ângulo ψ é nulo, de tal modo que é definida, sem perda de generalidade, a seguinte relação:

$$\begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} = \lambda \begin{bmatrix} \cos\theta X'_u - f \sin\theta \\ \sin\theta \sin\phi X'_u + \cos\phi Y'_u + f \cos\theta \sin\phi \\ \sin\theta \cos\phi X'_u - \sin\phi Y'_u + f \cos\theta \cos\phi \end{bmatrix}. \quad (Eq. 21)$$

Esta relação é válida desde que, se (X_u, Y_u) forem as coordenadas de um dado ponto sobre o plano imagem rodado de um ângulo ψ diferente de zero, então (X'_u, Y'_u) são as coordenadas do mesmo ponto sobre o plano imagem com o ângulo ψ nulo, e estão relacionados pela seguinte expressão:

$$\begin{bmatrix} X'_u \\ Y'_u \end{bmatrix} = \begin{bmatrix} \cos\psi & \sin\psi \\ -\sin\psi & \cos\psi \end{bmatrix} \begin{bmatrix} X_u \\ Y_u \end{bmatrix}. \quad (Eq. 22)$$

A observação que origina a determinação da matriz de rotação 3D, foi inicialmente proposta por Haralick [Batista, 1993] e baseia-se no facto de existir informação suficiente na projecção perspectiva 2D dos vértices de um rectângulo para a determinação dos ângulos da orientação relativa da câmara. Baseado nesta observação, considere-se a existência de um rectângulo de dimensões desconhecidas localizado perpendicularmente ao eixo x_w do sistema de coordenadas mundo, Fig. 10. Os vértices do rectângulo considerado são:

$$P_1 = \begin{bmatrix} x_l \\ y_l \\ z_l \end{bmatrix}, P_2 = \begin{bmatrix} x_l \\ y_l + W \\ z_l \end{bmatrix}, P_3 = \begin{bmatrix} x_l \\ y_l \\ z_l + L \end{bmatrix} \text{ e } P_4 = \begin{bmatrix} x_l \\ y_l + W \\ z_l + L \end{bmatrix},$$

onde $x_l > f$ e as correspondentes projecções perspectivas destes pontos no plano imagem são:

$$P_{u_1} = \begin{bmatrix} X_{u_1} \\ Y_{u_1} \end{bmatrix}, P_{u_2} = \begin{bmatrix} X_{u_2} \\ Y_{u_2} \end{bmatrix}, P_{u_3} = \begin{bmatrix} X_{u_3} \\ Y_{u_3} \end{bmatrix} \text{ e } P_{u_4} = \begin{bmatrix} X_{u_4} \\ Y_{u_4} \end{bmatrix}.$$

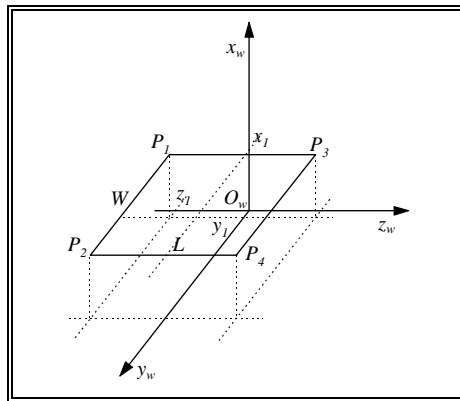


Fig. 10 - Orientação do rectângulo para a determinação da matriz de rotação.

A selecção destes pontos implica uma orientação particular do rectângulo, com o intuito de assegurar o paralelismo entre os lados do rectângulo e os eixos x_w e y_w do sistema de coordenadas mundo. As coordenadas destes pontos P_i estão expressas no sistema câmara. Como as coordenadas destes pontos em *pixels* só podem ser obtidas por intermédio da respectiva imagem, é necessário conhecer previamente alguns parâmetros intrínsecos, tais como os factores de escala de imagem e as coordenadas do centro da imagem na memória *frame*. Este problema é resolvido assumindo valores predefinidos para estes parâmetros, utilizando-se para tal as mesmas relações utilizadas por Tsai nos métodos apresentados nos pontos 1.2 e 1.4. Assim, assume-se que o centro da imagem na memória *frame* coincide com o centro desta, e os factores de escala de imagem são definidos pelas relações:

$$k_x = \frac{s_x N_{fx}}{d_x N_{cx}}, \quad k_y = \frac{I}{d_y}$$

Neste método, é considerado um factor de incerteza horizontal s_x unitário no primeiro estágio para a determinação da matriz de rotação.

Para a determinação da matriz de rotação é assumido um ângulo ψ nulo, que define a projecção perspectiva dos pontos P'_{u_i} através da Eq. 22. Com este procedimento, é assumido que é conhecido um valor para o ângulo ψ . Do mesmo modo, é assumido um valor predefinido conhecido para a distância focal f , que é a distância focal especificada para a lente na primeira iteração, e o valor obtido na anterior iteração nas seguintes.

Utilizando a Eq. 21 para definir as quatro linhas de projecção dos vértices do rectângulo, obtêm-se as seguintes equações matriciais (que são a base desta determinação):

$$P_i = \lambda_i \begin{bmatrix} \cos\theta X'_i - f \sin\theta \\ \sin\theta \sin\phi X'_i + \cos\phi Y'_i + f \cos\theta \sin\phi \\ \sin\theta \cos\phi X'_i - \sin\phi Y'_i + f \cos\theta \cos\phi \end{bmatrix} \text{ com } i = 1, \dots, 4. \quad (\text{Eq. 23})$$

Estas últimas cinco equações são suficientes para determinar os três ângulos de rotação (ψ , θ , ϕ), com a não necessidade do conhecimentos dos $\lambda_{i=1, \dots, 4}$ bem como das coordenadas 3D dos vértices do rectângulo considerado.

Através da análise destas equações matriciais, observa-se que a primeira e a terceira linha da Eq. 23 para P_1 e P_2 são idênticas no que respeita a x_1 e a z_1 . Conjugando estas duas equações obtém-se:

$$f \sin\theta \sin\phi (Y'_{u_2} - Y'_{u_1}) + \cos\theta \sin\phi (X'_{u_2} Y'_{u_1} - X'_{u_1} Y'_{u_2}) + f \cos\phi (X'_{u_1} - X'_{u_2}) = 0. \quad (\text{Eq. 24})$$

De maneira semelhante, a primeira e a terceira linha da Eq. 23 para P_3 e P_4 são idênticas no que respeita a x_1 e a $z_1 + L$; assim, conjugando estas duas equações obtém-se:

$$f \sin\theta \sin\phi (Y'_{u_4} - Y'_{u_3}) + \cos\theta \sin\phi (X'_{u_4} Y'_{u_3} - X'_{u_3} Y'_{u_4}) + f \cos\phi (X'_{u_3} - X'_{u_4}) = 0. \quad (\text{Eq. 25})$$

Multiplicando a Eq. 24 por $(X'_{u_3} - X'_{u_4})$ e a Eq. 25 por $(X'_{u_1} - X'_{u_2})$ e subtraindo, obtém-se:

$$\begin{aligned} & f \sin\theta [(Y'_{u_2} - Y'_{u_1})(X'_{u_3} - X'_{u_4}) - (Y'_{u_4} - Y'_{u_3})(X'_{u_1} - X'_{u_2})] = \\ & = \cos\theta [(X'_{d_4} Y'_{d_3} - X'_{d_3} Y'_{d_4})(X'_{d_1} - X'_{d_2}) - (X'_{d_2} Y'_{d_1} - X'_{d_1} Y'_{d_2})(X'_{d_3} - X'_{d_4})] \end{aligned} \quad (\text{Eq. 26})$$

que, resolvendo em ordem a θ :

$$\theta = \text{tg}^{-1} \frac{[(X'_{u_4} Y'_{u_3} - X'_{u_3} Y'_{u_4})(X'_{u_1} - X'_{u_2}) - (X'_{u_2} Y'_{u_1} - X'_{u_1} Y'_{u_2})(X'_{u_3} - X'_{u_4})]}{f [(Y'_{u_2} - Y'_{u_1})(X'_{u_3} - X'_{u_4}) - (Y'_{u_4} - Y'_{u_3})(X'_{u_1} - X'_{u_2})]}. \quad (\text{Eq. 27})$$

A solução para ϕ é obtida utilizando um procedimento similar ao utilizado para θ ; multiplicando a Eq. 24 por $(Y'_{u_4} - Y'_{u_3})$ e a Eq. 25 por $(Y'_{u_2} - Y'_{u_1})$ e subtraindo obtém-se:

$$\phi = \text{tg}^{-1} \frac{f [(X'_{u_3} - X'_{u_4})(Y'_{u_2} - Y'_{u_1}) - (X'_{u_1} - X'_{u_2})(Y'_{u_4} - Y'_{u_3})]}{\cos\theta [(X'_{u_2} Y'_{u_1} - X'_{u_1} Y'_{u_2})(Y'_{u_4} - Y'_{u_3}) - (X'_{u_4} Y'_{u_3} - X'_{u_3} Y'_{u_4})(Y'_{u_2} - Y'_{u_1})]}. \quad (\text{Eq. 28})$$

Observando as duas últimas equações, conclui-se que, conhecendo o valor do ângulo ψ , é possível determinar os valores dos ângulos θ e ϕ . A equação que define a solução para o ângulo θ não é única, pois é possível obter uma nova equação baseada no facto de a primeira e segunda linha da Eq. 23 para P_1 e P_3 serem idênticas no que respeita a x_1 e a y_1 , e a primeira e segunda linha da Eq. 23 para P_2 e P_4

são idênticas no que respeita a x_l e a $y_l + W$. Esta nova expressão para θ é obtida utilizando o mesmo procedimento descrito anteriormente, resultando a expressão alternativa e independente:

$$\theta = \text{tg}^{-1} \frac{[(X'_{u_2} Y'_{u_4} - X'_{u_4} Y'_{u_2})(X'_{u_1} - X'_{u_3}) - (X'_{u_1} Y'_{u_3} - X'_{u_3} Y'_{u_1})(X'_{u_2} - X'_{u_4})]}{f [(Y'_{u_1} - Y'_{u_3})(X'_{u_2} - X'_{u_4}) - (Y'_{u_2} - Y'_{u_4})(X'_{u_1} - X'_{u_3})]} \quad (\text{Eq. 29})$$

Com estas duas equações para o ângulo θ pode-se estabelecer a seguinte relação

$$\begin{aligned} & \frac{[(X'_{u_4} Y'_{u_3} - X'_{u_3} Y'_{u_4})(X'_{u_1} - X'_{u_2}) - (X'_{u_2} Y'_{u_1} - X'_{u_1} Y'_{u_2})(X'_{u_3} - X'_{u_4})]}{f [(Y'_{u_2} - Y'_{u_1})(X'_{u_3} - X'_{u_4}) - (Y'_{u_4} - Y'_{u_3})(X'_{u_1} - X'_{u_2})]} = \\ & = \frac{[(X'_{u_2} Y'_{u_4} - X'_{u_4} Y'_{u_2})(X'_{u_1} - X'_{u_3}) - (X'_{u_1} Y'_{u_3} - X'_{u_3} Y'_{u_1})(X'_{u_2} - X'_{u_4})]}{f [(Y'_{u_1} - Y'_{u_3})(X'_{u_2} - X'_{u_4}) - (Y'_{u_2} - Y'_{u_4})(X'_{u_1} - X'_{u_3})]} \end{aligned} \quad (\text{Eq. 30})$$

que, com a Eq. 22, define a expressão para a determinação do último ângulo, ψ . De referir que muitos dos termos incluídos na Eq. 30 são independentes do ângulo de rotação ψ . Deste modo, o ângulo de rotação ψ é definido como:

$$\psi = \text{tg}^{-1} \frac{A(X'_{u_1} - X'_{u_2}) - B(X'_{u_3} - X'_{u_4}) - C(X'_{u_1} - X'_{u_3}) + D(Y'_{u_2} - Y'_{u_4})}{-A(Y'_{u_1} - Y'_{u_2}) + B(Y'_{u_3} - Y'_{u_4}) + C(Y'_{u_1} - Y'_{u_2}) - D(Y'_{u_3} - Y'_{u_4})} \quad (\text{Eq. 31})$$

onde:

$$\begin{aligned} A &= \frac{X'_{u_4} Y'_{u_3} - X'_{u_3} Y'_{u_4}}{E}, \quad B = \frac{X'_{u_2} Y'_{u_1} - X'_{u_1} Y'_{u_2}}{E}, \quad C = \frac{X'_{u_2} Y'_{u_4} - X'_{u_4} Y'_{u_2}}{F}, \quad D = \frac{X'_{u_1} Y'_{u_3} - X'_{u_3} Y'_{u_1}}{F}, \\ E &= f [(Y'_{u_2} - Y'_{u_1})(X'_{u_3} - X'_{u_4}) - (Y'_{u_4} - Y'_{u_3})(X'_{u_1} - X'_{u_2})], \quad F = f [(Y'_{u_1} - Y'_{u_3})(X'_{u_2} - X'_{u_4}) - (Y'_{u_2} - Y'_{u_4})(X'_{u_1} - X'_{u_3})]. \end{aligned}$$

Os valores obtidos para o ângulo de rotação ψ , apresentam uma ambiguidade de 180 grau, enquanto os valores para os ângulos de rotação θ e ψ , apresentam uma ambiguidade de 90 grau.

1.5.4 - Determinação das translações segundo os eixos x e y e do factor de incerteza horizontal s_x

Até este ponto, na abordagem seguida, considerou-se que os dois sistemas de coordenadas câmara e mundo possuíam as respectivas origens coincidentes num mesmo ponto, isto é, não existia translação entres os dois sistemas. Contudo, esta consideração não é geral, assim como habitualmente a localização da lente da câmara não é conhecida previamente. No entanto, esta consideração não originou nenhum tipo de dificuldade, pois a determinação da matriz de rotação, não necessita das coordenadas 3D dos vértices do rectângulo utilizado.

Supondo a existência de um vector 3D de translação entre os dois sistemas de coordenadas, este vector de translação T é baseado na determinação de t_x , t_y e s_x , num primeiro estágio, seguido da determinação de t_z e do factor de distorção k_l , num segundo estágio.

Desde que a matriz de rotação esteja já determinada, pontos definidos no sistema de coordenadas mundo podem ser convertidos para pontos no sistema câmara através da relação $P_{câmara} = RP_{mundo}$. A partir da relação entre as coordenadas x e y de um ponto $P(x, y, z)$ definido no sistema de coordenadas câmara e as coordenadas imagem não distorcida X_u e Y_u , obtidas após a transformação perspectiva, obtém-se:

$$\frac{X_u}{Y_u} = \frac{x}{y} \quad (\text{Eq. 32})$$

Até este ponto considerou-se um factor de incerteza horizontal unitário, pois a determinação da matriz de rotação 3D não é afectada consideravelmente por esta consideração. Tal não acontece na

determinação do vector de translação, pois as componentes deste vector segundo as direcções x e z , são afectadas pelo valor do factor de incerteza horizontal.

Como no primeiro estágio é assumida uma geometria *pinhole* perfeita para a câmara sem distorção radial e, com os sistemas de coordenadas imagem e na memória *frame* perfeitamente ortogonais, obtém-se a seguinte equação matricial:

$$\begin{bmatrix} k_x(Y_{f_i}-C_y) & -k_y(X_{f_i}-C_x) & k_x(Y_{f_i}-C_y)X_{rot_i} \\ t_x & t_y & s_x \end{bmatrix} = \begin{bmatrix} k_y(X_{f_i}-C_x)Y_{rot_i} \end{bmatrix} \text{ com } i = 1, \dots, n, \quad (Eq. 33)$$

combinando a Eq. 32 com a equação $P_{câmara} = RP_{mundo}$ (com $R^{-1} = R^T$, pois R é ortogonal) e, com as seguintes equações:

$$X_f = s_x k_x X_u + C_x \text{ e } Y_f = k_y Y_u + C_y, \text{ onde } X_{rot_i} = r_1 x_{w_i} + r_2 y_{w_i} + r_3 z_{w_i} \text{ e } Y_{rot_i} = r_4 x_{w_i} + r_5 y_{w_i} + r_6 z_{w_i}.$$

Resolvendo a Eq. 33 para todos os pontos de calibração utilizados, obtém-se um sistema linear sobredeterminado, que pode ser resolvido, por exemplo, pelo método de regressão dos mínimos quadrados, em ordem às incógnitas t_x , t_y e s_x . Após a determinação do factor de incerteza horizontal, deve-se proceder à actualização do valor deste e utilizá-lo nas equações seguintes do procedimento de calibração.

1.5.5 - Determinação da translação segundo o eixo z , da distância focal f e do coeficiente de distorção radial k_1

O modelo que origina a metodologia utilizada neste método de calibração é baseado no emprego nos métodos apresentados nos pontos 1.2 e 1.4; mas é diferente no que diz respeito à determinação da distância focal. Assim, no modelo original, para integrar o coeficiente de distorção radial, é assumida a existência de dois pontos conjugados: um representando a projecção ideal no plano imagem e o outro correspondendo à imagem distorcida no mesmo plano. A relação entres estes dois pontos está definida nas Eqs. 16a e 16b. Combinando estas equações com as equações de projecção perspectiva obtém-se:

$$X_d(1+k_1r^2) = f \frac{r_1x_w + r_2y_w + r_3z_w + t_x}{r_7x_w + r_8y_w + r_9z_w + t_z}, \quad Y_d(1+k_1r^2) = f \frac{r_4x_w + r_5y_w + r_6z_w + t_y}{r_7x_w + r_8y_w + r_9z_w + t_z} \quad (Eqs. 34a \text{ e } 34b)$$

onde $r^2 = X_d^2 + Y_d^2$. Qualquer uma das duas últimas equações permite a determinação de t_z , f e k_1 , através do conhecimento prévio da matriz de rotação 3D e das componentes segundo x e y do vector de translação. Como o factor de escala segundo y , assumido inicialmente, é mais correcto que o factor de escala segundo x , é utilizada preferencialmente a Eq. 34b em vez da Eq. 34a. Ambas as equações incluem a distância focal f que possui dois desempenhos diferentes no procedimento de calibração:

1. se for utilizado um plano de calibração ortogonal ao eixo óptico, a distância focal f necessita de ser considerada como um parâmetro predefinido, pois utilizando estas equações não é possível determinar um mínimo correcto;
2. contudo, se for utilizado um plano de calibração com um ângulo de incidência superior a 20 grau, então a distância focal f pode ser considerada como um parâmetro desconhecido e determinada pela Eq. 34b.

Esta restrição, a da necessidade da existência de um ângulo de incidência entre o plano de calibração e o eixo óptico, é em certas aplicações uma desvantagem importante. Neste método, este problema é solucionado por novas equações para o cálculo da distância focal f , utilizando um valor inicial para a distância focal (obtido pelas especificações da lente) e pela lei para as lentes de Newton. Deste modo são assumidas duas distâncias focais diferentes, sendo uma a distância focal efectiva f e a outra, a distância focal da lente f , Fig. 11.

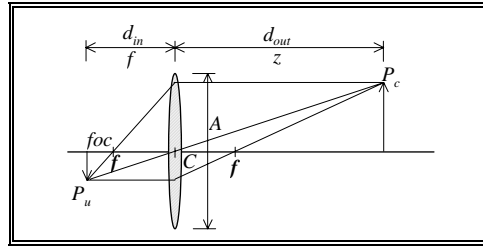


Fig. 11 - O modelo de Newton para a lente.

Como se pode observar da Fig. 11, a distância focal efectiva resulta da adição da distância focal da lente e da distância de focagem foc (isto é $f = f + foc$). Combinando esta observação com a lei para as lentes de Newton obtém-se $\frac{1}{f} + \frac{1}{z} = \frac{1}{f} = \frac{1}{f + foc} + \frac{1}{z}$, resultando para a distância de focagem foc e para a distância focal efectiva f as equações:

$$foc = \frac{f^2}{z-f}, \quad f = f + foc = \frac{zf}{z-f} \quad (\text{Eqs. 35a e 35b})$$

Com o intuito de obter um valor mais preciso para a distância focal efectiva, é utilizada esta última equação conjuntamente com a equação resultante da relação de coordenadas definida no plano que inclui a linha de projecção e o eixo óptico da lente, como representado na Fig. 12. A partir desta relação de coordenadas, obtém-se a equação:

$$\frac{f}{\sqrt{X_u^2 + Y_u^2}} = \frac{z}{\sqrt{x^2 + y^2}}, \quad (\text{Eq. 36})$$

que, conjuntamente com as Eqs. 35a e 35b, define as equações para a determinação da distância focal efectiva:

$$\begin{bmatrix} \sqrt{x_i^2 + y_i^2} \\ z_i - f \end{bmatrix} [f] = \begin{bmatrix} z_i \sqrt{X_{u_i}^2 + Y_{u_i}^2} \\ z_i f \end{bmatrix} \text{ com } i = 1, \dots, n. \quad (\text{Eq. 37})$$

1.5.6 - Determinação dos factores de escala de imagem, das coordenadas do centro da imagem na memória frame e dos desvios ortogonais do referencial imagem

Até este ponto, foram já descritos procedimentos para a determinação de todos os parâmetros extrínsecos, assim como de alguns parâmetros intrínsecos. Estes parâmetros, conjuntamente com os parâmetros cujos valores foram assumidos, caracterizam o modelo de calibração. Contudo, os valores dos parâmetros determinados são afectados pelos erros devidos à atribuição realizada no início do processo de calibração dos parâmetros intrínsecos.

Baseado no conhecimento destes parâmetros e utilizando as Eqs. 17a e 17b, pode-se determinar uma nova solução para os parâmetros intrínsecos a_x, a_y, b_x, b_y, C_x e C_y , utilizando a seguinte relação matricial:

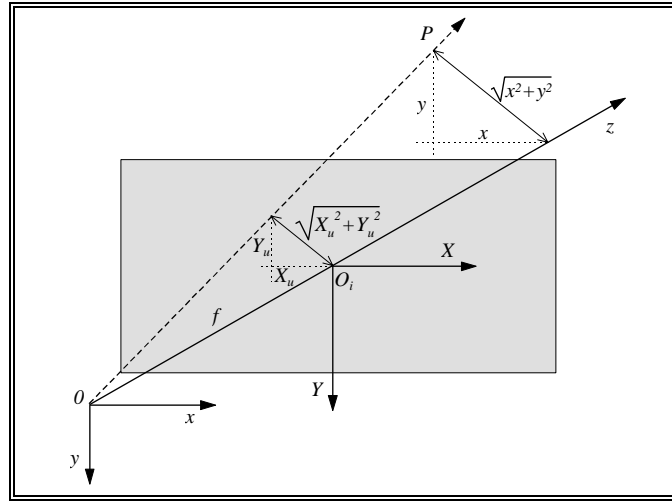


Fig. 12 - Relação de coordenadas definida no plano que inclui a linha de projecção e o eixo óptico da lente.

$$\begin{bmatrix} X_{d_i} & 0 & Y_{d_i} & 0 & 1 & 0 \\ 0 & X_{d_i} & 0 & Y_{d_i} & 0 & 1 \end{bmatrix} A = \begin{bmatrix} X_{f_i} \\ Y_{f_i} \end{bmatrix} \text{ com } i = 1, \dots, n \text{ e } A = [a_x \ a_y \ b_x \ b_y \ C_x \ C_y]^T. \quad (\text{Eq. 38})$$

As coordenadas distorcidas dos pontos de calibração podem ser facilmente obtidas através das transformações definidas nos passos 1, 2 e 3, descritas no ponto 1.5.2, pois todos os parâmetros desconhecidos envolvidos nestas transformações foram já determinados. As coordenadas na memória *frame* (X_f, Y_f) dos pontos de calibração, podem ser obtidas directamente a partir da imagem na memória *frame*.

Os novos valores determinados para estes parâmetros são muito mais precisos que os inicialmente assumidos.

1.5.7 - Calibração iterativa

Após a determinação de valores mais precisos para os parâmetros intrínsecos ($a_x, a_y, b_x, b_y, C_x, C_y$), os parâmetros calibrados podem ser mais precisos, se forem utilizados num processo de calibração iterativo, Fig. 13.

Baseado em tal facto, pode-se recalibrar novamente utilizando desta vez as Eqs. 17a e 17b, em vez das equações utilizadas nos métodos apresentados nos pontos 1.2 e 1.4. Uma variável para definir o critério de convergência do processo iterativo pode ser a distância focal efectiva f .

A determinação dos parâmetros de calibração pode ser novamente melhorada, se for utilizada uma correcção das coordenadas dos pontos de calibração na memória *frame*, através de funções polinomiais de segunda ordem. Este processo de correcção tende a repor as actuais coordenadas imagem na memória *frame* (X_f, Y_f), novamente para as coordenadas imagem originais na mesma memória (X'_f, Y'_f), dos pontos de calibração utilizados, corrigindo assim possíveis componentes de distorção resultantes de parâmetros ou pontos de calibração com ruído. A transformação de correcção é:

$$X'_f = a_0 + a_1 X_f + a_2 Y_f + a_3 X_f^2 + a_4 X_f Y_f + a_5 Y_f^2, \quad Y'_f = b_0 + b_1 X_f + b_2 Y_f + b_3 X_f^2 + b_4 X_f Y_f + b_5 Y_f^2, \quad (\text{Eqs. 39a e 39b})$$

onde os coeficientes $a_{i,i=0,\dots,5}$ e $b_{i,i=0,\dots,5}$ são constantes de parametrização do processo de correcção para a câmara, e podem ser representados pelos dois vectores $\bar{a} = [a_0 \ a_1 \ a_2 \ a_3 \ a_4 \ a_5]^T$ e $\bar{b} = [b_0 \ b_1 \ b_2 \ b_3 \ b_4 \ b_5]^T$.

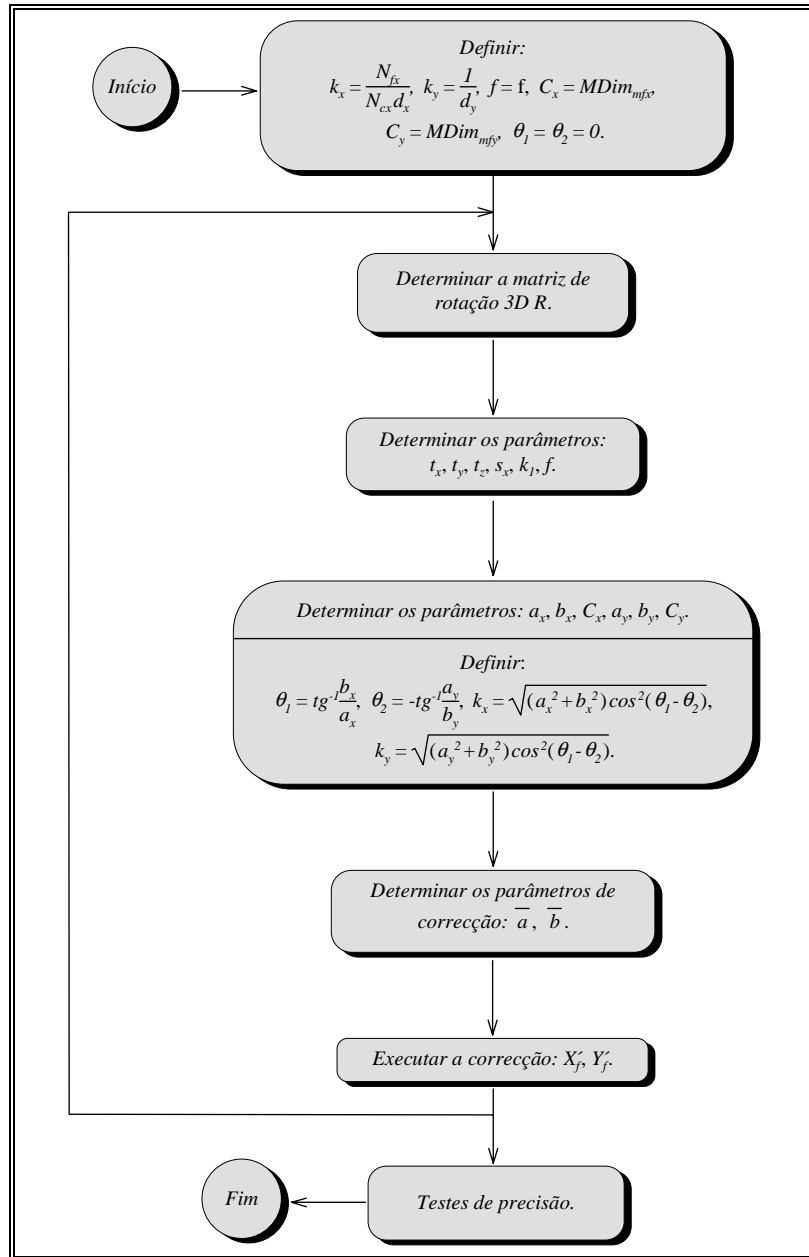


Fig. 13 - Esquema do processo iterativo para a calibração de uma dada câmara. Nesta figura $MDIM_{mfx}$ e $MDIM_{mfy}$ representam metade da dimensão da memória frame segundo as direcções x e y , respectivamente.

O processo de correcção tem como função minimizar o erro entre a posição dos pontos ideais (X_f^i, Y_f^i) e a dos pontos corrigidos $(X_f'^i, Y_f'^i)$. O erro (em *pixels* quadrados) entre estes pontos homólogos é $Erro = (X_f^i - X_f'^i)^2 + (Y_f^i - Y_f'^i)^2$. A norma Euclidiana de um vector r de dimensão n , isto é $\|r\| = \sqrt{|r_1|^2 + \dots + |r_n|^2}$, pode ser utilizada para descrever o erro para n pontos como $Erro = \|\overline{X}_i - P \overline{a}\|^2 + \|\overline{Y}_i - P \overline{b}\|^2$, onde \overline{X}_i e \overline{Y}_i são vectores coluna formados pelas coordenadas $X_f'^i$ e $Y_f'^i$ respectivamente, e P a matriz:

$$P = \begin{bmatrix} 1 & X_{f_1} & Y_{f_1} & X_{f_1}^2 & X_{f_1} Y_{f_1} & Y_{f_1}^2 \\ 1 & X_{f_2} & Y_{f_2} & X_{f_2}^2 & X_{f_2} Y_{f_2} & Y_{f_2}^2 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & X_{f_n} & Y_{f_n} & X_{f_n}^2 & X_{f_n} Y_{f_n} & Y_{f_n}^2 \end{bmatrix},$$

que inclui os termos de segunda ordem das coordenadas imagem dos pontos de calibração (X_f, Y_f) .

Os vectores \bar{a} e \bar{b} , que minimizam o erro do processo de correcção, podem ser obtidos utilizando por exemplo o processo pseudo inverso de Moore-Penrose⁵:

$$\bar{a} = (P^T P)^{-1} P^T \bar{X}_p, \quad \bar{b} = (P^T P)^{-1} P^T \bar{Y}_p$$

O processo de correcção pode ser ainda mais preciso se a imagem original na memória *frame* for dividida em várias janelas e ele for aplicado a cada uma destas janelas.

1.5.5 - Algumas observações

Este método de calibração tem como grandes vantagens o modelamento utilizado para a câmara, nomeadamente: a inclusão da distorção radial da lente; a consideração da possibilidade de os eixos segundo as direcções x e y , do referencial de coordenadas na memória *frame*, não serem paralelos aos eixos respectivos do referencial imagem; a utilização de pontos coplanares sem o requisito de o plano de calibração não ser paralelo ao plano câmara; e a determinação pelo processo de calibração de todos os parâmetros utilizados no modelamento. Contudo, apresenta como desvantagens o facto de necessitar de estimativas iniciais para certos parâmetros (nomeadamente para os parâmetros k_x e f), o que em certas aplicações pode ser inconveniente, assim como uma elevada dependência de todo o processo de calibração da determinação dos três ângulos de rotação 3D, que por sua vez dependem fortemente de apenas quatro pontos - os quatro vértices do rectângulo utilizado na determinação da orientação 3D da câmara.

1.6 - Método de calibração concebido e implementado no âmbito desta dissertação

No âmbito desta dissertação, foi concebido e implementado um método de calibração de câmaras, baseado no proposto em [Tsai, 1987] e apresentado no ponto **1.4**; deste modo, utiliza o mesmo modelo para a câmara, um *setup* experimental idêntico e, da mesma forma, um conjunto de pontos de calibração coplanares; no entanto, tem a principal vantagem de a determinação das coordenadas do centro da imagem na memória *frame* poder ser, desta feita, realizada pelo processo de calibração. A determinação do factor de incerteza horizontal continua a não ser possível pelo processo de calibração propriamente dito; no entanto, foi desenvolvido um método para a sua determinação uma determinação independente dos restantes parâmetros de calibração. O processo de calibração global é constituído pelas seguintes três etapas:

- I-** determinação do factor de incerteza horizontal s_x ;
- II-** determinação das coordenadas do centro da imagem na memória *frame* C_x e C_y ;
- III-** determinação dos restantes parâmetros de calibração, nomeadamente da matriz de rotação 3D R , do vector de translação 3D T , da distância focal efectiva f e do factor de distorção radial k_r .

Na *Fig. 14* estão representadas estas três etapas do processo de calibração global. Obviamente, as duas primeiras etapas apenas necessitam de ser realizadas uma única vez, para um mesmo conjunto de

⁵ Ver, por exemplo [Chapra, 1988].

câmara, lente e *hardware* de aquisição de imagem; as duas últimas etapas podem-se fundir numa única, como mais adiante será descrito.

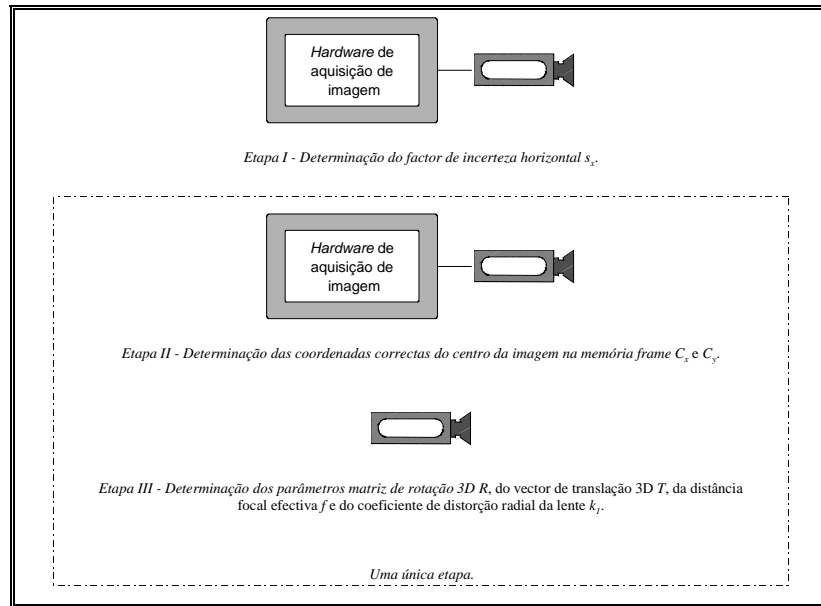


Fig. 14 - Etapas de determinação dos parâmetros intrínsecos e extrínsecos de uma câmara utilizadas no método global de calibração concebido e implementado no âmbito desta dissertação.

1.6.1 - Determinação do factor de incerteza horizontal s_x

O método descrito neste ponto para a determinação do factor de incerteza horizontal s_x baseia-se no apresentado em [Lenz, 1988] e aplica-se apenas a câmaras de elementos sensores discretos, por exemplo CCD, CID, MOS, etc. O factor de incerteza horizontal s_x é utilizado para a determinação do espaçamento d'_x entre *pixels* adjacentes no sensor de imagem como surgem na memória *frame*, Eq. 6c.

Uma fraca estimativa para s_x é obtida por:

$$s_x = \frac{N_{cx}}{N_{fx}}, \quad (Eq. 40)$$

onde N_{cx} é o número de elementos sensores na direcção x e N_{fx} é o número de elementos imagem numa linha discretizada pelo computador.

A Eq. 40 apenas é válida quando o tempo de activação da parcela da linha vídeo é igual ao do circuito de digitalização. Geralmente isto não se verifica. Desvios de mais do que cinco por cento têm sido observados. O factor de incerteza horizontal s_x é realmente igual a

$$s_x = \frac{f_c}{f_f}, \quad (Eq. 41)$$

onde f_c é a frequência de relógio dos elementos sensores da câmara e f_f é a frequência de amostragem do conversor analógico/digital.

Se o conversor analógico/digital e os elementos sensores da câmara tiverem frequência de relógio conhecidas e estáveis, então a Eq. 41 origina uma estimativa bastante precisa para o factor de incerteza horizontal s_x . Contudo, mesmo que ambos os relógios sejam excitados por cristal, pode não ser possível utilizar as suas frequências específicas pois, geralmente, um relógio é utilizado como mestre e o outro é posto em fase através dos sinais de sincronização TV. Se o utilizador tiver à sua disposição o equipamento apropriado, uma maneira de determinar o factor de incerteza horizontal s_x é simplesmente medir as frequências reais.

O método descrito neste ponto não necessita do conhecimento das frequências dos relógios. Utiliza uma propriedade específica comum a todas as câmaras de elementos sensor discretos (por exemplo câmaras Fairchild, GE, Javelin), que é o sinal analógico de saída ser formado pela sobreposição ao valor de saída ideal representando a intensidade da imagem, de ruído e pequenos picos de sinais da mesma frequência do varrimento dos elementos sensores da câmara. Estes picos são normalmente suprimidos por filtros passa-baixo existentes entre a câmara e o conversor analógico/digital. Contudo, removendo estes filtros passa-baixo, a imagem digitalizada apresentará *aliasing* na forma de faixas verticais. A frequência espacial f_s destas faixas corresponde à diferença de frequências entre f_c e f_f :

$$f_s = \frac{|f_c - f_f|}{f_f}. \quad (\text{Eq. 42})$$

Da Eq. 41 e da Eq. 42, o factor de incerteza horizontal s_x torna-se:

$$s_x = 1 \pm f_s \quad (\text{Eq. 43})$$

onde o sinal - é valido quando $f_f > f_c$ ou de forma equivalente $N_{fx} > N_{cx}$. A frequência espacial f_s é determinada pela análise do espectro $S_i(0, \dots, N-1)$ resultante da transformada discreta de Fourier⁶ de uma linha i . Distintos picos no espectro de amplitude $AS_i(0, \dots, N-1)$ são gerados por f_s e pelos seus harmónicos. A frequência f_s é determinada procurando um máximo no espectro de amplitude; uma boa indicação da localização deste máximo é dada pelas Eqs. 40 a 43. Com o objectivo de aumentar a precisão, deve-se interpolar a representação espectral acrescentando zeros antes de realizar a transformada discreta de Fourier. Também com este objectivo, deve-se utilizar a zona central da imagem para evitar margens negras e, em vez de uma linha da imagem, deve-se considerar a média de várias linhas da mesma. Na Fig. 15 está representado, de forma esquemática, o método para a determinação do factor de incerteza horizontal.

1.6.2 - Determinação das coordenadas do centro da imagem na memória frame

O centro da imagem é definido pelas coordenadas (C_x, C_y) na memória *frame* do ponto de intersecção do eixo óptico com o plano imagem. Geralmente é utilizado como origem do processo de formação de uma imagem e pertence aos parâmetros presentes nas equações de projecção perspectiva. Em aplicações de alta precisão, é utilizado também como centro do modelamento da distorção radial da lente.

Em várias aplicações de visão por computador, por exemplo análise de padrões 2D, considera-se o centro da imagem como localizado no centro da memória *frame*; contudo, normalmente, existem consideráveis desvios⁷, o que origina a necessidade da determinação do centro da imagem correcto para aplicações de maior precisão, como por exemplo em visão tridimensional por computador.

Existem vários métodos para a determinação do centro da imagem; neste ponto, após a descrição da influência de uma determinação incorrecta do centro da imagem na restrição do alinhamento radial proposta em [Tsai, 1987] e apresentado nos pontos 1.2 e 1.4, é apresentado o método concebido e implementado, baseado num método apresentado em [Lenz, 1988]; são também apresentados mais dois métodos, descritos inicialmente também em [Lenz, 1988].

1.6.2.1- Influência de uma determinação incorrecta do centro da imagem na restrição do alinhamento radial

Seja $(\Delta C_x, \Delta C_y)$ a diferença entre as coordenadas no plano imagem do centro ideal da imagem e do centro determinado incorrectamente. Então, as coordenadas (X_d, Y_d) pelas Eqs. 7a e 7b, tornam-se:

⁶ Ver, por exemplo [Chapra, 1988].

⁷ Experiências reportadas por vários investigadores na área, indicam que por vezes surgem desvios da ordem dos 25 *pixels* e que não seria grande surpresa se surgissem desvios na ordem dos 30/40 *pixels* [Tsai, 1987].

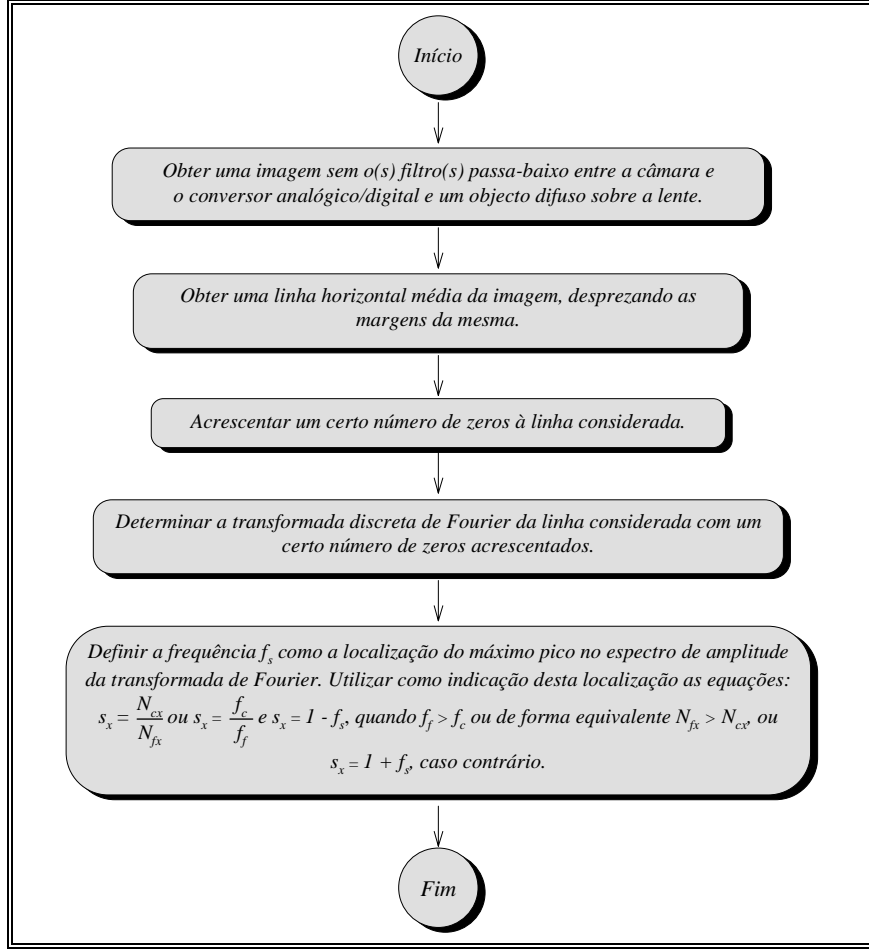


Fig. 15 - Método para a determinação do factor de incerteza horizontal concebido e implementado no âmbito desta dissertação.

$$X'_d = X_d + \Delta C_x = \frac{f}{z(1 + k_1 r^2)} \left(x + \Delta C_x \frac{f}{z} + \Delta C_x \frac{f}{z} k_1 r^2 \right), \quad (\text{Eq. 44a})$$

$$Y'_d = Y_d + \Delta C_y = \frac{f}{z(1 + k_1 r^2)} \left(y + \Delta C_y \frac{f}{z} + \Delta C_y \frac{f}{z} k_1 r^2 \right). \quad (\text{Eq. 44b})$$

Da Eq. 3 e da Eq. 10, obtém-se:

$$Y'_d x_w r'_1 + Y'_d y_w r'_2 + Y'_d t'_x - X'_d x_w r'_4 - X'_d y_w r'_5 - X'_d t'_y = \Delta C_x X'_d \frac{z}{f} k_1 r^2 - \Delta C_y Y'_d \frac{z}{f} = k_1 r^2, \quad (\text{Eq. 45})$$

onde $r'_1 = r_1 + \alpha r_7$, $r'_2 = r_2 + \alpha r_8$, $r'_4 = r_4 + \beta r_7$, $r'_5 = r_5 + \beta r_8$, $t'_x = t_x + \alpha t_z$, $t'_y = t_y + \beta t_z$, $\alpha = \Delta C_x / f$, $\beta = \Delta C_y / f$.

Comparando a Eq. 45 com a Eq. 10, verifica-se que a restrição do alinhamento radial apresenta um erro residual devido a $(\Delta C_x, \Delta C_y)$ com a forma:

$$ERRO = \sqrt{k_1^2 (K_x \Delta C_x^2 + K_y \Delta C_y^2)}. \quad (\text{Eq. 46})$$

As constantes K_x e K_y dependem do *setup* experimental que está a ser utilizado.

1.6.2.2 - Método para a determinação das coordenadas correctas do centro da imagem na memória frame

Como o centro da imagem está presente na Eq. 10, uma maneira de estimar o centro da imagem é procurar pelo centro que minimiza o erro residual da Eq. 10. Apesar de se tratar de uma optimização não linear com duas incógnitas, torna-se numa abordagem atractiva pois são necessárias poucas iterações para a obtenção da solução correcta, devido à dependência sistemática do erro residual com ΔC_x e ΔC_y , verificada pela Eq. 46. Esta equação indica também que este método apenas resulta na presença de distorção radial da lente e quando o erro residual devido à extracção das coordenadas imagem na memória frame dos pontos de calibração for pequeno, quando comparado com a Eq. 46.

A equação resultante a otimizar é:

$$\Delta C_x \left[\frac{Y_{d_i}}{f} (x_{w_i} r_7 + y_{w_i} r_8 + t_z) - x_{w_i} r_4 - y_{w_i} r_5 - t_y \right] + \Delta C_y \left[-\frac{X_{d_i}}{f} (x_{w_i} r_7 + y_{w_i} r_8 + t_z) + x_{w_i} r_1 + y_{w_i} r_2 + t_x \right] = -Y_{d_i} (x_{w_i} r_1 + y_{w_i} r_2 + t_x) + X_{d_i} (x_{w_i} r_4 + y_{w_i} r_5 + t_y) \quad (Eq. 47)$$

com $i = 1, \dots, N$, sendo N o número de pontos de calibração coplanares utilizados. A Eq. 47 pode ser resolvida em ordem às incógnitas ΔC_x e ΔC_y , por exemplo pelo método Levenberg-Marquardt, com a solução inicial $\Delta C_x = \Delta C_y = 0$. Com o objectivo de aumentar a precisão, deve-se proceder à actualização dos restantes parâmetros presentes na Eq. 47, de forma adequada, isto é, em primeiro deve-se determinar os restantes parâmetros presentes na Eq. 47 como foi descrito anteriormente no ponto 1.4, seguidamente determinar ΔC_x e ΔC_y , depois voltar a determinar os restantes parâmetros presentes na Eq. 47, determinar novamente ΔC_x e ΔC_y , até a uma solução global aceitável, Fig. 16.

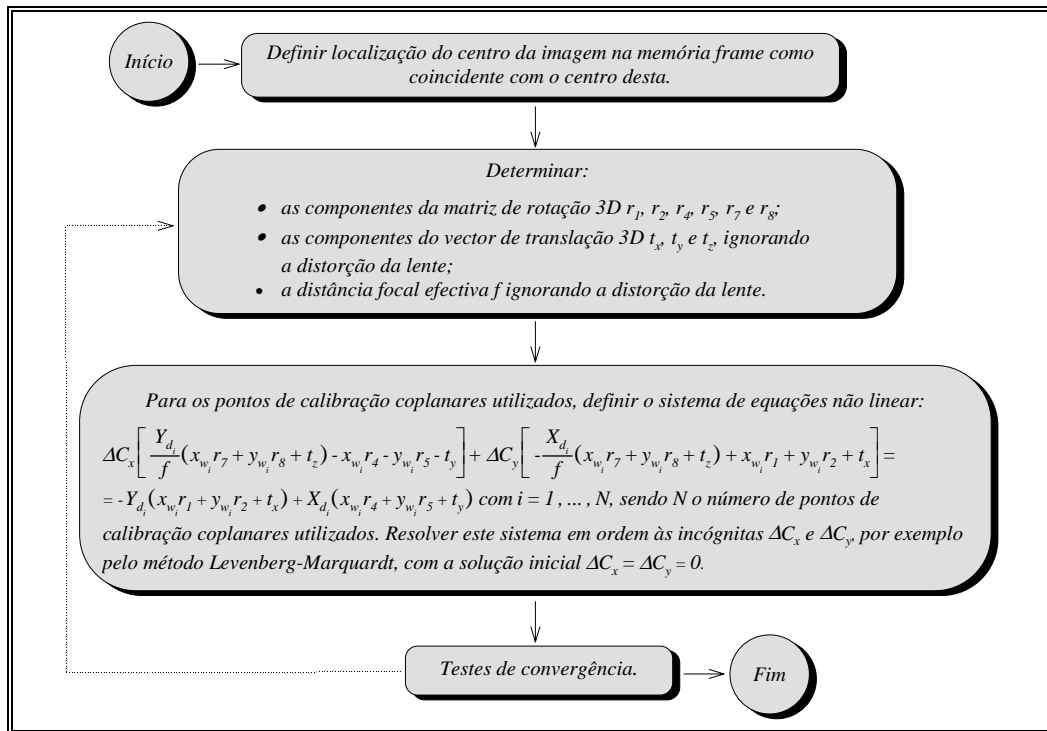


Fig. 16 - Esquema do método iterativo para a determinação das coordenadas correctas do centro da imagem concebido e implementado no âmbito desta dissertação.

Este método foi o implementado no âmbito desta dissertação, sendo por isso recomendado; no entanto, no ponto seguinte são apresentados outros dois métodos, inicialmente apresentados em [Lenz, 1988].

1.6.2.2.1 - Outros métodos para a determinação das coordenadas correctas do centro da imagem na memória frame

Neste ponto são apresentados mais dois métodos, inicialmente apresentados em [Lenz, 1988], para a determinação das coordenadas correctas do centro da imagem na memória *frame*. Trata-se do método óptico, onde é utilizada uma fonte laser, e do método da variação da distância focal, que utiliza para o efeito um sistema de câmara e lente com distância focal ajustável, um conjunto de duas lentes ou uma única lente de distância focal ajustável.

I- Método óptico para a determinação das coordenadas correctas do centro da imagem na memória frame

Este método para a determinação das coordenadas correctas do centro da imagem na memória *frame*, usa a reflexão de superfícies ópticas e foi inicialmente proposto em [Lenz, 1988]. Contudo, em vez da utilização de um espelho como superfície reflectora, é utilizada uma lente previamente montada na câmara.

Quando um feixe laser é apontado para a lente, previamente montada, parte deste é reflectido aquando da sua entrada no vidro e, do mesmo modo, aquando da sua saída do mesmo. Assim, reflexões múltiplas ocorrem quando o feixe laser é apontado para a lente que podem ser observadas sobre um alvo colocado à frente da fonte laser e que contenha um orifício de passagem para o feixe incidir sobre a referida lente. Com alguma habilidade, o feixe laser incidente pode ser direccionado de tal forma que todos os feixes reflectidos coincidam com o primeiro ficando, desta forma, alinhado com o eixo óptico. Quando o alinhamento é obtido, um filtro de atenuação é colocado entre a lente e a fonte laser, a câmara pode ser alimentada e a localização da imagem do feixe laser na memória *frame* coincide com a localização correcta do centro da imagem.

Se a lente for desmontada da câmara, a reflexão do plano sensor dá uma indicação da perpendicularidade deste relativamente ao eixo óptico. Quando um feixe laser de pequena potência (< 10 mW) é utilizado, não existe a possibilidade de ocorrência de danos nos elementos discretos do plano sensor.

Segundo a descrição em [Lenz, 1988], este método origina os resultados mais reprodutíveis.

II- Método de variação da distância focal para a determinação das coordenadas correctas do centro da imagem na memória frame

Quando a distância focal efectiva de um sistema câmara-lente é alterada, a imagem sofrerá um processo de *zoom*. Apenas um ponto da imagem, o centro do *zoom*, permanecerá inalterado. Utilizando um modelo *pinhole* para a câmara, e assumindo que o centro da lente se desloca sobre o eixo óptico quando a distância focal é variada, o centro do *zoom* é obviamente a intersecção do eixo óptico com o plano imagem, isto é, o centro da imagem. Dois processos para variação da distância focal efectiva serão apresentados:

a) **Varição da distância de uma lente:** Uma alteração na distância de uma lente provoca que a distância focal efectiva f se altere para f' . Das Eqs. 4a, 4b, 5a, 5b, 6a, 6b e 6c, obtém-se:

$$\frac{X'_f - C_x}{X_f - C_x} = \frac{X'_d}{X_d} = \frac{f'z(1 + k_1r^2)}{fz(1 + k_1r^2)} = \frac{Y'_d}{Y_d} = \frac{Y'_f - C_y}{Y_f - C_y} (\cong 1 + z_f). \quad (Eq. 48)$$

Deste modo:

$$C_x(Y_f - Y'_f) + C_y(X'_f - X_f) = X'_f Y_f - X_f Y'_f, \quad (Eq. 49)$$

onde (X_f, Y_f) são as coordenadas na memória *frame* dos pontos de calibração utilizados antes da alteração da distância focal, (X'_f, Y'_f) são as coordenadas, na mesma memória, dos referidos pontos depois da alteração e z_f é o factor de *zoom* ($z_f = 0$ indica que não houve alteração).

A Eq. 49 pode ser resolvida em ordem às coordenadas correctas do centro da imagem na memória *frame* C_x e C_y utilizando dois ou mais pontos de calibração, por exemplo pelo método de regressão dos mínimos quadrados, e é independente da distorção da lente, dos factores de escala e das coordenadas 3D dos pontos de calibração utilizados.

Infelizmente, a consideração de que o centro da lente se desloca sobre o eixo óptico quando a distância focal é alterada não é exactamente verdadeira. Quando o anel da lente é rodado para variar a distância focal, a lente desloca-se normalmente segundo uma hélice. A este tipo de movimento é somado um outro, de forma aleatória, devido ao sistema de roscagem do referido anel. Mesmo quando o movimento global é pequeno, o que é geralmente o caso ($\cong 20 \mu\text{m} \cong 1$ píxel [Lenz, 1988]), este tipo de deslocamento tem um considerável efeito sobre a localização do centro de *zoom*. A razão deste facto é que este tipo de deslocação implica um erro sistemático em (X'_f, Y'_f) , que é amplificado com o inverso do factor de *zoom* $1/z_f$ como se depreende da Eq. 48. Um valor típico para o factor de *zoom*, utilizando uma gama de ajuste da distância focal de 25 mm, é $z_f = 0.05$ [Lenz, 1988]. Para factores de *zoom* desta ordem de grandeza, a diferença entre o centro de *zoom* e o centro correcto da imagem pode ser da ordem dos 20 *pixels*. Como o erro do movimento em hélice é sistemático por natureza, pode ser compensado pela consideração de várias imagens e registando o ângulo de rotação relativa do anel de ajuste da distância focal. Todavia, o movimento aleatório devido ao sistema de roscagem não pode ser compensado. Deste modo, este método dá resultados de pouca precisão e baixa reprodutibilidade [Lenz, 1988].

b) Variação da distância focal com a utilização de duas lentes: A ideia é a mesma do caso anterior; contudo, desta feita, a variação da distância focal pode ser maior e a imagem pode permanecer focada, o que não acontece no caso anterior. Novamente, utilizando-se dois ou mais pontos de calibração o centro correcto da imagem pode ser determinado pela Eq. 49.

A desvantagem deste método prende-se com o facto da necessidade da montagem e desmontagem das lentes durante o processo de calibração, o que pode provocar uma mudança no posicionamento da câmara se esta não estiver montada de forma suficientemente rígida. Outra desvantagem deste método prende-se com a precisão de fabrico das lentes; a localização relativa do eixo óptico das duas lentes em relação ao eixo de montagem na câmara geralmente varia muito mais de $20 \mu\text{m}$ [Lenz, 1988]. Estas desvantagens podem ser eliminadas pela utilização de uma lente com distância focal variável.

1.6.3 - Método de calibração de todos os parâmetros intrínsecos e extrínsecos de uma câmara exceptuando o factor de incerteza horizontal s_x

Neste ponto é apresentado o método de calibração de câmaras concebido e implementado no âmbito desta dissertação. Trata-se de um método de calibração com o qual é possível a determinação de todos os parâmetros intrínsecos e extrínsecos de uma câmara, exceptuando do factor de incerteza horizontal s_x . Desta forma, com este método, e utilizando um conjunto de pontos de calibração coplanares, é possível determinar as coordenadas correctas do centro da imagem na memória *frame* C_x e C_y , a matriz de rotação 3D R , o vector de translação 3D T , a distância focal efectiva f e o factor de distorção radial da lente k_l . Na realidade, este método de calibração é uma fusão do método iterativo para a determinação das coordenadas correctas do centro da imagem na memória *frame* apresentado no ponto 1.6.2.2, e o método de calibração de câmaras apresentado no ponto 1.4. A Fig. 17 representa o esquema do método. Como se observa na Fig. 17, as coordenadas correctas do centro da imagem na memória *frame* C_x e C_y são determinadas utilizando o método iterativo apresentado no ponto 1.6.2.2, sendo os restantes parâmetros envolvidos nesta determinação - a matriz de rotação 3D R , o vector de translação 3D T e a distância focal efectiva f - determinados segundo o método apresentado no ponto 1.4, ignorando a distorção da lente; os restantes parâmetros da calibração - a matriz de rotação 3D R , o vector de translação 3D T , a distância focal efectiva f e o factor de distorção radial da lente k_l - são determinados do mesmo modo que no método apresentado no ponto 1.4.

Na implementação do método, o critério utilizado na verificação da convergência da determinação das coordenadas do centro da imagem na memória *frame* foi que a diferença entre as coordenadas

obtidas não diferisse mais do que um dado valor, arbitrado pelo utilizador, em duas iterações consecutivas.

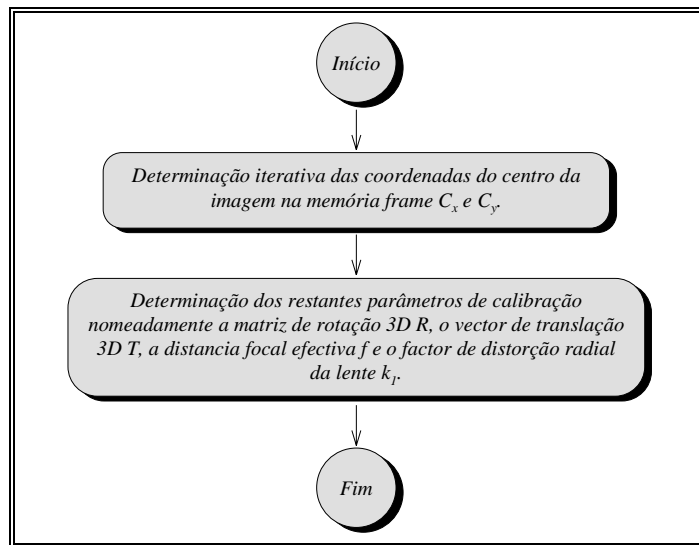


Fig. 17 - Esquema do método de calibração implementado.

1.6.3.1 - Algumas observações

Este método de calibração tem como grandes vantagens o modelamento utilizado para a câmara, nomeadamente: a inclusão da distorção radial da lente e projecção perspectiva; a utilização de pontos coplanares; a não necessidade de estimativas iniciais para qualquer parâmetro; a possibilidade da determinação das coordenadas do centro da imagem na memória *frame* conjuntamente com os restantes parâmetros, exceptuando o factor de incerteza horizontal. Contudo, apresenta como desvantagens: a necessidade de o plano de calibração não ser paralelo ao plano câmara, que em certas aplicações pode ser uma desvantagem; a não consideração da possibilidade de os eixos, segundo as direcções x e y do referencial de coordenadas na memória *frame*, não serem paralelos aos eixos respectivos do referencial imagem; a necessidade de determinar o factor de incerteza horizontal por um processo independente que, embora possa só ser determinado uma única vez para o mesmo sistema câmara e lente (por exemplo utilizando o método apresentado no ponto 1.6.1), em certas aplicações pode ser indesejável se não mesmo impossível - nestes casos, deve-se utilizar as estimativas para o factor de incerteza horizontal dadas pelas Eqs. 40 e 41, sendo esta última a que corresponde a estimativas mais correctas.

2 - Determinação das coordenadas dos pontos de calibração na memória *frame*

Quando se pretende calibrar uma dada câmara, ou um conjunto de câmaras, deve-se primeiro determinar as coordenadas dos pontos de calibração na memória *frame*, Fig. 18. Como já foi referido na secção anterior, a qualidade dos resultados obtidos por um qualquer método de calibração depende fortemente da precisão na determinação das coordenadas dos pontos de calibração na memória *frame*. Nesta secção, são apresentados alguns métodos existentes, assim como o concebido e implementado neste trabalho.

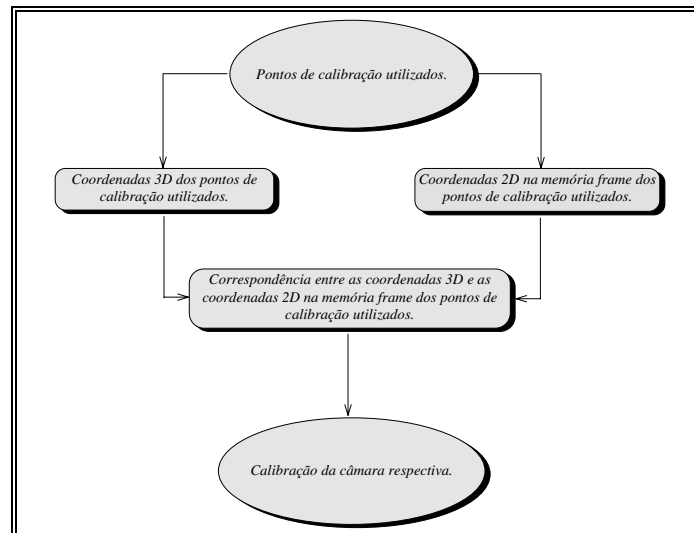


Fig. 18 - Necessidade da determinação das coordenadas na memória frame dos pontos de calibração utilizados.

2.1 - Alguns métodos existentes para a determinação das coordenadas dos pontos de calibração na memória frame

Um dos métodos mais utilizados para a determinação das coordenadas dos pontos de calibração na memória *frame* consiste na consideração dos centros das figuras geométricas presentes na(s) imagens(s) de calibração como pontos de calibração. Desta forma, surge a necessidade da determinação dos centros das figuras geométricas respectivas, Fig. 19.

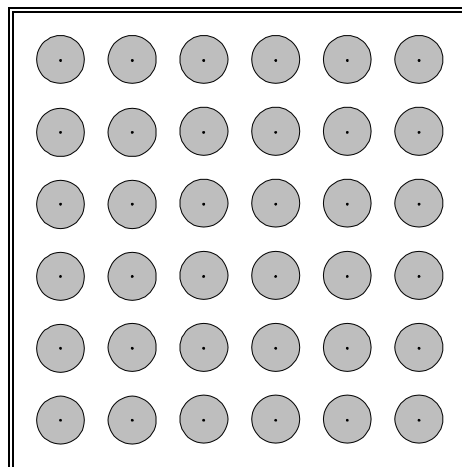


Fig. 19 - Neste exemplo, os centros dos círculos representados são considerados como os pontos de calibração utilizados.

Por exemplo, em [Lenz, 1988] esta determinação é efectuada do seguinte modo:

- 1) realizar a aquisição da imagem de cinzentos;
- 2) determinar as fronteiras dos círculos, utilizando interpolação na escala de cinzentos, com precisão ao nível do *subpixel*;
- 3) determinar os centros, com precisão ao nível de $1/30$ *pixel*, através da média da fronteira respectiva determinada na fase anterior.

Já em [Wu, 1989], onde os círculos considerados são de menor dimensão, esta determinação é efectuada do seguinte modo:

- 1) realizar a aquisição da imagem de cinzentos;
- 2) sujeitar a imagem de calibração, ou conjunto de imagens, a um processo de detecção de fronteiras pela utilização do operador de Sobel⁸;
- 3) detectar os pontos de cada círculo usando um padrão com forma quadrada de dimensões apropriadas;
- 4) detectar os pontos sob o padrão quadrado e determinar as suas coordenadas na imagem respectiva;
- 5) considerar as coordenadas x e y de cada ponto de calibração, como a média das coordenadas x e y dos *pixels* representando o ponto na imagem.

Em [Tsai, 1987], o método apresentado para a determinação das coordenadas na memória *frame* é diferente dos casos anteriores. Assim, neste método é considerado que a imagem de calibração é constituída por quadrados igualmente distribuídos, sendo os vértices destes os pontos de calibração a considerar, *Fig. 20*.

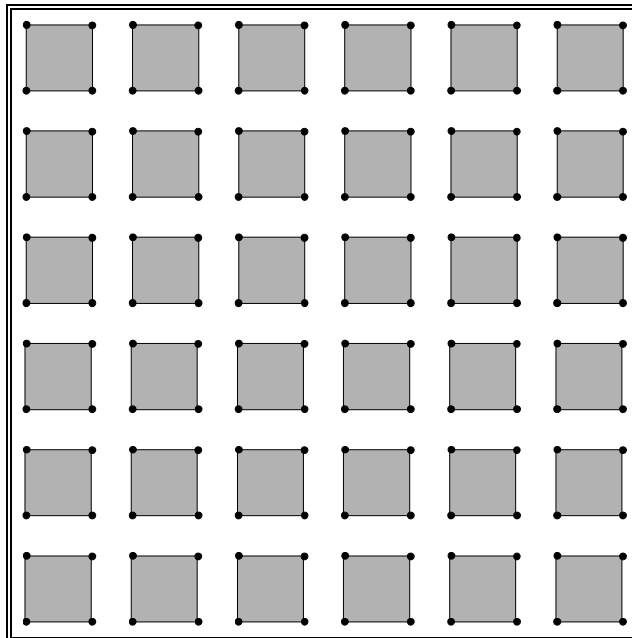


Fig. 20 - Consideração dos vértices dos quadrados como os pontos de calibração utilizados.

A determinação dos vértices dos quadrados, isto é, dos pontos de calibração, é neste método realizada do seguinte modo:

- 1) realizar a aquisição da imagem de cinzentos;
- 2) realizar uma limiarização da imagem de modo a obter-se uma imagem binária; o valor do limiar utilizado não é crítico e pode ser obtido manualmente pela observação do histograma da imagem inicial;
- 3) realizar a ligação dos pontos das fronteiras dos quadrados presentes na imagem binária, de modo a obter-se o conjunto dos lados aproximados dos quadrados;
- 4) percorrer a imagem original nas direcções perpendiculares às dos lados aproximados dos quadrados, de modo a determinar os “verdadeiros” pontos dos lados dos quadrados utilizando para o efeito interpolação;
- 5) ajustar linhas rectas aos “verdadeiros” pontos dos lados dos quadrados;

⁸ Este detector é apresentado de forma sucinta no próximo capítulo desta dissertação.

- 6) determinar a intersecção entre as linhas rectas de ajuste, de modo a obter-se a localização dos pontos de calibração com precisão ao nível de $1/2$ a $1/3$ *pixel*.

Em [Batista, 1993] é referido um método que também considera a imagem de calibração, ou conjunto de imagens, constituída por quadrados igualmente distribuídos, sendo os vértices destes os pontos de calibração a considerar. A determinação dos vértices dos quadrados é, neste método, realizada do seguinte modo:

- 1) realizar a aquisição da imagem de cinzentos;
- 2) realizar uma limiarização da imagem de modo a obter-se uma imagem binária;
- 3) realizar a detecção da localização aproximada dos vértices dos quadrados, utilizando análise recursiva dos contornos;
- 4) ajustar linhas rectas aos vértices dos quadrados cuja localização aproximada foi determinada no passo anterior;
- 5) determinar a intersecção entre as linhas rectas de ajuste, de modo a obter-se a localização dos pontos de calibração.

2.2 - Método para a determinação das coordenadas dos pontos de calibração na memória *frame* concebido e implementado no âmbito desta dissertação

Neste ponto, é apresentado o método para determinação das coordenadas na memória *frame* dos pontos de calibração, que foi concebido e implementado no âmbito desta dissertação. É considerado que a imagem de calibração é constituída por figuras geométricas com quatro lados paralelos dois a dois e igualmente distribuídas; exemplos de tais figuras são quadrados, rectângulos, etc. Os vértices de tais figuras geométricas são considerados como os pontos de calibração pretendidos. A precisão conseguida na localização dos vértices das figuras geométricas situa-se ao nível do *subpixel*.

O referido método consiste então nas seguintes etapas:

- 1) realizar a aquisição da imagem de cinzentos;
- 2) sujeitar a imagem de calibração a um detector de orlas de intensidade⁹, para classificar os *pixels* da imagem original em termos de amplitude A e de direcção θ (entre 0 e 180 grau);
- 3) determinar as duas direcções principais θ_1 e θ_2 obtidas pela classificação do passo 2) que difiram pelo menos de um dado valor θ_d (determinado em função das figuras geométricas consideradas), isto é $|\theta_1 - \theta_2| \geq \theta_d$, considerando nesta determinação apenas a direcção dos *pixels* i que apresentem um valor de amplitude A_i igual ou superior a um dado limiar L (desprezando-se assim pontos de ruído);
- 4) classificação dos *pixels* resultantes da detecção das orlas de intensidade na imagem de calibração em duas classes C_1 e C_2 :
 - uma, constituída pelos *pixels* i que apresentam um valor de amplitude A_i igual ou superior a um dado limiar L e uma direcção θ_i aproximadamente igual a θ_1 , isto é $A_i \geq L$ e $(\theta_1 - \theta_c) \leq \theta_i \leq (\theta_1 + \theta_c)$, onde θ_c é a máxima diferença de direcção admissível na classificação;
 - outra, constituída pelos *pixels* i que apresentam um valor de amplitude A_i igual ou superior a um dado limiar L e uma direcção θ_i aproximadamente igual a θ_2 , isto é $A_i \geq L$ e $(\theta_2 - \theta_c) \leq \theta_i \leq (\theta_2 + \theta_c)$;

⁹ Estes detectores são apresentados de forma sucinta no próximo capítulo desta dissertação.

- 5) determinar as linhas rectas constituídas pelos lados das figuras geométricas na classe C_1 ; esta determinação de cada linha recta L_i é realizada do seguinte modo:
 - a) determinar o primeiro elemento da classe C_1 ainda não considerado, classificá-lo como já considerado;
 - b) definir a recta que passa pelo ponto anteriormente determinado e que tem direcção igual a θ_i ;
 - c) determinar o próximo ponto na classe C_1 ainda não considerado que pertence à linha recta L_i ; esta determinação é baseada na distância do ponto que está a ser considerado à linha recta definida em b); se esta distância for menor do que um dado limiar D_{max} o ponto é definido como pertencente à linha L_i e classificado como já considerado;
 - d) retornar a b) até ser atingido o último elemento da classe C_1 ;
 - e) executar a regressão linear dos pontos classificados como pertencentes à linha L_i , caso o número destes seja superior a um dado limiar N (podendo-se assim eliminar falsas linhas), de forma a obter a equação da linha recta L_i ;
 - f) determinar uma nova linha voltando ao ponto a), até não existirem mais elementos na classe C_1 ainda não considerados.
- 6) determinar as linhas rectas constituídas pelos lados das figuras geométricas na classe C_2 ; esta determinação é realizada do mesmo modo que no caso da classe C_1 considerando, no entanto, θ_2 em vez de θ_1 ;
- 7) determinar as intersecções das linhas rectas determinadas na classe C_1 com as linhas rectas determinadas na classe C_2 ;
- 8) definir os pontos de intersecção, determinados no passo anterior, como os pontos de calibração pretendidos.

Na Fig. 21 está representado, de forma esquemática, o funcionamento deste método.

3 - Implementações desenvolvidas*

Nesta secção são apresentadas, de forma sucinta¹⁰, quatro implementações desenvolvidas no domínio da calibração de câmaras. Estas implementações foram desenvolvidas em linguagem C em ambiente UNIX numa estação de trabalho SunSparc. As implementações integram-se no ambiente de processamento de imagem XITE¹¹, do qual utilizam algumas funções e tipos de variáveis já previamente definidas. Assim, são apresentadas as seguintes implementações:

- ✓ Uma implementação para calibração do factor de incerteza horizontal s_x de uma câmara: *findsx*.
- ✓ Uma implementação para a calibração de todos os parâmetros intrínsecos e extrínsecos de uma câmara, exceptuando o factor de incerteza horizontal s_x : *calcamera*.
- ✓ Uma implementação para a determinação das coordenadas imagem na memória *frame* de pontos de calibração utilizados: *calpoint*.

* Esta secção foi introduzida por razões de completude, mas não é essencial à compreensão das secções seguintes.

¹⁰ Ver [Tavares, 1995], onde as implementações estão descritas de forma mais detalhada.

¹¹ X-based Image processing Tools an Environment, [Lønnestad, 1992].

- ✓ Uma implementação para simulação de uma câmara de geometria interna conhecida: *simcamera*.

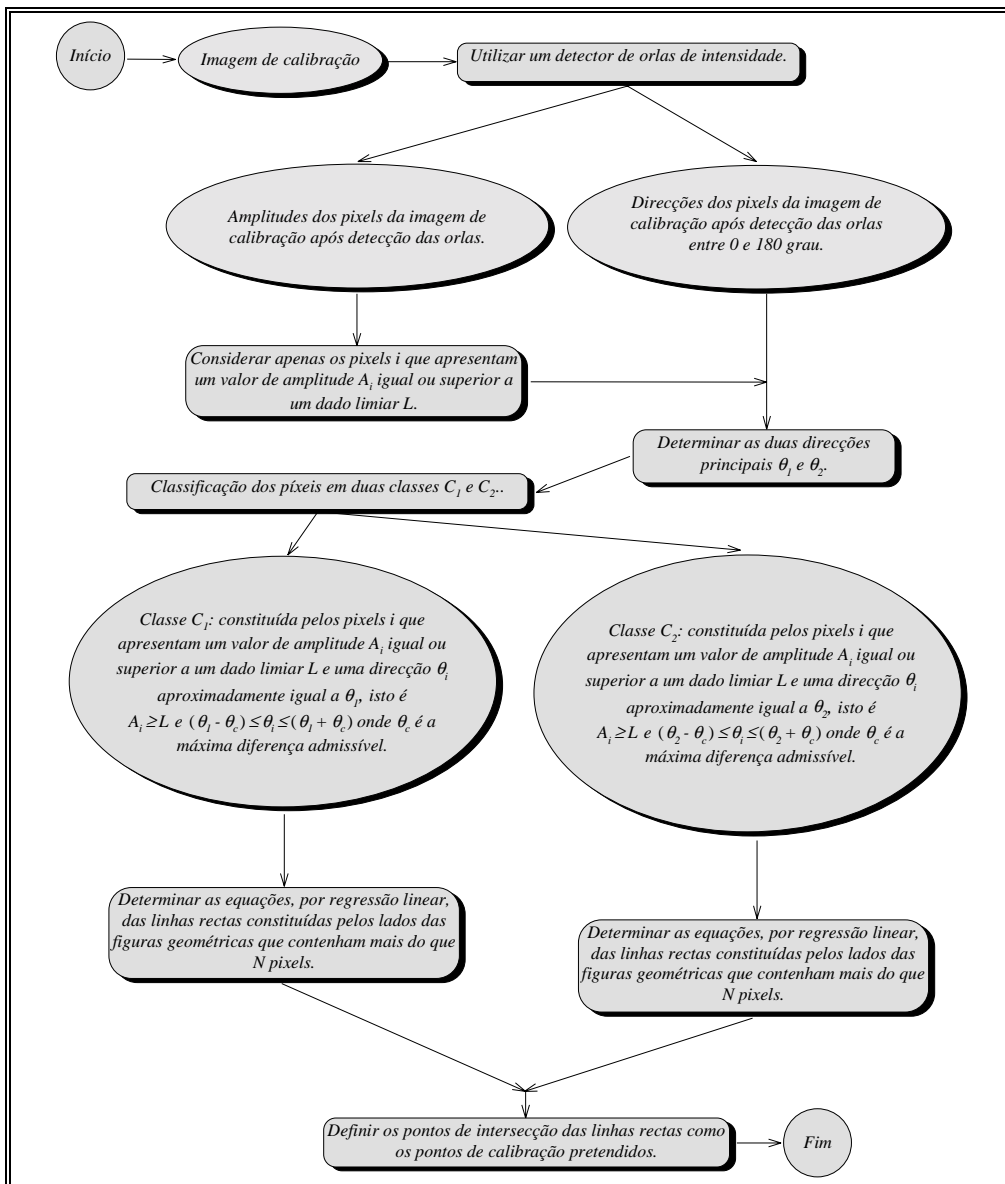


Fig. 21 - Esquema do método para determinação das coordenadas dos pontos de calibração na memória frame concebido e implementado no âmbito desta dissertação.

O desenvolvimento da implementação *simcamera* deveu-se à necessidade de se obter resultados simulados de forma a testar convenientemente a implementação *calcamera* sem a necessidade dos diferentes *setups* que, em testes reais, seriam inevitavelmente requeridos. Com esta implementação torna-se possível obter, de forma automática, as coordenadas 3D e as coordenadas na memória *frame* dos pontos de calibração coplanares, para uma dada geometria interna e posição e orientação da câmara relativamente ao plano de calibração.

3.1 - Calibração do factor de incerteza horizontal de uma câmara: *findsx*

Neste ponto é apresentada, de forma sumária, uma implementação para a calibração do factor de incerteza horizontal de uma câmara s_x , segundo o método descrito no ponto 1.6.1. A referida implementação é constituída pelos módulos *principal*, *dft* e *find*. Apresenta-se de seguida a descrição de cada um destes módulos.

a) **Módulo principal**, responsável pela interface implementação/utilizador. A seu cargo está:

- ✓ Leitura dos argumentos opcionais:
 - *filename*, nome do ficheiro para escrita dos resultados obtidos pela implementação nomeadamente, as condições da realização da calibração e o factor de incerteza horizontal s_x determinado. Por defeito é “NULL”, o que implica a não abertura do ficheiro.
 - *nlines*, número de linhas a considerar, centradas no centro da imagem, para obtenção de uma linha de valores médios. Por defeito, é igual a 19.
 - *ncolumns*, número de colunas, centradas no centro da imagem, a considerar. Por defeito, é igual a 19.
 - *nzeros*, número de zeros a acrescentar às amostras para interpolar a representação espectral antes de realizar a transformada discreta de Fourier. Por defeito, é igual a 60.
- ✓ Leitura dos argumentos passados à implementação pela linha de comando:
 - *Ncx/fc*, número de elementos sensores na direcção x (N_{cx}), ou frequência de relógio dos elementos sensores da câmara f_c .
 - *Nfx/ff*, número de elementos imagem numa linha amostrados pelo computador N_{fx} , ou frequência de amostragem do conversor analógico/digital f_p .
 - *input*, nome da imagem obtida sem o filtro passa-baixo entre a câmara e o conversor analógico/digital e com um objecto difuso colocado sobre a lente.
- ✓ Chamada do módulo **find**.
- ✓ Apresentação dos resultados obtidos.
- ✓ Comunicação da existência de erros em argumentos de entrada ou ao longo do decurso da execução.

Como restrições na utilização deste módulo têm-se:

- ⊗ o número de linhas centradas no centro da imagem a considerar para obtenção de uma linha de valores médios deve ser ímpar e menor que o tamanho da imagem segundo a direcção y ;
- ⊗ o número de colunas centradas no centro da imagem a considerar deve ser ímpar e menor que o tamanho da imagem segundo a direcção x ;
- ⊗ o número de elementos imagem numa linha amostrados pelo computador N_{fx} , ou a frequência de amostragem do conversor analógico/digital f_p , deve ser diferente de zero;
- ⊗ a imagem de entrada deve ser do tipo “UNS_BYTE” e constituída por apenas uma banda.

b) **Módulo dft**, responsável pela execução da transformada discreta de Fourier (DFT) de um conjunto de n amostras.

c) **Módulo find**, responsável pela calibração propriamente dita do factor de incerteza horizontal s_x de uma câmara, segundo o método descrito no ponto 1.6.1.

Na Fig. 22, estão representados os módulos que constituem esta implementação assim como a relação existente entre estes.

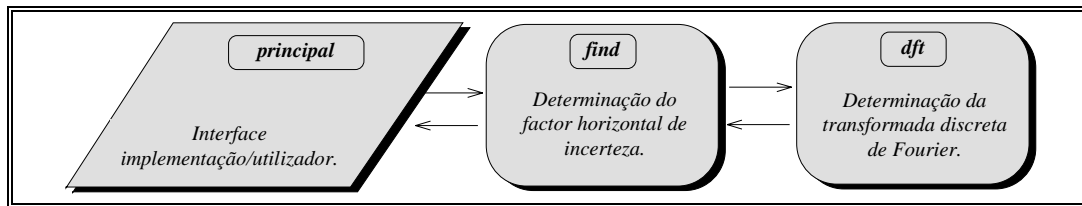


Fig. 22 - Módulos integrantes da implementação *findsx*.

3.2 - Calibração de uma dada câmara: *calcamera*

Neste ponto é apresentada, de forma sumária, uma implementação para calibração de uma dada câmara segundo o método descrito no ponto 1.6.3. Com esta implementação, é possível a calibração de todos os parâmetros de uma câmara excepto o factor de incerteza horizontal s_x . A referida implementação é constituída pelos módulos *principal*, *linleastsquar*, *solve* e *findcenter*. Apresenta-se de seguida a descrição de cada um destes módulos.

a) **Módulo *principal***, é responsável pela interface implementação/utilizador e pela calibração de todos os parâmetros de uma câmara, excepto o factor de incerteza horizontal s_x e as coordenadas do centro da imagem na memória *frame* (C_x, C_y). A seu cargo está:

✓ Leitura dos argumentos opcionais:

- *outfile*, nome do ficheiro para escrita dos resultados obtidos, nomeadamente, as condições da realização da calibração e os parâmetros obtidos. Por defeito é “NULL”, o que implica a não abertura do ficheiro.
- *nlstop*, tolerância de erro para as soluções exactas da distância focal efectiva f , da componente t_z do vector de translação T e do factor de distorção radial da lente k_1 , determinadas pela resolução do sistema de equações não linear respectivo e utilizando o método Levenberg-Marquardt. Por defeito, é igual a 0.0000001.
- *sx*, factor de incerteza horizontal s_x . Por defeito, é igual a 0.710935.
- *findcenter*; quando *findcenter* é especificado pelo utilizador como igual a 1, a implementação determina as coordenadas exactas do centro da imagem na memória *frame* (C_x, C_y). Por defeito, esta determinação não é realizada.

✓ Leitura dos argumentos passados à implementação pela linha de comando:

- *dx*, distância entre centros dos elementos sensores vizinhos na direcção x , d_x .
- *dy*, distância entre centros dos sensores CCD vizinhos na direcção y , d_y .
- *Cx*, coordenada do centro da imagem na memória *frame* segundo a direcção x , C_x . Quando C_x não é conhecido exactamente, o utilizador deve especificar C_x igual a metade da dimensão da memória *frame* segundo a direcção x e especificar à implementação que pretende que esta determine a coordenada correcta. Deste modo, o valor especificado para C_x é apenas a solução inicial sendo a implementação responsável pela determinação da solução exacta.
- *Cy*, coordenada do centro da imagem na memória *frame* segundo a direcção y , C_y . Quando C_y não é conhecido exactamente, o utilizador deve especificar C_y igual a metade da dimensão da memória *frame* segundo a direcção y e especificar à implementação que pretende que esta determine a coordenada correcta. Deste modo, o valor especificado para C_y é apenas a solução inicial sendo a implementação responsável pela determinação da solução exacta.

- *infilename*, nome do ficheiro de entrada que contém as coordenadas 3D mundo $(x_{w_i}, y_{w_i}, z_{w_i})$ e as coordenadas 2D imagem na memória *frame* (X_f, Y_f) para todos os pontos de calibração *i*.
- ✓ Determinação das coordenadas do centro da imagem na memória *frame* (C_x, C_y) quando pretendido pelo utilizador; para tal, este módulo utiliza o módulo *findcenter*.
- ✓ Calibração de todos os restantes parâmetros da câmara, excepto o factor de incerteza horizontal s_x ; para tal, este módulo utiliza os módulos *linleastsquar* e *solve*.
- ✓ Apresentação dos resultados obtidos.
- ✓ Comunicação da existência de erros em argumentos de entrada ou ao longo do decurso da execução.

Como restrições na utilização deste módulo têm-se:

- ⊗ o número de pontos de calibração deve ser maior do que cinco;
- ⊗ o sistema de coordenadas 3D mundo deve ser escolhido de forma que todos os pontos de calibração tenham a coordenada 3D mundo z igual a zero e a componente t_y do vector de translação T não seja exactamente igual a zero;
- ⊗ o factor de incerteza horizontal s_x deve ser diferente de zero.

- b) **Módulo *linleastsquar***, responsável pela resolução de um sistema sobredeterminado de equações lineares, $z [m] [n] \times a [n] = y [m]$. O método de resolução utilizado é o método dos mínimos quadrados, com resolução do respectivo sistema de equações normais. Para tal, utiliza-se a decomposição de Cholesky¹² da matriz $[[m] [n] \quad [z] [n]]$, com posterior substituição para a frente e para trás. Esta decomposição é das mais eficientes em termos de memória consumida; no entanto, a sua utilização só é possível em matrizes $[[m] [n] \quad [z] [n]]$ definidas positivamente.
- c) **Módulo *solve***, responsável pela resolução de um sistema de equações lineares $alfa [NMAXI] [NMAXI] \times delta [NMAXI] = beta [NMAXI]$. O método de resolução utilizado é a decomposição de Cholesky da matriz $alfa [NMAXI] [NMAXI]$ com posterior substituição para a frente e para trás.
- d) **Módulo *findcenter***, responsável pela determinação das coordenadas do centro da imagem na memória *frame* (C_x, C_y) , segundo o método iterativo apresentado no ponto 1.6.2.2. Durante a determinação, todos os restantes parâmetros de calibração são também determinados novamente até o módulo atingir uma solução global óptima.

Na Fig. 23 estão representados os módulos que constituem esta implementação, assim como a relação existente entre estes.

¹² Ver, por exemplo [Chapra, 1988].

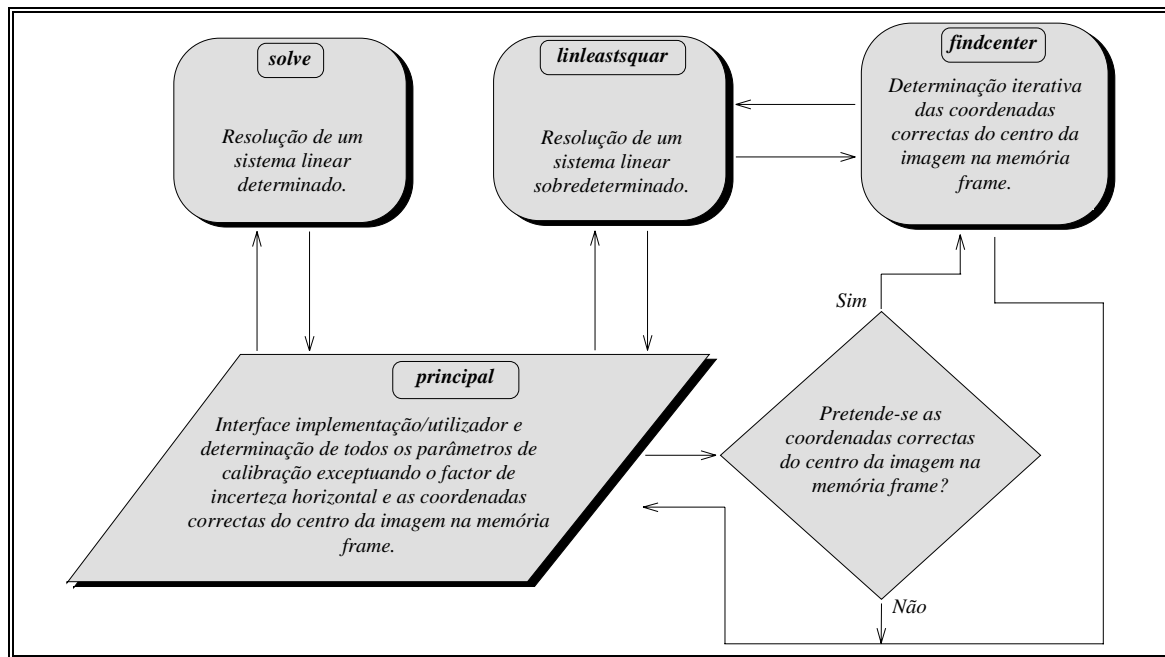


Fig. 23 - Módulos integrantes da implementação calcamera.

3.3 - Determinação das coordenadas na memória frame dos pontos de calibração: calpoint

Neste ponto é apresentada, de forma sumária, a implementação do método descrito no ponto 2.2 para determinação das coordenadas imagem na memória frame (X_f, Y_f) dos pontos de calibração considerados. Com esta implementação obtêm-se:

- uma imagem de saída com três bandas: a primeira, contém todos os *pixels* identificados com amplitude superior a um dado limiar especificado pelo utilizador e direcção aproximadamente igual a uma direcção principal; a segunda, contém todos os *pixels* identificados com amplitude superior ao mesmo limiar e direcção aproximadamente igual à outra direcção principal; na terceira, são identificados os pontos de calibração e assinalados por intermédio de uma cruz;
- um ficheiro com a escrita dos resultados obtidos, nomeadamente as coordenadas imagem na memória frame (X_f, Y_f) dos pontos de calibração determinados.

A referida implementação é constituída pelos módulos: *principal*, *lregression*, *mark*, *findlines* e *findpoints*. Apresenta-se de seguida a descrição de cada um destes módulos.

a) **Módulo principal**, responsável pela interface implementação/utilizador. A seu cargo está:

✓ Leitura dos argumentos opcionais:

- *title*, título da imagem de saída. Por defeito, é igual a “Calibration points”.
- *erthresang*, tolerância de erro na classificação de um dado *pixel* quanto à sua direcção. Por defeito, é igual a 2.
- *difangmax*, diferença mínima entre as duas direcções principais. Por defeito, é igual a 32.
- *maxdist*, máxima distância de um dado *pixel* a uma determinada recta para ser classificado como pertencente a esta. Por defeito, é igual a 10.0.
- *thresamp*, valor mínimo de amplitude que um dado *pixel* deverá apresentar para não ser classificado como ruído. Por defeito, é igual a 50.

✓ Leitura dos argumentos passados à implementação pela linha de comando:

- *inimage*, nome da imagem de entrada.
 - *outimage*, nome da imagem de saída.
 - *outfile*, nome do ficheiro para escrita dos resultados obtidos, nomeadamente as coordenadas imagem na memória *frame* (X_f, Y_f) dos pontos de calibração determinados.
- ✓ Chamada do módulo *findpoints*.
 - ✓ Apresentação do número de pontos de calibração determinados.
 - ✓ Escrita em disco da imagem de saída com a respectiva descrição das condições de obtenção.
 - ✓ Comunicação da existência de erros em argumentos de entrada ou ao longo do decurso da execução.

Como restrições na utilização deste módulo têm-se:

- ⊗ a imagem de entrada deve ser do tipo “*UNS_BYTE*”;
 - ⊗ a imagem de entrada deverá conter figuras geométricas de quatro lados paralelos dois a dois, perfeitamente iguais e paralelas entre si;
 - ⊗ a imagem de entrada deverá ser constituída por duas bandas de igual tamanho, resultantes de um detector de orlas de intensidade, por exemplo o detector de Deriche¹³. Este detector deverá classificar cada *pixel* na primeira banda quanto à amplitude, e na segunda banda quanto à direcção (entre 0 e 180 grau).
- b) **Módulo *lregression***, responsável pela determinação da equação de uma recta que melhor se ajusta a um dado conjunto de pontos. Esta determinação é realizada utilizando a regressão linear pelo método dos mínimos quadrados¹⁴.
- c) **Módulo *mark***, responsável pelo desenho, numa dada banda de entrada, de uma cruz de dimensões 9×9 centrada num determinado *pixel*, cujas coordenadas são passadas ao módulo.
- d) **Módulo *findlines***, responsável pela determinação das equações das rectas que são definidos por um conjunto de *pixels*, numa dada banda de entrada, que têm mais ou menos a mesma direcção. Durante a execução, o módulo opta automaticamente pela melhor regressão linear, isto é, para rectas com declives entre -45° e 45° a regressão linear é do tipo $y = mx + b$ e, para os restantes declives, a regressão linear é do tipo $x = m'y + b'$. Para determinar se um *pixel* pertence a uma dada recta, a sua distância a esta é calculada e comparada com a máxima distância admissível especificada pelo utilizador. Para eliminação de falsas linhas, o módulo só considera uma dada recta se o número de *pixels* que esta contém é maior do que um determinado valor.
- e) **Módulo *findpoints***, responsável pela:
- ✓ determinação das duas direcções principais presentes na imagem de entrada;
 - ✓ classificação dos *pixels*, na primeira banda da imagem de saída, que têm amplitude superior a um dado limiar e que têm uma direcção próxima de uma das direcções principais;
 - ✓ classificação dos *pixels*, na segunda banda da imagem de saída, que têm amplitude superior a um dado limiar e que têm uma direcção próxima da outra direcção principal;
 - ✓ determinação das rectas que existem em cada uma destas bandas; para tal utiliza o módulo *findlines* descrito anteriormente;

¹³ Este detector é apresentado de forma sucinta no próximo capítulo desta dissertação.

¹⁴ Ver, por exemplo [Chapra, 1988].

- ✓ determinação das coordenadas dos pontos de calibração através da intersecção entre as várias rectas definidas analiticamente;
- ✓ registo dos pontos de calibração na terceira banda de saída; para tal utiliza o módulo *mark* descrito anteriormente.

Na Fig. 24 estão representados os módulos que constituem esta implementação, assim como a relação existente entre estes.

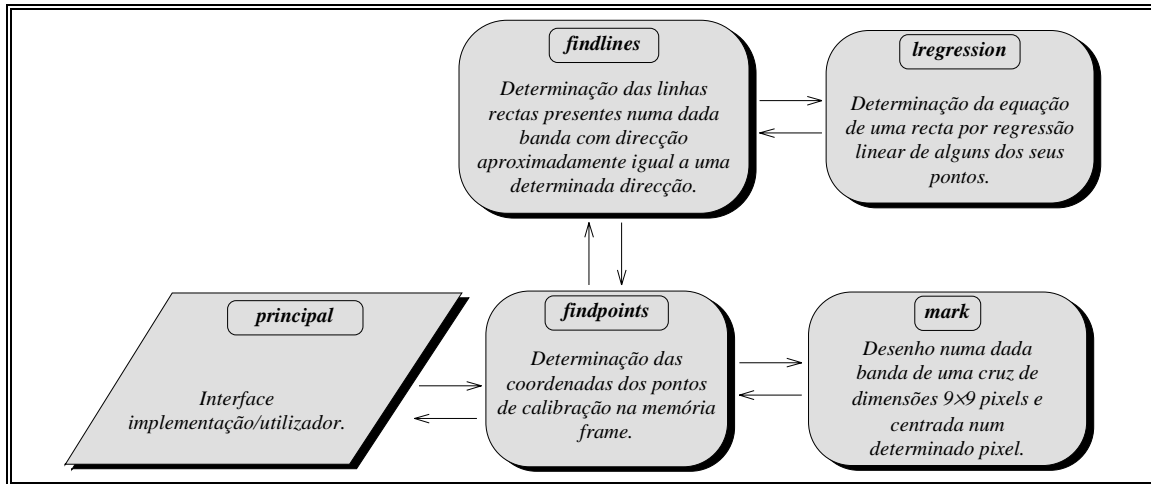


Fig. 24 - Módulos integrantes da implementação *calpoint*.

3.4 - Simulador de uma dada câmara: *simcamera*

Neste ponto é apresentada uma implementação de um simulador de uma dada câmara para um dado conjunto de pontos coplanares. A partir dos respectivos parâmetros de entrada, a própria implementação determina automaticamente as coordenadas 3D mundo (x_w, y_w, z_w) e as coordenadas na memória *frame* (X_f, Y_f), para cada ponto de calibração, do seguinte modo:

- I) Determinação da projecção inversa, sobre o plano de calibração considerado, dos quatro vértices que definem as dimensões da memória *frame*. Esta projecção inversa é possível pois conhece-se a geometria interna da câmara, a posição e orientação desta relativamente ao plano de calibração considerado, a coordenada 3D mundo z do plano de calibração e as coordenadas 2D na memória *frame* dos pontos pretendidos - $(1, 1)$, $(1, Dim_{fy})$, $(Dim_{fx}, 1)$ e (Dim_{fx}, Dim_{fy}) onde Dim_{fx} é a dimensão da memória *frame* segundo a direcção x e Dim_{fy} é a dimensão segundo a direcção y .
- II) Considerar o paralelogramo definido pelas projecções sobre o plano de calibração definidas anteriormente. Este paralelogramo tem lados mais ou menos paralelos dois a dois.
- III) Definir o conjunto de n - definido como igual à raiz quadrada do número de pontos de calibração pretendidos - linhas rectas distribuídas uniformemente entre dois lados mais ou menos paralelos.
- IV) Definir o conjunto de n linhas rectas distribuídas uniformemente entre os restantes dois lados do paralelogramo.
- V) Definir os pontos de calibração como situados sobre o plano de calibração e coincidentes com as intersecções, sobre este plano, das linhas rectas definidas em III) com as linhas rectas definidas em IV).
- VI) Determinar para cada ponto de calibração, cujas coordenadas 3D foram determinadas anteriormente, as coordenadas respectivas na memória *frame*.

Na Fig. 25 está representada, de forma esquemática, a determinação das coordenadas 3D mundo e 2D na memória *frame* dos pontos de calibração pretendidos, realizada por esta implementação.

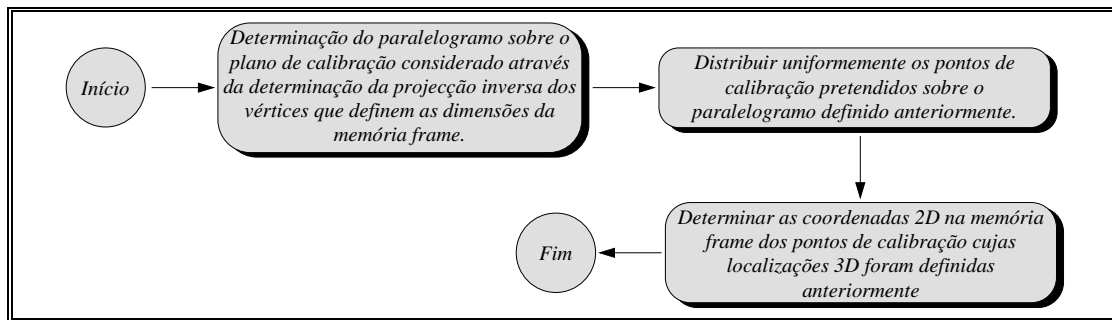


Fig. 25 - Esquema da determinação automática das coordenadas 3D mundo e 2D na memória *frame* dos pontos de calibração pretendidos.

Esta implementação é constituída unicamente pelo módulo **principal**. Este módulo é então responsável por:

- ✓ Leitura dos argumentos opcionais:
 - *npoint*, número de pontos de calibração pretendido. Por defeito, é igual a 36.
 - *zwplane*, coordenada 3D mundo *z* para o plano de calibração. Por defeito, é igual a 0.0.
 - *xcenterbuf*, coordenada C_x do centro da imagem na memória *frame*, segundo a direcção *x*. Por defeito, é igual a metade da dimensão da referida memória segundo a mesma direcção.
 - *ycenterbuf*, coordenada C_y do centro da imagem na memória *frame*, segundo a direcção *y*. Por defeito, é igual a metade da dimensão da referida memória segundo a mesma direcção.
- ✓ Leitura dos argumentos passados à implementação pela linha de comando:
 - *xsizebuf*, dimensão da memória *frame* segundo a direcção *x*.
 - *ysizebuf*, dimensão da memória *frame* segundo a direcção *y*.
 - *rotx*, valor em grau da rotação segundo o eixo *x*, $R_{(x, \phi)}$.
 - *roty*, valor em grau da rotação segundo o eixo *y*, $R_{(y, \theta)}$.
 - *rotz*, valor em grau da rotação segundo o eixo *z*, $R_{(z, \psi)}$.
 - *tx*, componente t_x do vector de translação, isto é a translação segundo o eixo *x*.
 - *ty*, componente t_y do vector de translação, isto é a translação segundo o eixo *y*.
 - *tz*, componente t_z do vector de translação, isto é a translação segundo o eixo *z*.
 - *f*, distância focal efectiva *f*.
 - *dx*, distância entre centros dos elementos sensores vizinhos na direcção *x*, d_x .
 - *dy*, distância entre centros dos sensores CCD vizinhos na direcção *y*, d_y .
 - *sx*, factor de incerteza horizontal s_x .
 - *k1*, factor de distorção radial da lente k_1 .

- *filename*, nome do ficheiro para escrita dos resultados obtidos, nomeadamente as coordenadas 3D mundo (x_w, y_w, z_w) e as coordenadas 2D imagem na memória *frame* (X_f, Y_f) determinadas para cada ponto de calibração.
- ✓ Apresentação das componentes determinadas da matriz de rotação 3D R do sistema de coordenadas mundo para o sistema câmara.
- ✓ Execução da simulação da câmara propriamente dita.
- ✓ Escrita no ficheiro de saída das coordenadas 3D mundo (x_w, y_w, z_w) e das coordenadas 2D imagem na memória *frame* (X_f, Y_f) determinadas para todos os pontos de calibração.
- ✓ Comunicação da existência de erros em argumentos de entrada ou ao longo do decurso da execução.

Como restrições na utilização deste módulo têm-se:

- ☒ o valor especificado para o número de pontos de calibração pretendidos deve ter raiz quadrada inteira, por exemplo 16, 36, 49, etc.;
- ☒ os valores especificados para as rotações segundo os três eixos principais deverão estar definidos em grau e compreendidos entre -360 e 360.

4 - Alguns resultados experimentais obtidos pelas implementações desenvolvidas

Nesta secção são apresentados alguns resultados experimentais¹⁵ obtidos pelas várias implementações descritas de forma sumária na última secção. Assim serão apresentados:

- a) resultados experimentais obtidos na utilização da implementação *calpoint* para a determinação das coordenadas na memória *frame* dos pontos de calibração considerados;
- b) resultados experimentais obtidos na utilização da implementação *simcamera* para a simulação de câmaras de geometria interna conhecida;
- c) resultados experimentais obtidos pela implementação *calcamera* para a calibração de câmaras.

Infelizmente, até esta data não é possível apresentar resultados experimentais obtidos pela implementação *findsx*, pois ainda não existem condições práticas para tal, nomeadamente não foi possível, para a câmara disponível, desligar o filtro passa-baixo.

4.1 - Utilizando a implementação *calpoint* para a determinação das coordenadas na memória *frame* dos pontos de calibração considerados

Com o objectivo de testar a implementação *calpoint*, para a determinação das coordenadas 2D na memória *frame* (X_f, Y_f) dos pontos de calibração considerados, foi criada uma imagem de 8 bits com dimensões 611×885 pixels e contendo 24 quadrados. A referida imagem¹⁶ está representada na Fig. 26 e, como se pode observar facilmente da mesma, contém 96 possíveis pontos de calibração.

Com o objectivo de utilizar a implementação *calpoint* (descrita no ponto 3.3) na imagem de teste, submeteu-se esta previamente ao detector de orlas de intensidade de Deriche. As bandas que constituem a imagem obtida, como resultado deste detector, estão representadas nas Figs. 27 e 28.

¹⁵ No capítulo V são apresentados outros resultados obtidos pelas implementações *calpoint* e *calcamera*.

¹⁶ Todas as imagens apresentadas, excepto zonas ampliadas, sofreram um escalonamento de 70%.

Note-se que as bandas representadas foram previamente negadas (com o objectivo de a sua visualização ser mais clara).

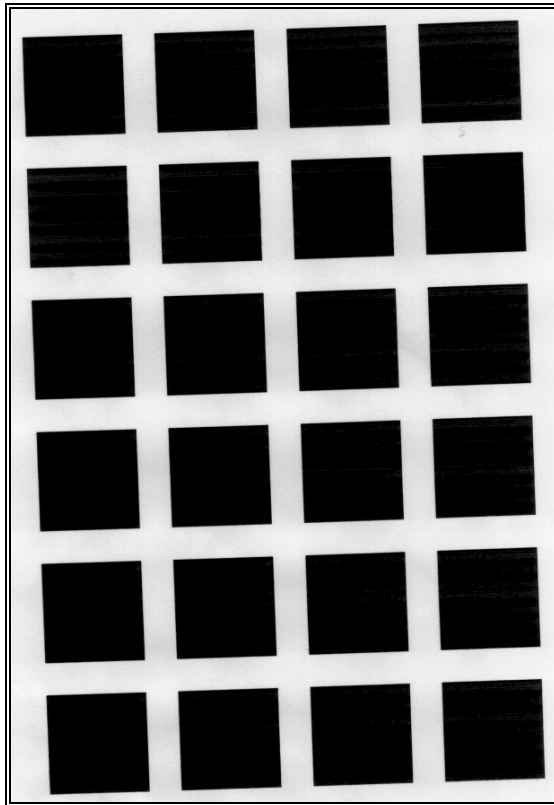


Fig. 26 - Imagem criada para teste da implementação *calpoint*.

Aplicou-se então a implementação *calpoint* a esta imagem, constituída pelas duas bandas representadas nas *Figs. 27 e 28*. Como parâmetros de entrada para a implementação *calpoint*, utilizaram-se os respectivos valores por defeito. As três bandas que se obtiveram como resultado estão representadas nas *Figs. 29, 30 e 31*. Assim, na *Fig. 29* está representada a primeira banda da imagem de saída da implementação *calpoint*, que apresenta todos os *pixels* que têm um valor de amplitude superior a 50 e direcção aproximadamente igual a uma das direcções principais. Na *Fig. 30* está representada a segunda banda da imagem de saída da implementação *calpoint*. Esta banda apresenta todos os *pixels* que têm um valor de amplitude superior a 50 e direcção aproximadamente igual à segunda das direcções principais. Na *Fig. 31* está representada a terceira banda da imagem de saída da implementação *calpoint*, onde estão assinaladas por intermédio de cruces as localizações dos pontos de calibração existentes na imagem de teste, isto é, os vértices dos quadrados.

Com o objectivo de determinar a qualidade dos resultados obtidos, executou-se a soma lógica da imagem de teste, representada na *Fig. 26*, com a terceira banda da imagem de saída da implementação *calpoint*, representada na *Fig. 31*. O resultado é apresentado na *Fig. 32*.

Com o mesmo objectivo, procedeu-se também à soma lógica da primeira banda da imagem obtida pelo detector de Deriche, representada na *Fig. 27*, com a terceira banda da imagem de saída da implementação *calpoint*, mostrando-se o resultado na *Fig. 33*.

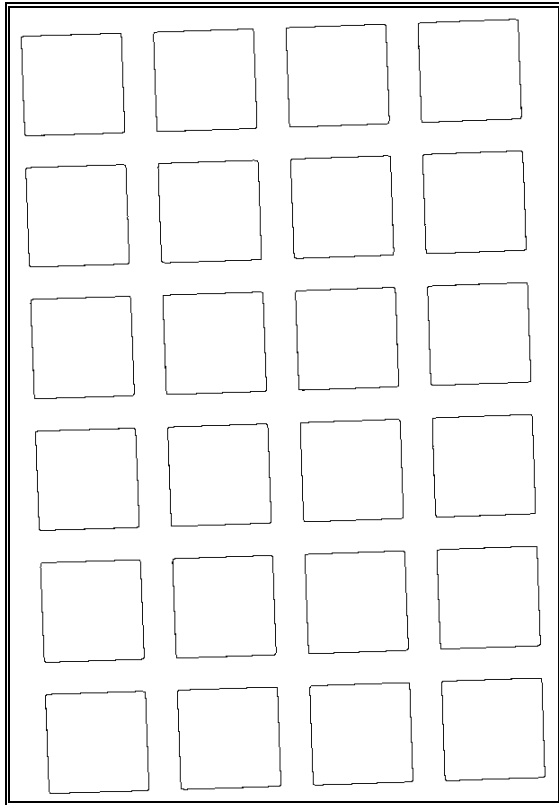


Fig. 27 - Banda de amplitudes, previamente negada, resultante da aplicação do detector de Deriche na imagem de teste.

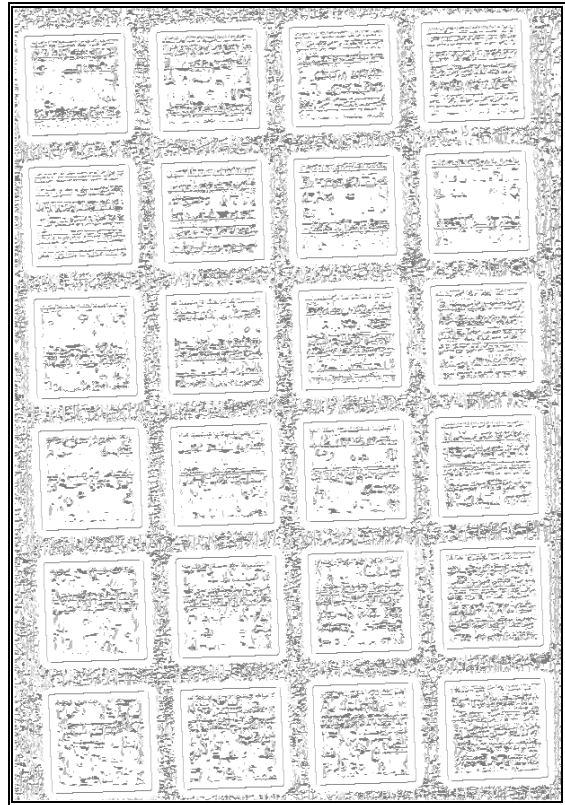


Fig. 28 - Banda de direcções entre 0 e 180 grau, previamente negada, resultante da aplicação do detector de Deriche na imagem de teste.

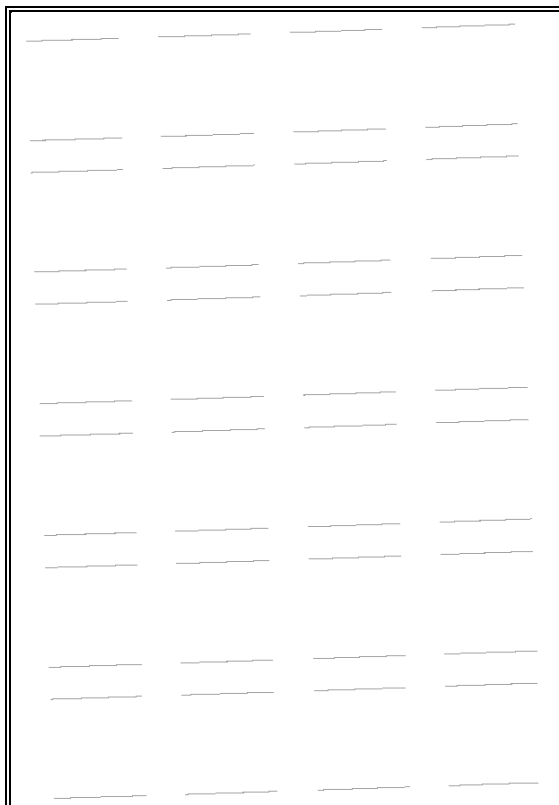


Fig. 29 - Primeira banda da imagem de saída da implementação *calpoint*.

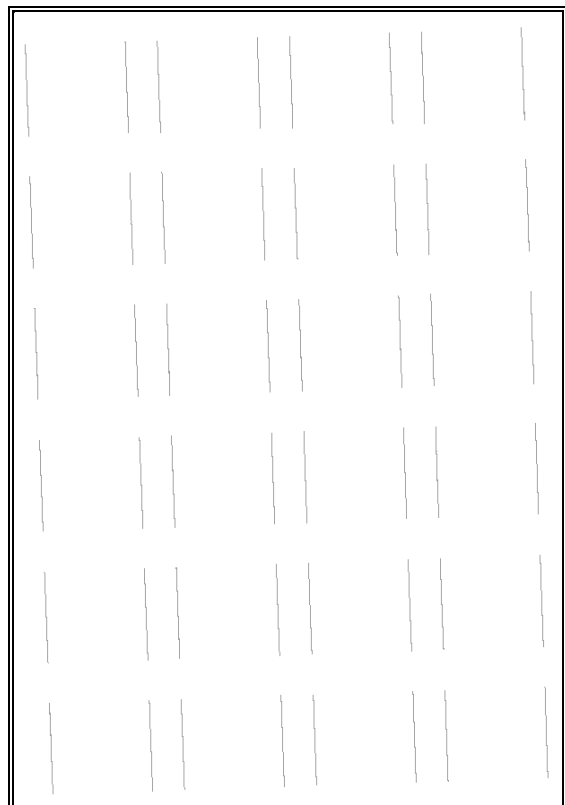


Fig. 30 - Segunda banda da imagem de saída da implementação *calpoint*.

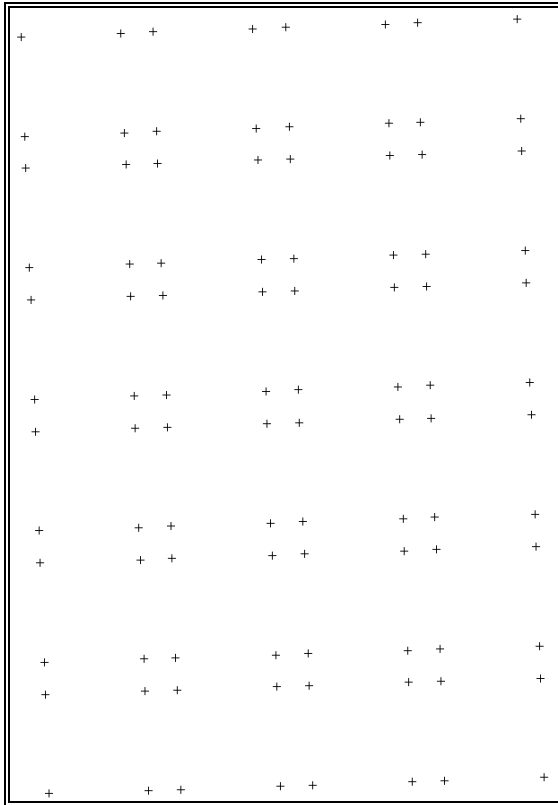


Fig. 31 - Terceira banda da imagem de saída da implementação *calpoint*.

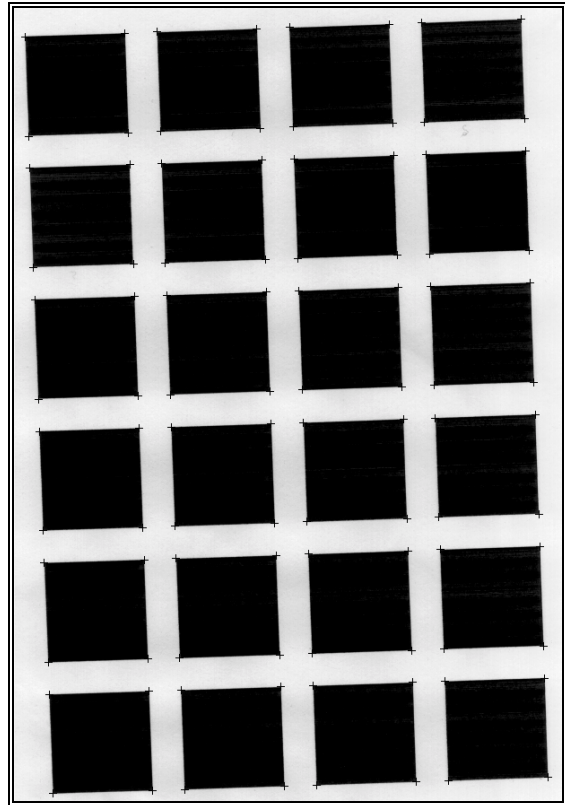


Fig. 32 - Resultado da soma lógica entre a imagem de teste e a terceira banda da imagem de saída da implementação *calpoint*.

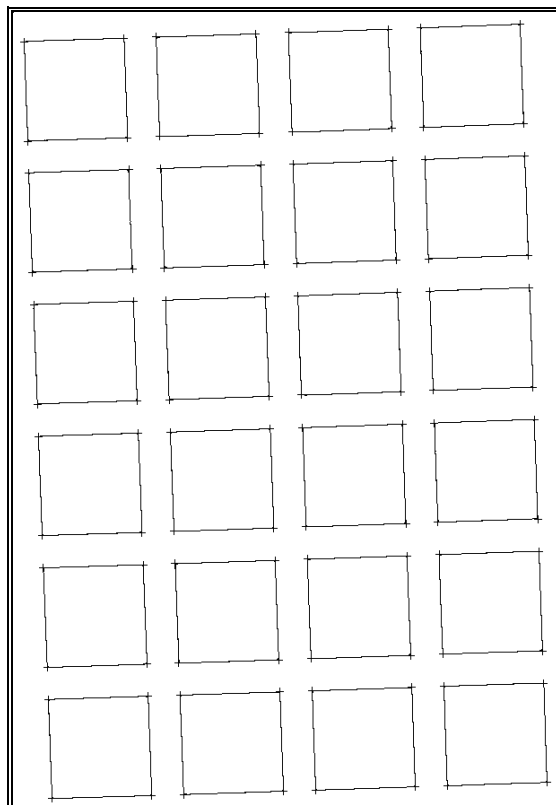


Fig. 33 - Resultado da soma lógica da primeira banda da imagem obtida pelo detector de Deriche, previamente negada, com a terceira banda da imagem resultante da implementação *calpoint*.

Como é possível observar-se a partir das *Figs. 32 e 33*, a implementação *calpoint* apresenta resultados de boa qualidade com resolução ao nível do *subpixel*. Esta boa qualidade é mais uma vez verificada nas *Fig. 34 e 35*, onde são apresentadas zonas ampliadas, escolhidas aleatoriamente, das *Figs. 32 e 33*. Assim, esta implementação é uma boa solução para o problema da precisão requerida na determinação das coordenadas 2D na memória *frame* (X_f, Y_f) dos pontos a serem utilizados na calibração de câmaras.

A título de exemplo, na *Fig. 36* está representado o conteúdo do ficheiro de saída da implementação *calpoint*, nomeadamente as coordenadas 2D na memória *frame* (X_f, Y_f) de todos os pontos de calibração determinados na imagem de entrada apresentada na *Fig. 26*.

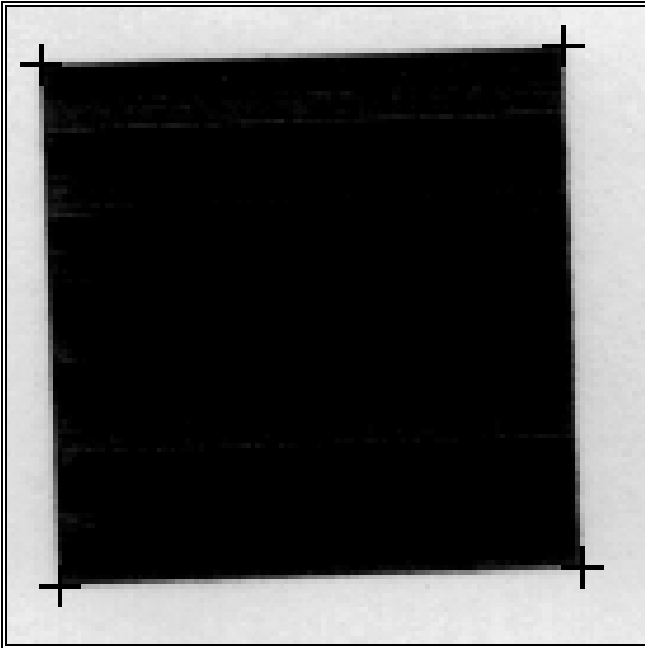


Fig. 34 - Ampliação de uma zona escolhida aleatoriamente da imagem representada na Fig. 32.

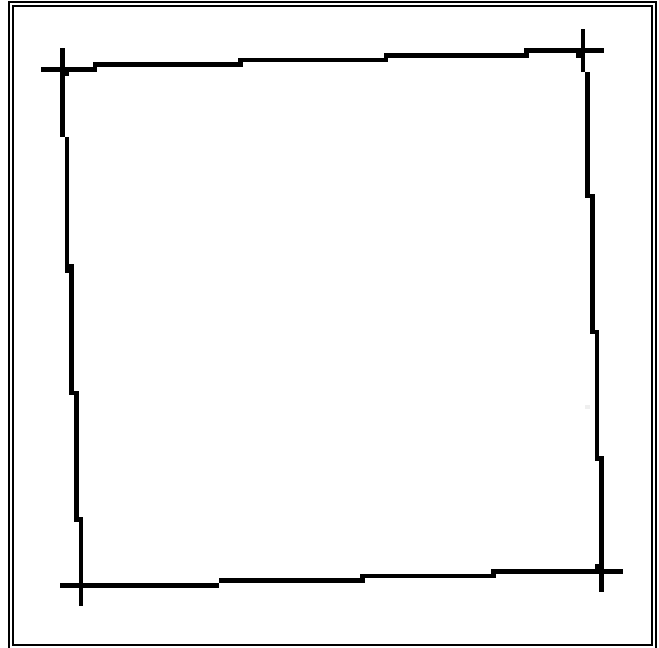


Fig. 35 - Ampliação de uma zona escolhida aleatoriamente da imagem representada na Fig. 33.

4.2 - Utilizando a implementação *simcamera* para a simulação de câmaras de geometria interna conhecida

Neste ponto, são apresentados os resultados de algumas simulações de câmaras de geometria interna conhecida utilizando a implementação *simcamera*, apresentada no ponto 3.4.

Na primeira simulação efectuada, foram utilizados os parâmetros de entrada para a implementação *simcamera* apresentados na *Tab. I*.

<i>filename</i>	<i>npoint</i>	<i>zwplane</i>	<i>xcenterbuf</i>	<i>ycenterbuf</i>	<i>xsizebuf</i>
out1.txt	16	0.0	256	256	512
<i>ysizebuf</i>	<i>rotx</i>	<i>roty</i>	<i>rotz</i>	<i>tx</i>	<i>ty</i>
512	-30.0	-30.0	-30.0	100.0	100.0
<i>tz</i>	<i>f</i>	<i>dx</i>	<i>dy</i>	<i>sx</i>	<i>k1</i>
2000.0	50.0	8,37765957e-3	8.075601357e-3	0.710935	0.01

Tab. I - Parâmetros de entrada para a implementação *simcamera* utilizados na primeira simulação apresentada.

11.7411	33.5974				
417.674	19.5356	123.116	29.7393	158.908	28.4994	270.382	24.6379	306.503	23.3866
162.963	139.449	453.511	18.2942	564.927	14.4347	15.8474	144.546	127.205	140.688
568.892	125.384	274.415	135.587	310.51	134.336	421.675	130.485	457.486	129.244
311.794	169.891	17.1567	179.92	128.512	176.13	164.259	174.913	275.707	171.12
132.596	286.943	422.96	166.108	458.763	164.889	570.169	161.097	21.2585	290.748
462.731	275.661	168.309	285.723	279.734	281.915	315.796	280.682	426.955	276.883
281.042	317.901	574.127	271.854	22.5894	326.708	133.921	322.914	169.623	321.698
26.6904	437.514	317.096	316.672	428.253	312.884	464.02	311.665	575.414	307.869
432.25	423.712	138.005	433.725	173.673	432.511	285.071	428.72	321.099	427.494
174.994	468.655	467.99	422.495	579.375	418.705	28.0283	473.662	139.337	469.869
580.667	454.833	286.384	464.86	322.404	463.632	433.554	459.846	469.285	458.628
326.403	574.337	32.126	584.378	143.418	580.581	179.04	579.366	290.408	575.565
144.744	616.578	437.546	570.545	473.25	569.327	584.623	565.526	33.457	620.341
474.543	605.424	180.356	615.373	291.718	611.607	327.705	610.39	438.848	606.632
295.751	722.539	585.914	601.658	37.562	731.255	148.832	727.498	184.41	726.298
38.8798	766.861	331.712	721.325	442.849	717.574	478.517	716.37	589.88	712.611
444.141	753.397	150.147	763.164	185.714	761.983	297.05	758.284	333.004	757.089
189.758	872.633	479.801	752.212	591.163	748.513	42.9725	877.443	154.224	873.797
595.124	859.35	301.074	868.985	337.003	867.808	448.136	864.166	483.77	862.998
...

Fig. 36 - Conteúdo do ficheiro de resultados obtidos pela implementação *calpoint* para a imagem de teste representada na Fig. 26.

A matriz de rotação 3D resultante desta primeira simulação é $R = \begin{bmatrix} 0.75 & -0.433013 & 0.5 \\ 0.649519 & 0.625 & -0.433013 \\ -0.125 & 0.649519 & 0.75 \end{bmatrix}$.

A distribuição dos pontos de calibração, considerados na primeira simulação, no plano de calibração e na memória *frame* está representada na Fig. 37.

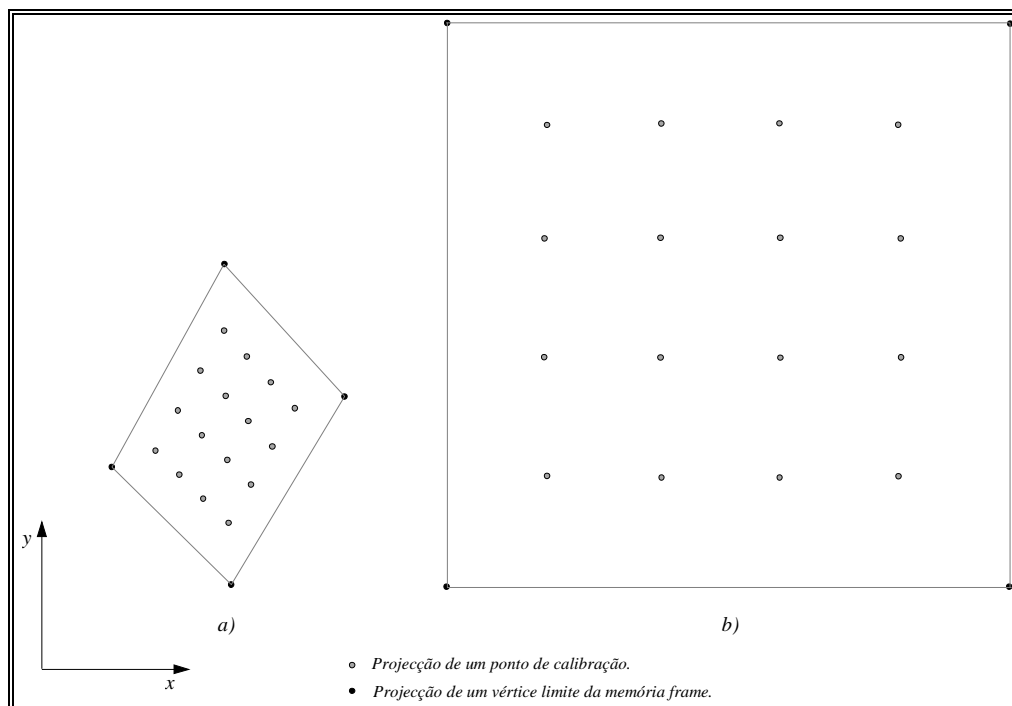


Fig. 37 - Distribuição dos pontos de calibração considerados na primeira simulação apresentada: a) no plano de calibração, b) na memória *frame*.

Como se observa da Fig. 37, a distribuição dos pontos de calibração é regular tanto na memória *frame* como no plano de calibração. No caso da memória *frame*, esta regularidade deve-se ao baixo valor da distorção radial da lente k_1 . Para o caso do plano de calibração, a regularidade deve-se à filosofia da implementação *simcamera*, que consiste em distribuir os pontos de calibração regularmente sobre o referido plano. Neste plano, a forma do conjunto dos pontos de calibração deve-se aos valores utilizados para as rotações segundo os três eixos principais.

O objectivo da segunda simulação, é demonstrar a influência do valor da distorção radial da lente k_1 . Deste modo, os parâmetros de entrada utilizados para a implementação *simcamera* foram os apresentados na Tab. II.

<i>filename</i>	<i>npoint</i>	<i>zwplane</i>	<i>xcenterbuf</i>	<i>ycenterbuf</i>	<i>xsizebuf</i>
out2.txt	16	0.0	256	256	512
<i>ysizebuf</i>	<i>rotx</i>	<i>roty</i>	<i>rotz</i>	<i>tx</i>	<i>ty</i>
512	-30.0	-30.0	-30.0	100.0	100.0
<i>tz</i>	<i>f</i>	<i>dx</i>	<i>dy</i>	<i>sx</i>	<i>k1</i>
2000.0	50.0	8,37765957e-3	8.075601357e-3	0.710935	0.1

Tab. II - Parâmetros de entrada para a implementação *simcamera* utilizados na segunda simulação apresentada.

A matriz de rotação 3D resultante desta segunda simulação é $R = \begin{bmatrix} 0.75 & -0.433013 & 0.5 \\ 0.649519 & 0.625 & -0.433013 \\ -0.125 & 0.649519 & 0.75 \end{bmatrix}$.

A distribuição dos pontos de calibração considerados, na segunda simulação, no plano de calibração e na memória *frame*, está representada na Fig. 38.

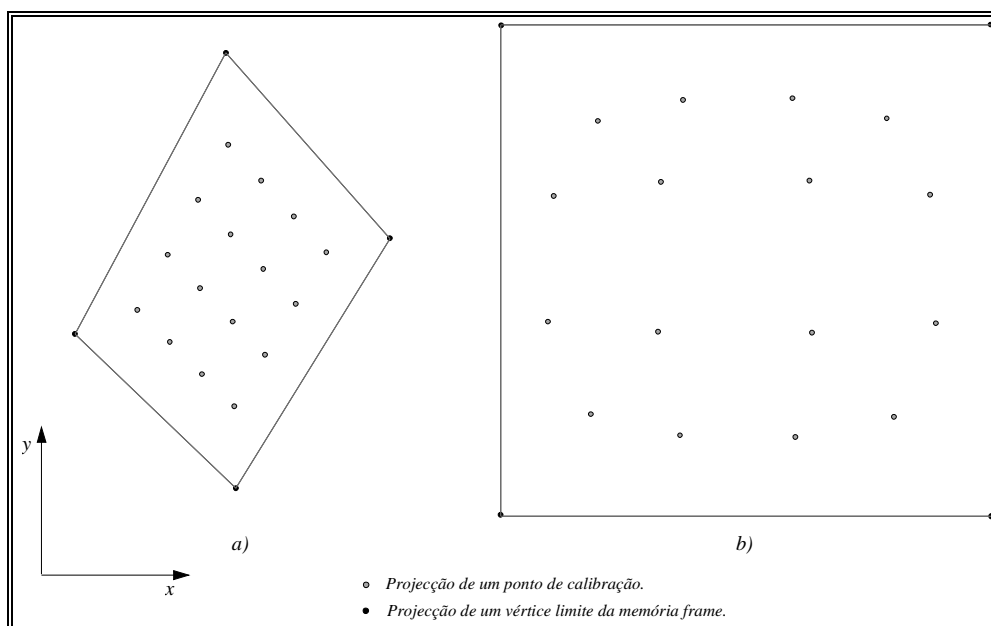


Fig. 38 - Distribuição dos pontos de calibração para a segunda simulação apresentada: a) no plano de calibração, b) na memória *frame*.

Como se observa da Fig. 38, a distribuição dos pontos de calibração já não é regular na memória *frame* mas apenas no plano de calibração. No caso da memória *frame*, esta não regularidade deve-se ao facto de o valor da distorção radial da lente k_1 ser elevado. Para o caso do plano de calibração, a regularidade deve-se novamente à filosofia da implementação *simcamera*.

Com o mesmo objectivo da simulação anterior, demonstrar a influência do valor da distorção radial da lente k_1 , foi executada a terceira e última simulação apresentada. Deste modo, os parâmetros de entrada utilizados para a implementação *simcamera* foram os apresentados na *Tab. III*.

A matriz de rotação 3D resultante desta terceira simulação é $R = \begin{bmatrix} 0.75 & -0.433013 & 0.5 \\ 0.649519 & 0.625 & -0.433013 \\ -0.125 & 0.649519 & 0.75 \end{bmatrix}$.

A distribuição dos pontos de calibração considerados, na terceira simulação, no plano de calibração e na memória *frame* está representada na *Fig. 39*.

<i>filename</i>	<i>npoint</i>	<i>zwplane</i>	<i>xcenterbuf</i>	<i>ycenterbuf</i>	<i>xsizebuf</i>
out3.txt	16	0.0	256	256	512
<i>ysizebuf</i>	<i>rotx</i>	<i>roty</i>	<i>rotz</i>	<i>tx</i>	<i>ty</i>
512	-30.0	-30.0	-30.0	100.0	100.0
<i>tz</i>	<i>f</i>	<i>dx</i>	<i>dy</i>	<i>sx</i>	<i>k1</i>
2000.0	50.0	8,37765957e-3	8.075601357e-3	0.710935	-0.1

Tab. III - Parâmetros de entrada para a implementação *simcamera* utilizados na terceira simulação apresentada.

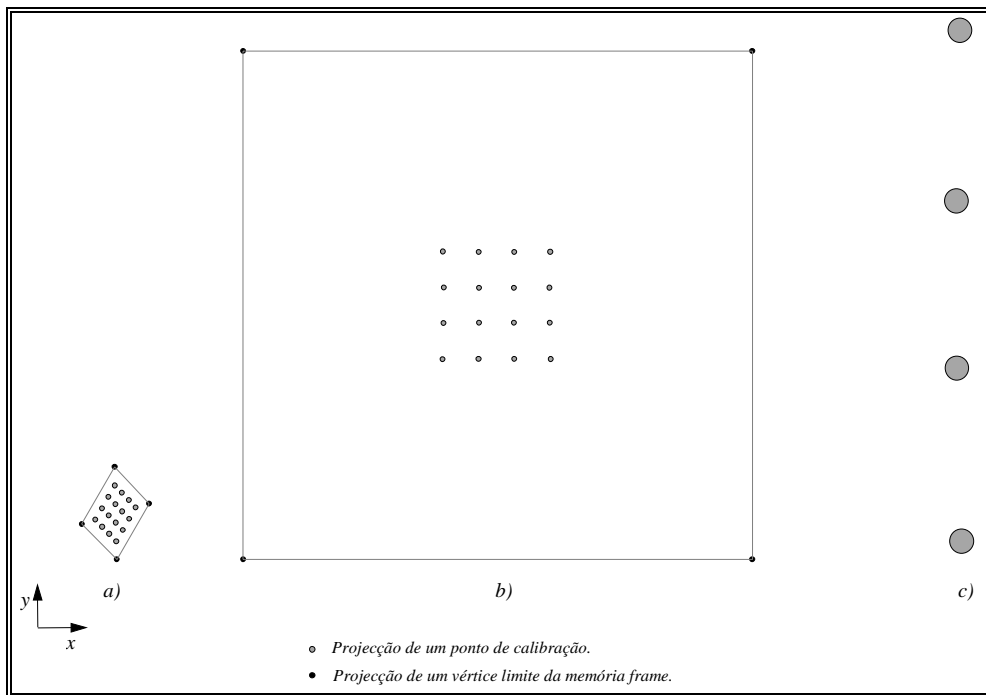


Fig. 39 - Distribuição dos pontos de calibração para a terceira simulação apresentada: a) no plano de calibração, b) na memória *frame*, c) ampliação da coluna dos pontos de calibração na memória *frame* mais à direita.

Como se observa da *Fig. 39*, a distribuição dos pontos de calibração também não é regular na memória *frame*. Esta não regularidade deve-se ao elevado valor da distorção radial da lente k_1 , devendo-se notar que, desta feita, a forma do conjunto dos pontos de calibração na memória *frame* é côncava, ao contrário da simulação anterior em que era convexa. Para o caso do plano de calibração, a regularidade deve-se novamente à filosofia da implementação *simcamera*.

A título de exemplo está representado na *Fig. 40* o conteúdo do ficheiro de saída “out3.txt” da implementação *simcamera*, que contém as coordenadas 3D mundo (x_w, y_w, z_w) e as coordenadas 2D na memória *frame* (X_f, Y_f) para os pontos de calibração considerados na última simulação apresentada.

-140.445 -40.9279 0 309.713 202.274	...
-154.461 -26.5676 0 237.441 202.626	-147.453 -33.7477 0 273.243 202.691
-133.908 -29.6422 0 308.742 238.746	-161.469 -19.3874 0 201.339 202.083
-147.892 -15.1754 0 237.772 238.86	-140.9 -22.4088 0 272.932 238.879
-127.371 -18.3565 0 308.646 274.123	-154.884 -7.94198 0 202.33 238.691
-141.323 -3.78324 0 237.802 274.012	-134.347 -11.0699 0 272.901 273.988
-120.834 -7.07081 0 309.405 310.148	-148.299 3.5034 0 202.424 274.193
-134.754 7.60892 0 237.538 309.805	-127.794 0.269056 0 273.141 309.736
...	-141.714 14.9488 0 201.643 310.35

Fig. 40 - Conteúdo do ficheiro de saída da implementação *simcamera* para a última simulação apresentada.

Como conclusão dos resultados obtidos nas simulações executadas por *simcamera*, quer as que aqui estão relatadas, quer muitas outras, pode-se salientar:

- a distribuição dos pontos de calibração no plano de calibração é sempre regular, pois esta é a filosofia da implementação *simcamera*;
- a forma do conjunto dos pontos de calibração no plano de calibração depende dos valores utilizados para as rotações segundo os três eixos principais x , y e z ;
- a distribuição dos pontos de calibração na memória *frame* é mais ou menos regular, a menos que o factor de distorção radial da lente k_1 apresente valores elevados, sendo de forma convexa quando este factor é positivo e côncavo quando é negativo;
- no plano de calibração, uma alteração da distância focal efectiva f implica um processo de *zoom* na distribuição dos pontos de calibração.

4.3 - Utilizando a implementação *calcamera* para calibração de câmaras

Neste ponto, são apresentados resultados de algumas calibrações de câmaras utilizando a implementação *calcamera*, apresentada no ponto 3.2.

Na primeira calibração, utilizou-se um conjunto pontos de calibração determinados pela implementação *simcamera* (apresentada no ponto 3.4), com os parâmetros de entrada apresentados na Tab. IV.

<i>filename</i>	<i>npoint</i>	<i>zwplane</i>	<i>xcenterbuf</i>	<i>ycenterbuf</i>	<i>xsizebuf</i>
out1.txt	16	0.0	256	256	512
<i>ysizebuf</i>	<i>rotx</i>	<i>roty</i>	<i>rotz</i>	<i>tx</i>	<i>ty</i>
512	-30.0	-30.0	-30.0	100.0	100.0
<i>tz</i>	<i>f</i>	<i>dx</i>	<i>dy</i>	<i>sx</i>	<i>k1</i>
2000.0	50.0	8,37765957e-3	8.075601357e-3	0.710935	0.001

Tab. IV - Parâmetros de entrada para a implementação *simcamera* utilizados na geração do conjunto de pontos de calibração utilizados na primeira calibração apresentada.

Com as rotações segundo os eixos principais indicadas na Tab. IV, a matriz de rotação 3D respectiva é

$$R = \begin{bmatrix} 0.75 & -0.433013 & 0.5 \\ 0.649519 & 0.625 & -0.433013 \\ -0.125 & 0.649519 & 0.75 \end{bmatrix}$$

Com este conjunto de pontos de calibração utilizou-se a implementação *calcamera* com os parâmetros de entrada apresentados na Tab. V. Os resultados obtidos pela implementação *calcamera* estão apresentados na Fig. 41, através do conteúdo do ficheiro de saída “*output.txt*”.

Pela observação da Fig. 41 conclui-se que a implementação *calcamera*, mesmo para apenas dezasseis pontos de calibração, apresenta resultados de boa qualidade. De notar que para *nlstop* igual a $1e-6$ os resultados já seriam de qualidade inferior, Fig. 42. Desta forma, verifica-se a influência do

parâmetro $nlstop$ na qualidade dos resultados obtidos, nomeadamente, na distância focal efectiva f , na translação segundo o eixo z e na distorção radial da lente k_1 .

<i>outfile</i>	<i>nlstop</i>	<i>sx</i>	<i>dx</i>	<i>dy</i>
output.txt	1e-8	0.710935	8.37765957e-3	8.075601357e-3
	<i>Cx</i>	<i>Cy</i>	<i>findcenter</i>	<i>infile</i>
	256	256	0	out1.txt

Tab. V - Parâmetros de entrada para a implementação *calcamera* utilizados na primeira calibração apresentada.

```

Results for CALCAMERA with:

  Stop for nonlinear equations=1e-08, Find center=0, sx=0.710935, dx=0.00837766,
  dy=0.0080756, Cx=256, Cy=256, Input file: out1.txt with 16 calibration points.

Rotation matrix R:

      +0.750001 -0.433016  +0.499996
      +0.649519 +0.624998  -0.433015
      -0.124994 +0.649519  +0.750001

Translation vector T: (+100.000, +100.000, +2000.00)

Focal length f: +50.0006

Radial lens distortion k1: +0.00101099
    
```

Fig. 41 - Conteúdo do ficheiro de saída da implementação *calcamera* para a primeira calibração apresentada.

```

Results for CALCAMERA with:

  Stop for nonlinear equations=1e-06, Find center=0, sx=0.710935, dx=0.00837766,
  dy=0.0080756, Cx=256, Cy=256, Input file: out1.txt with 16 calibration points.

Rotation matrix R:

      +0.750001 -0.433016  +0.499996
      +0.649519 +0.624998  -0.433015
      -0.124994 +0.649519  +0.750001

Translation vector T: (+100.000, +100.000, +1994.47)

Focal length f: +49.8619

Radial lens distortion k1: +0.000997387
    
```

Fig. 42 - Conteúdo do ficheiro de saída da implementação *calcamera* para a primeira calibração apresentada com $nlstop$ igual a $1e-6$.

Na segunda calibração, o objectivo principal foi a constatação da influência das coordenadas do centro da imagem (C_x, C_y). Para tal, utilizou-se um conjunto de pontos de calibração determinados novamente pela implementação *simcamera* com os parâmetros de entrada apresentados na Tab. VI. Com as rotações segundo os eixos principais indicadas na Tab. VI, a matriz de rotação 3D respectiva é

$$R = \begin{bmatrix} 0.75 & -0.433013 & 0.5 \\ 0.649519 & 0.625 & -0.433013 \\ -0.125 & 0.649519 & 0.75 \end{bmatrix}$$

Com este conjunto de pontos de calibração utilizou-se a implementação *calcamera* com os parâmetros de entrada apresentados na Tab. VII. Os resultados obtidos pela implementação *calcamera* estão apresentados na Fig. 43, através do conteúdo do ficheiro de saída “output.txt”.

<i>filename</i>	<i>npoint</i>	<i>zwplane</i>	<i>xcenterbuf</i>	<i>ycenterbuf</i>	<i>xsizebuf</i>
out2.txt	36	0.0	250	260	512
<i>ysizebuf</i>	<i>rotx</i>	<i>roty</i>	<i>rotz</i>	<i>tx</i>	<i>ty</i>
512	-30.0	-30.0	-30.0	100.0	100.0
<i>tz</i>	<i>f</i>	<i>dx</i>	<i>dy</i>	<i>sx</i>	<i>k1</i>
2000.0	50.0	8,37765957e-3	8.075601357e-3	0.710935	0.001

Tab. VI - Parâmetros de entrada para a implementação *simcamera* utilizados na geração do conjunto de pontos de calibração utilizados na segunda calibração apresentada.

<i>outfile</i>	<i>nstop</i>	<i>sx</i>	<i>dx</i>	<i>dy</i>
output.txt	1e-8	0.710935	8.37765957e-3	8.075601357e-3
<i>Cx</i>	<i>Cy</i>	<i>findcenter</i>	<i>infile</i>	
256	256	0	out2.txt	

Tab. VII - Parâmetros de entrada para a implementação *calcamera* utilizados na segunda calibração apresentada.

```

Results for CALCAMERA with:

  Stop for nonlinear equations=1e-08, Find center=0, sx=0.710935, dx=0.00837766,
  dy=0.0080756, Cx=256, Cy=256, Input file: out2.txt with 36 calibration points.

Rotation matrix R:

      +0.750088 -0.433475  +0.499467
      +0.649440 +0.625415  -0.432532
      -0.124882 +0.648811  +0.750632

Translation vector T: (+98.5671, +101.295, +2009.71)

Focal length f: +50.2419

Radial lens distortion k1: +0.00100699
    
```

Fig. 43 - Conteúdo do ficheiro de saída da implementação *calcamera* para a segunda calibração apresentada.

Pela observação da Fig. 43 conclui-se que a implementação *calcamera*, para trinta e seis pontos de calibração, já não apresenta resultados de tão boa qualidade como na calibração anterior. Esta perda de qualidade deve-se ao facto das coordenadas correctas do centro da imagem não serem iguais às passadas como parâmetros à implementação *calcamera*. Assim se o parâmetro *findcenter* for igual a 1, a implementação *calcamera* determinará as coordenadas correctas do centro da imagem e os resultados obtidos serão novamente de boa qualidade, Fig. 44.

Se, em vez de ser considerado um conjunto de trinta e seis pontos de calibração, for considerado um conjunto de oitenta e um pontos, mantendo todas as outras variáveis constantes, obtêm-se resultados ainda de melhor qualidade, Fig. 45.

Na terceira e última calibração apresentada, o objectivo principal foi a constatação de que mesmo para condições de *setup* teoricamente mais adversas (valores nulos para *rotx*, *rotz* e *tx*, e valores reduzidos para *roty* e *ty*), a implementação *calcamera* continua a oferecer resultados de boa qualidade. Para tal, utilizou-se um conjunto de pontos de calibração determinados novamente pela implementação *simcamera* com os parâmetros de entrada apresentados na Tab. VIII. Com as rotações segundo os eixos principais indicadas na Tab. VIII, a matriz de rotação 3D respectiva é

$$R = \begin{bmatrix} 0.996195 & 0 & 0.0871557 \\ 0 & 1 & 0 \\ -0.0871557 & 0 & 0.996195 \end{bmatrix}$$


```

Results for CALCAMERA with:

  Stop for nonlinear equations=1e-08, Find center=1, sx=0.710935, dx=0.00837766,
  dy=0.0080756, Cx=250, Cy=260, Input file: out2.txt with 36 calibration points.

Rotation matrix R:

      +0.750000 -0.433011  +0.500002
      +0.649520 +0.624995  -0.433018
      -0.124997 +0.649525  +0.749996

Translation vector T: (+100.000, +100.000, +1999.90)

Focal length f: +49.9982

Radial lens distortion k1: +0.00100932
    
```

Fig. 44 - Conteúdo do ficheiro de saída da implementação *calcamera* para a segunda calibração apresentada com *findcenter* igual a 1.

```

Results for CALCAMERA with:

  Stop for nonlinear equations=1e-08, Find center=1, sx=0.710935, dx=0.00837766,
  dy=0.0080756, Cx=250, Cy=260, Input file: out3.txt with 81 calibration points.

Rotation matrix R:

      +0.750000 -0.433013  +0.500000
      +0.649519 +0.625000  -0.433013
      -0.125000 +0.649519  +0.750000

Translation vector T: (+100.000, +100.000, +1999.96)

Focal length f: +49.9997

Radial lens distortion k1: +0.00101039
    
```

Fig. 45 - Conteúdo do ficheiro de saída da implementação *calcamera* para a segunda calibração apresentada com *findcenter* igual a 1 e oitenta e um pontos de calibração utilizados.

<i>filename</i>	<i>npoint</i>	<i>zwplane</i>	<i>xcenterbuf</i>	<i>ycenterbuf</i>	<i>xsizebuf</i>
out4.txt	36	0.0	250	260	512
<i>ysizebuf</i>	<i>rotx</i>	<i>roty</i>	<i>rotz</i>	<i>tx</i>	<i>ty</i>
512	0.0	-5.0	0.0	0.0	5.0
<i>tz</i>	<i>f</i>	<i>dx</i>	<i>dy</i>	<i>sx</i>	<i>k1</i>
2000.0	50.0	8.37765957e-3	8.075601357e-3	0.710935	0.001

Tab. VIII - Parâmetros de entrada para a implementação *simcamera* utilizados na geração do conjunto de pontos de calibração utilizados na terceira calibração apresentada.

Com este conjunto de pontos de calibração utilizou-se a implementação *calcamera* com os parâmetros de entrada apresentados na Tab. IX. Os resultados obtidos pela implementação *calcamera* estão apresentados na Fig. 46, através do conteúdo do ficheiro de saída “*output.txt*”.

<i>outfilename</i>	<i>nstop</i>	<i>sx</i>	<i>dx</i>	<i>dy</i>
output.txt	1e-8	0.710935	8.37765957e-3	8.075601357e-3
<i>Cx</i>	<i>Cy</i>	<i>findcenter</i>	<i>infilename</i>	
256	256	1	out4.txt	

Tab. IX - Parâmetros de entrada para a implementação *calcamera* utilizados na terceira calibração apresentada.

```

Results for CALCAMERA with:

  Stop for nonlinear equations=1e-08, Find center=1, sx=0.710935, dx=0.00837766,
  dy=0.0080756, Cx=250, Cy=260, Input file: out4.txt with 36 calibration points.

Rotation matrix R:

      +0.996195   -5.39719e-07   +0.0871497
      -1.80379e-07  +1.00000   +8.25637e-06
      -0.0871497   -8.24067e-06   +0.996195

Translation vector T: (+1.43148e-05, +5.00001, +2000.22)

Focal length f: +50.0059

Radial lens distortion k1: +0.00100863
    
```

Fig. 46 - Conteúdo do ficheiro de saída da implementação *calcamera* para a terceira calibração apresentada.

Pela observação da Fig. 46 conclui-se que, mesmo para condições de *setup* teoricamente mais adversas, a implementação *calcamera* continua a apresentar resultados de boa qualidade.

Como conclusão, das calibrações efectuadas verifica-se que a implementação *calcamera* obtém em média resultados de boa qualidade. No entanto, esta qualidade é afectada por algumas características do *setup* prático, nomeadamente:

- as rotações segundo os três eixos principais: a rotação segundo o eixo principal y não deverá ser nula, e em média obtêm-se resultados de melhor qualidade quando as rotações segundo os três eixos principais têm todas valores não nulos;
- a translação segundo o eixo principal y não deverá ser nula;
- a distorção radial da lente não deverá ser elevada nem demasiado baixa, em particular quando se pretende determinar as coordenadas correctas do centro da imagem na memória *frame*;
- o factor de incerteza horizontal deverá ser o mais correcto possível, pois uma pequena variação no seu valor implica grandes variações nos parâmetros determinados;
- as coordenadas 3D dos pontos de calibração deverão ser o mais precisas possível, pois a qualidade dos parâmetros determinados é fortemente dependente desta precisão;
- as coordenadas 2D dos pontos de calibração na memória *frame* deverão ser o mais precisas possível, pois a qualidade dos parâmetros determinados é fortemente dependente desta precisão;
- o número de pontos de calibração deverá ser razoável (na ordem de algumas dezenas).

As duas primeiras características do *setup* prático são facilmente obtidas de modo a corresponder a resultados da calibração de boa qualidade. Com o mesmo objectivo, a terceira característica é geralmente sempre verificada, pois as actuais lentes são em média de boa qualidade. Do mesmo modo, a quarta característica deverá ser conseguida, quer pela utilização da implementação *findsx*, quer pela medição das frequências dos relógios do conversor analógico/digital e de varrimento dos sensores da câmara envolvida. A quinta característica deverá ser necessariamente conseguida, nem que isso obrigue a dispor de meios sofisticados de medição. Sem dúvida que esta característica, que o *setup* prático deverá possuir, de forma aos resultados de calibração serem de boa qualidade, é a mais difícil de se conseguir na prática (como foi constatado no decurso do trabalho para esta dissertação). Com o mesmo objectivo, a sexta característica deverá ser conseguida com êxito com uma implementação de um qualquer método de determinação das coordenadas na memória *frame* dos pontos de calibração que possua precisão ao nível do *subpixel*, por exemplo, com a implementação *calpoint*, apresentada no ponto 3.3. A última característica é geralmente satisfeita, de forma a corresponder a bons resultados de calibração, quando é utilizado um conjunto de trinta e seis pontos de calibração.

Refira-se que o valor do parâmetro de entrada *nlstop* para a implementação *calcamera* - variável de controle da convergência da determinação não linear da translação segundo o eixo *z*, da distância focal efectiva *f* e da distorção radial da lente *k_r*, pelo método não linear de Levenberg-Marquardt - deverá ter um valor apropriado de forma a corresponder a resultados com a melhor qualidade possível. Este valor pode ser determinado se for conhecido previamente o valor que deverá apresentar a distância focal efectiva *f* e/ou a translação segundo o eixo *z*, de forma a que os resultados obtidos pela calibração se aproximem o mais possível dos esperados.

5 - Algumas conclusões finais

Neste capítulo são apresentados alguns métodos para a calibração de câmaras. Assim, são descritos seis métodos de calibração, começando por um método mais simples e menos apropriado ao domínio da visão tridimensional - que considera um modelo perfeito *pinhole* para a câmara e utiliza um conjunto de pontos de calibração não coplanares - até a um método bastante mais complexo e mais apropriado ao mesmo domínio - este método considera distorção radial da lente modelada por um termo, um factor de incerteza horizontal (devido a vários factores, como má sincronização temporal entre o *hardware* de aquisição de imagem e o de varrimento da câmara, ou de imprecisões no sincronismo de varrimento TV), a possibilidade de determinação das coordenadas correctas do centro da imagem na memória *frame*, a consideração de o referencial de coordenadas na memória *frame* não ter os eixos *x* e *y* paralelos aos mesmos eixos do referencial câmara (o que na pratica é considerar que a imagem sofre uma transformação entre estes dois referenciais constituída por uma rotação e uma distorção além do escalonamento habitualmente considerado) e utiliza pontos de calibração coplanares. É também apresentado o método de calibração concebido e implementado no âmbito desta dissertação; trata-se de um método constituído por três etapas, das quais a primeira consiste na determinação do factor de incerteza horizontal, a segunda consiste na determinação das coordenadas correctas do centro da imagem na memória *frame*, e a terceira e última etapa, consiste na determinação dos restantes parâmetros intrínsecos e extrínsecos da câmara. Este método de calibração utiliza também pontos de calibração coplanares, o que se traduz numa solução prática interessante, pois é ainda difícil, e em algumas situações de todo indesejável, conseguir-se um conjunto de pontos não coplanares de forma rigorosa e precisa. Do mesmo modo, são também apresentadas uma implementação deste método de calibração, assim como alguns resultados experimentais obtidos. Como se verifica pelos resultados apresentados, as exigências ao nível do *setup* prático deste método de calibração são facilmente conseguidas de modo a obter-se resultados de boa qualidade.

Neste domínio da visão por computador, é também necessário um método para determinar com boa precisão as coordenadas 2D na memória *frame* de pontos característicos para serem utilizados como pontos de calibração. Assim, são também apresentados neste capítulo vários métodos existentes para este fim, bem como o método concebido no âmbito desta dissertação, com resolução ao nível do *subpixel*. Trata-se de uma implementação que apresenta bons resultados e que pode ser facilmente utilizada.

Com o intuito de obter um simulador de câmaras de geometria interna conhecida, utilizando o mesmo modelamento teórico do processo de aquisição de imagem do método de calibração concebido, são apresentados um método para tal, uma sua implementação e alguns resultados experimentais obtidos. Torna-se, assim, possível determinar de forma automática as coordenadas 3D mundo e as respectivas coordenadas 2D na memória *frame* de conjuntos de pontos de calibração coplanares, para serem utilizados posteriormente com a implementação do método de calibração de câmaras, sem a necessidade de montar todo o *setup* prático, constituindo assim, uma solução mais flexível e onde todas as variáveis envolvidas são facilmente controladas.

Todas as implementações apresentadas oferecem resultados de boa qualidade, constituindo um conjunto de ferramentas apropriadas para uma solução global do problema de calibração de câmaras utilizadas em visão tridimensional.

Capítulo III

INTRODUÇÃO À DETECÇÃO DE ORLAS DE INTENSIDADE SEGUIMENTO DE LINHAS E APROXIMAÇÃO POLIGONAL UTILIZANDO FAIXAS FUSÃO DE SEGMENTOS DE RECTA PRÓXIMOS E DE DIRECÇÕES SIMILARES

Neste capítulo é induzida uma solução global para a determinação numa dada imagem das entidades que irão ser objecto do seguimento ao longo da sequência; isto é, a determinação dos segmentos de recta mais significativos presentes em cada imagem.

A abordagem usada consiste em classificar os *pixels* que constituem a imagem original em termos de direcção e amplitude do gradiente - utilizando para o efeito um detector de orlas de intensidade -, realizar de seguida um seguimento das linhas presentes na mesma imagem e a subsequente aproximação poligonal e, finalmente, realizar fusões de segmentos de recta que provavelmente pertencerão a uma mesma entidade.

Assim, neste capítulo é apresentada uma introdução à detecção de orlas de intensidade, métodos de aproximação poligonal utilizando faixas, um método para o seguimento de linhas (baseado na direcção e em dois níveis de amplitude) e consequente aproximação poligonal utilizando faixas dinâmicas e métodos de fusão de segmentos de recta que sejam próximos e de direcção similares.

São ainda apresentadas as implementações desenvolvidas, assim como alguns resultados experimentais obtidos.

Introdução

A detecção de orlas de intensidade - variações de intensidade provocadas por vários fenómenos físicos - tem um papel muito significativo nos mais variados domínios da visão por computador, como, por exemplo, na segmentação de uma imagem, na extracção de características, na determinação de características do movimento, na obtenção de estrutura tridimensional, etc. Basicamente, pode-se afirmar que quando se pretende analisar uma dada imagem torna-se, na maior parte das vezes, necessário determinar as orlas de intensidade; tal determinação torna a imagem original num mapa de orlas que possui uma quantidade de informação visual drasticamente reduzida conservando, no entanto, toda a informação original da estrutura dos objectos e da cena.

Sendo esta detecção de tão grande importância para a visão por computador entendeu-se por bem apresentar, na primeira secção deste capítulo, uma breve introdução a esse tema. De referir, no entanto, que não se procedeu a nenhum desenvolvimento nesta área; esta introdução é essencialmente adaptada de [Correia, 1995]; o filtro de detecção utilizado - o filtro de Deriche - foi desenvolvido e implementado pelo autor da referida publicação.

Qualquer imagem de uma cena real será inevitavelmente constituída por um elevado número de linhas, sendo assim bastante comum a consideração de tais entidades como estruturas elementares. Com o intuito de diminuir o número destas entidades, sem contudo diminuir demasiadamente a informação contida na imagem, é normalmente efectuada uma aproximação às entidades originais por entidades mais simples. Uma das aproximações mais comuns é a poligonal, que basicamente consiste na substituição de uma linha por um conjunto ligado de segmentos de recta, de forma a satisfazer determinados critérios.

Na segunda secção são apresentados dois métodos de aproximação poligonal utilizando faixas de aproximação: um, utilizando faixas estáticas, e um segundo que utiliza faixas dinâmicas. É ainda apresentado um método, concebido e implementado no âmbito desta dissertação, para o seguimento de linhas e consequente aproximação poligonal utilizando faixas dinâmicas. Neste método considera-se que todos os *pixels* da imagem original estão previamente classificados, utilizando um detector de orlas de intensidade apropriado, em termos de amplitude e de direcção. No seguimento dos diversos pontos que constituem as diferentes linhas originais é utilizada a direcção e dois níveis de amplitude.

Após o seguimento e a aproximação poligonal das linhas presentes numa imagem, pode ainda ser útil a fusão de segmentos de recta que correspondam, presumivelmente, a uma mesma entidade. Esta correspondência pode ser graduada em exigência, utilizando critérios adequados, tais como proximidade e similitude de direcções. Assim, na terceira secção deste capítulo são apresentados três métodos para se proceder a tal fusão, incluindo o concebido e implementado no âmbito desta dissertação.

Nas três últimas secções do capítulo, são ainda apresentadas as três implementações desenvolvidas neste domínio, alguns resultados experimentais obtidos por tais implementações, e algumas conclusões finais.

De referir que, com a abordagem global descrita neste capítulo, se torna possível passar de uma imagem constituída por um elevado número de linhas a um conjunto, com um número provavelmente bastante mais reduzido, de segmentos de recta e preservando, no entanto, a informação essencial, *Fig. 1*.

1 - Introdução à detecção de orlas de intensidade

Uma imagem apresenta variações de intensidade que são causadas por vários fenómenos físicos distintos. Diferenças na iluminação - como, por exemplo, fontes de luz localizadas, sombras e espelhamentos -, diferenças na orientação e distância de superfícies de objectos relativamente ao observador, variações mais ou menos graduais da reflexão de superfícies, textura e cor dos objectos na cena, são alguns desses fenómenos.

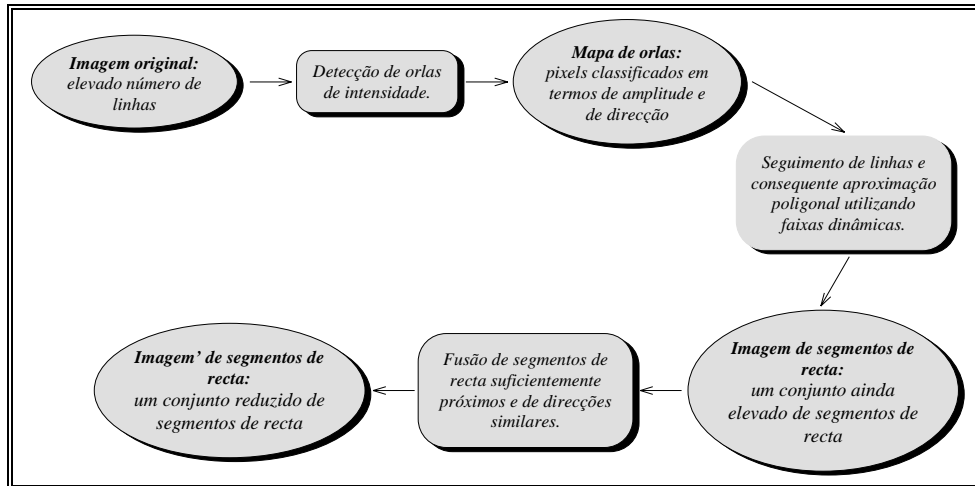


Fig. 1- A abordagem global usada neste capítulo.

Numa imagem dum cena real encontram-se, normalmente, um misto de regiões nas quais a intensidade varia suavemente e de regiões que apresentam variações bruscas, dando origem a uma vasta gama de possíveis transições de intensidade, quer seja na sua extensão, amplitude ou forma. Essas transições caracterizam-se por perfis espaciais. Na Fig. 2, apresentam-se alguns exemplos típicos de variações de intensidade em imagens em tons de cinzento, bem como os perfis que as caracterizam.

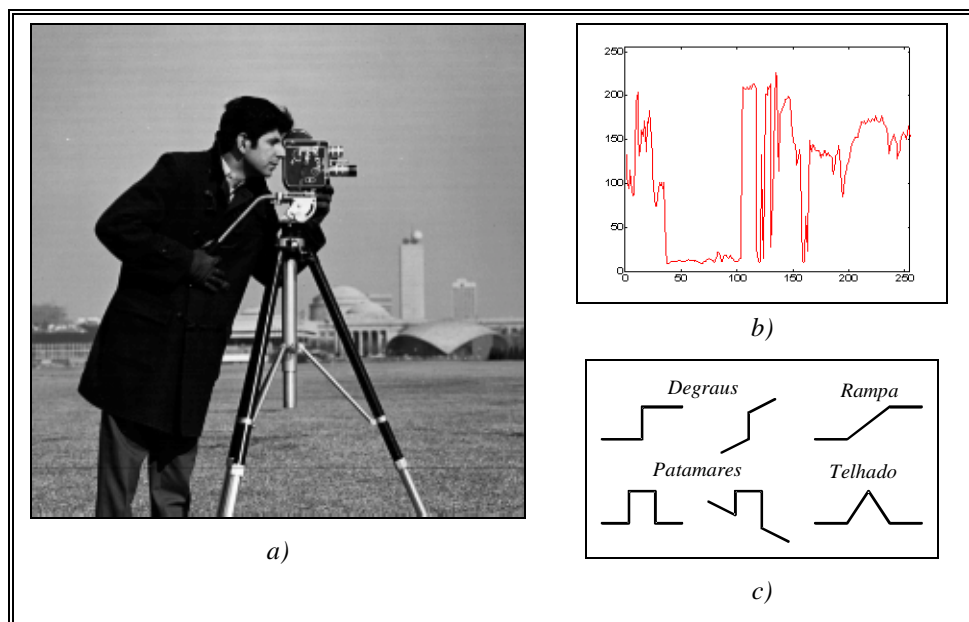


Fig. 2 - a) Imagem em tons de cinzento; b) perfil de variações numa linha da imagem; c) perfis modelo de variações de intensidade.

As orlas de intensidade surgem nas fronteiras entre objectos, nos contornos das suas sombras e nas discontinuidades em superfícies, constituindo, por isso, uma das mais importantes características que permitem dar um primeiro passo no sentido de interpretar e compreender uma imagem dum cena real. A detecção de orlas tem a propriedade de reduzir drasticamente a quantidade de informação visual a processar, preservando, no entanto, toda a informação de estrutura dos objectos e da cena. A imagem resultante, consistindo apenas de orlas, é bastante inteligível, uma vez que se assemelha a um esboço ou desenho geométrico dos objectos presentes na cena.

A detecção de orlas de intensidade contribui significativamente para as aplicações de segmentação, detecção de características, análise de movimento, obtenção de estrutura tridimensional, etc.

A detecção de orlas de intensidade é, pois, das operações mais elementares e fundamentais na percepção visual. As técnicas e métodos de detecção têm sido estudados num contexto psico-fisiológico e desenvolvidos na perspectiva do processamento digital de imagens. Existem basicamente três categorias diferentes para esta operação: métodos estocásticos, ajuste de superfícies e filtragem. Nesta secção apresentam-se, de forma introdutória, apenas métodos pertencentes à última categoria - a filtragem. Esta escolha deve-se ao facto de serem métodos de realização mais simples e que podem facilmente ser implementados por algoritmos rápidos, quer em *software*, quer em *hardware*. Revisões aos outros métodos encontram-se nos trabalhos de Huang e Tseng [Huang, 1988] - método estocástico -, de Haralick [Haralick, 1984] e de Nalwa e Binford [Nalwa, 1986] - ajuste de superfícies.

No ponto que se segue abordam-se as operações de diferenciação e as suas expressões, vantagens e inconvenientes como método de detecção de orlas de intensidade em imagens. Apresenta-se no ponto **1.2** o método baseado na aplicação de filtros óptimos de detecção, adaptados às formas e escalas das variações de interesse. É também apresentado o filtro utilizado ao longo do trabalho desta dissertação - o filtro de Deriche -, a justificação da sua escolha, e um exemplo de resultados obtidos por este filtro numa imagem real. Os critérios de desempenho mais comuns são também apresentados neste ponto. No ponto **1.3** referem-se algumas operações fundamentais, que devem ser realizadas após a filtragem, para uma correcta detecção das orlas de intensidade em imagens.

1.1 - Diferenciação e operadores afins

Uma imagem pode ser modelada por uma função a duas dimensões definida num domínio normalmente limitado que representa o campo de visão. Esta abordagem é bastante comum no contexto do processamento digital de imagens, permitindo descrever as operações sobre imagens de forma objectiva pelo recurso à formulação matemática, como é o caso da detecção de orlas. Nesta perspectiva, as orlas de intensidade correspondem a variações acentuadas da função-imagem e a sua detecção requer a aplicação duma operação adequada.

A diferenciação é a operação matemática que permite determinar e caracterizar as variações de uma função. Os pontos em que essa variação é acentuada e que, portanto, possam constituir uma orla, podem então ser obtidos por determinação dos máximos de amplitude da primeira derivada ou das passagens por zero da segunda derivada da função em análise. Os operadores de diferenciação, que se apresentam de seguida, constituem uma primeira abordagem para o problema da detecção de orlas de intensidade em imagens.

1.1.1 - Gradiente de uma imagem

Considerando uma imagem como uma função bidimensional $I(x, y)$, a primeira derivada corresponde ao gradiente dado por:

$$\nabla I(x, y) = \frac{\partial I(x, y)}{\partial x} \hat{u}_x + \frac{\partial I(x, y)}{\partial y} \hat{u}_y, \quad (Eq. 1)$$

em que \hat{u}_x e \hat{u}_y são os versores associados ao referencial XY da função-imagem.

Os pontos nos quais a amplitude de $\nabla I(x, y)$ seja superior a um nível arbitrado e se verifique um máximo local são considerados elementos de orla e marcados na imagem como tal, *Fig. 3*.

O gradiente permite também determinar a amplitude e a orientação de cada elemento de orla pelo cálculo de:

$$\left| \nabla f(x, y) \right| = \sqrt{(I'_x(x, y))^2 + (I'_y(x, y))^2}, \quad \angle(\nabla I(x, y)) = \arctan\left(\frac{I'_y(x, y)}{I'_x(x, y)}\right), \quad (Eqs. 2a e 2b)$$

em que $I'_x(x, y)$ e $I'_y(x, y)$ são as primeiras derivadas da função em ordem a x e y , respectivamente. A orientação de cada elemento é uma característica importante, uma vez que pode ser usada para relacionar pontos pertencentes a uma mesma orla física.

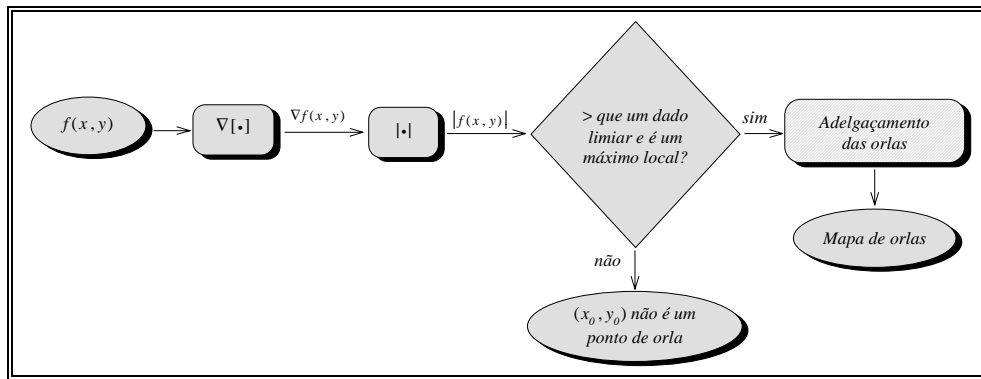


Fig. 3 - Esquema de funcionamento do detector de gradiente.

No domínio discreto, o gradiente é transformado em equações às diferenças que, a duas dimensões, constituem máscaras discretas de convolução. As aproximações mais vulgarmente utilizadas são as máscaras de Sobel, Roberts e Prewitt, que se representam na Fig. 4. As máscaras h_x equivalem à derivada segundo x , e as máscaras h_y equivalem à derivada segundo y , tendo, por isso, um carácter direccional.

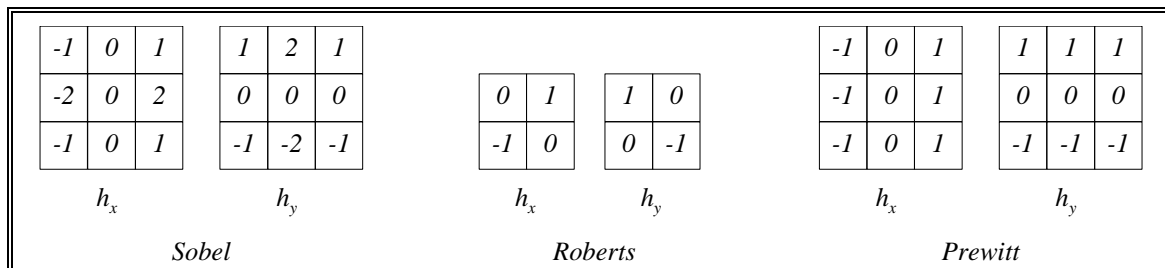


Fig. 4 - Máscaras de convolução do gradiente.

A operação de diferenciação de uma imagem, no domínio discreto, consiste então na convolução dessa imagem com o filtro cuja resposta impulsional é dada por $h_n(x, y)$ em que o versor \hat{n} indica a direcção de maior variação de intensidade em cada área da imagem - o gradiente - que, por definição, corresponde à direcção normal às linhas de nível da intensidade.

A detecção de orlas, com uma orientação inicialmente desconhecida, é realizada através da combinação dos resultados da convolução da imagem original com pelo menos duas máscaras com respostas direccionais perpendiculares, dado que a derivada segundo uma direcção arbitrária num espaço bidimensional pode ser expressa, por exemplo, em termos das derivadas parciais segundo os eixos coordenados.

No caso de imagens sem ruído basta usar essas duas direcções perpendiculares; todavia, numa imagem com ruído, a consideração de um maior número de direcções permite melhorar a relação Sinal-Ruído do resultado da convolução [Torre, 1986].

Na utilização destes operadores surgem, no entanto, algumas dificuldades:

- A selecção do nível de amplitude a partir do qual se marcam as orlas pode dar origem ao aparecimento de traços largos em vez de orlas bem demarcadas e estreitas; nesse caso é necessária uma operação de adelgaçamento aplicada à imagem resultante da convolução. Esta operação consiste em eliminar todos os pontos que não sejam máximos locais, de preferência segundo a direcção da orla a que possam pertencer.
- Outro efeito indesejável que surge na aplicação dos operadores de gradiente é a descontinuidade dos contornos detectados, inerente ao carácter discreto das máscaras de convolução. As orlas que apresentam uma amplitude elevada segundo uma direcção terão certamente variações fracas na direcção perpendicular e, como tal, um contorno curvilíneo poderá ser detectado como várias pequenas linhas descontínuas. Para evitar essas quebras

podem-se utilizar métodos de limiarização específicos, como por exemplo, limiarização a dois níveis com histerese ou com base em histogramas locais (ver ponto 1.3).

1.1.2 - Laplaciano de uma imagem

Outra alternativa para a detecção de orlas consiste em determinar as passagens por zero da segunda derivada da função-imagem, mas apenas segundo a direcção de variação máxima em cada ponto. Embora numa primeira análise seja uma operação idêntica à anterior, apresenta a vantagem de se poder usar um operador independente da orientação - o Laplaciano -, cuja expressão num espaço bidimensional tem a forma:

$$\nabla^2 I(x, y) = \nabla(\nabla I(x, y)) = \frac{\partial^2 I(x, y)}{\partial x^2} + \frac{\partial^2 I(x, y)}{\partial y^2}.$$

Também neste caso, podemos passar ao domínio discreto e aproximar o Laplaciano por equações às diferenças, mas agora de segunda ordem, obtendo-se máscaras de convolução idênticas às que se ilustram na Fig. 5. Estas máscaras resultam também de se considerarem aproximações diferentes; contudo, ao contrário das anteriores, têm um carácter não-direccional.

0	1	0	1	1	1	-1	2	-1
1	-4	1	1	-8	1	2	-4	2
0	1	0	1	1	1	-1	2	-1
h_{14}			h_{18}			h_{24}		

Fig. 5 - Máscaras de convolução do Laplaciano.

O Laplaciano duma imagem obtém-se por uma única convolução $I(x, y) * h(x, y)$, sendo $h(x, y)$ dado por uma das máscaras da Fig. 5, da qual resultam as orlas presentes qualquer que seja a sua orientação. A máscara h_{14} na Fig. 5, no entanto, é pouco sensível às variações nas diagonais.

Contrariamente aos máximos do gradiente, as linhas de passagem por zero tendem a ser contínuas, fechadas e estreitas pois correspondem às denominadas linhas de nível de intensidade (neste caso nível zero). Assim, são desnecessárias as operações de adelgaçamento e as condições para garantir a continuidade dos contornos. Todavia, há uma condição que se deve verificar - as variações de intensidade na imagem devem ser lineares na sua direcção longitudinal para que o Laplaciano apresente resultados correctos [Marr, 1980; Torre, 1986]. Se tal não acontecer (por exemplo, nos cantos), as linhas de passagem por zero aparecem ligeiramente deslocadas junto das variações irregulares. Há também uma grande quantidade de pontos a serem falsamente marcados como orlas dado não ser imposta qualquer condição de amplitude mínima aceitável, como no caso do gradiente, ou de declive mínimo, que, neste caso, seria o mais adequado. Este factor é ainda mais agravado nas regiões em que a variância local da imagem é reduzida, pois qualquer pequena variação de intensidade na imagem produz passagens por zero do seu Laplaciano. Uma operação útil para reduzir este efeito é, precisamente, impor um limite mínimo de variância local para que um ponto possa ser marcado como elemento duma orla numa dada região [Lim, 1990], Fig. 6.

A maior desvantagem das operações de diferenciação é a sua grande sensibilidade ao ruído e a pequenos detalhes presentes na imagem, que está bem patente quer no Laplaciano quer no Gradiente, conforme se ilustra nas imagens da Fig. 7. É, por isso, comum fazer uma filtragem prévia para remoção de ruído que, no entanto, tem também efeitos sobre a própria localização das orlas.

Na Tab. 1 sumarizam-se as características das operações de diferenciação duma imagem. Torre e Poggio analisam exhaustivamente o problema da diferenciação numérica e demonstram analiticamente estas características, bem como a aplicação de alguns filtros para remoção de ruído, em [Torre, 1986].

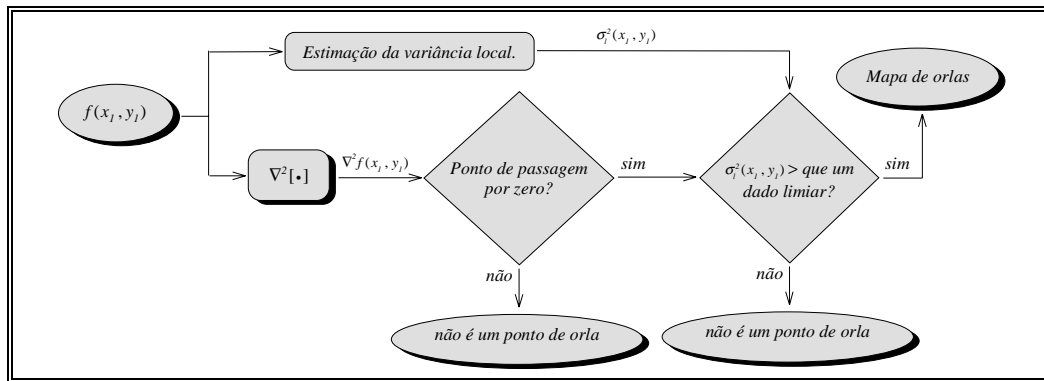


Fig. 6 - Esquema de funcionamento do detector Laplaciano de forma a não produzir demasiadas orlas falsas.

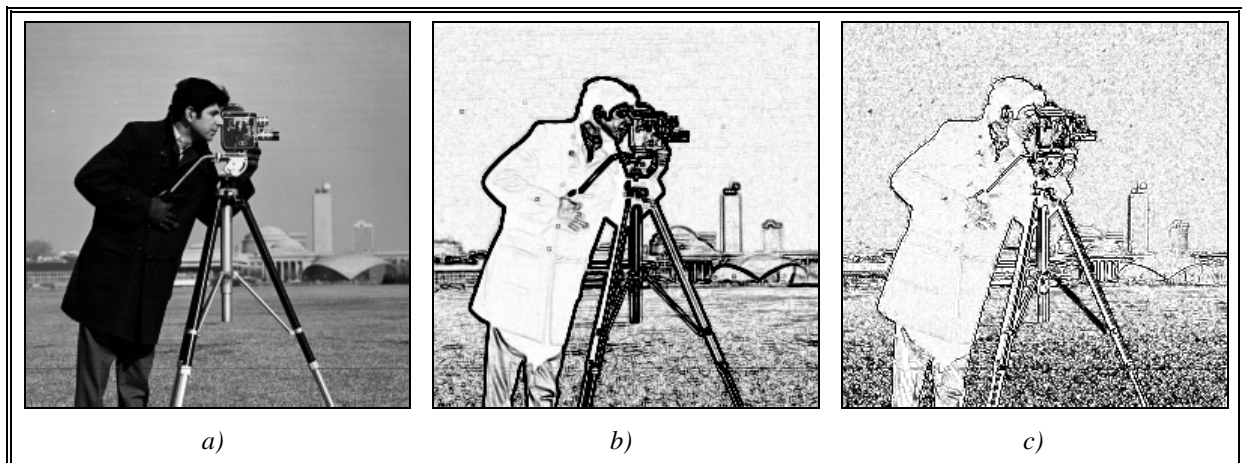


Fig. 7 - Resultados da detecção de orlas numa imagem real a) com: b) o operador de gradiente de Sobel, c) o Laplaciano (h_{18}).

	<i>Gradiente</i>	<i>Laplaciano</i>
<i>Direccionalidade dos operadores</i>	<i>sim</i>	<i>não</i>
<i>Número mínimo de convoluções</i>	<i>2</i>	<i>1</i>
<i>Linearidade das variações na imagem</i>	<i>favorável</i>	<i>necessária</i>
<i>Continuidade dos contornos detectados</i>	<i>fraca</i>	<i>boa</i>
<i>Espessura dos contornos detectados</i>	<i>variável</i>	<i>fixa</i>
<i>Detecção de orlas falsas</i>	<i>não</i>	<i>sim</i>
<i>Operações auxiliares</i>	<i>Adelgaçamento, limiarização</i>	<i>Validação das orlas</i>

Tab. 1 - Características da diferenciação de imagens.

1.2 - Filtros para detecção de orlas de intensidade

Experiências psico-fisiológicas levadas a cabo por vários investigadores durante as últimas décadas revelaram alguns dos aspectos quantitativos dos processos biológicos de visão [Marr, 1980]. É com base nessa informação e no conhecimento da fase periférica da percepção visual humana que assenta uma das teorias para detecção de orlas de intensidade em imagens.

Segundo Marr [Marr, 1982], a fase periférica da percepção visual humana tem por objectivo elaborar uma representação simples, mas completa, das imagens e fá-lo à custa de uma decomposição espectral em múltiplos canais de frequência que respondem selectivamente às variações de intensidade presentes nas imagens. Esses canais apresentam como resultado um conjunto de elementos simples - segmentos, manchas, etc. -, em diversas escalas espaciais, que correspondem a orlas de intensidade.

Esses elementos são posteriormente seleccionados e agrupados, de acordo com relações geométricas, para formar objectos mais complexos. Esta foi a ideia básica no trabalho de Marr e Hildreth em [Marr, 1980]: detectar as orlas de intensidade na imagem em várias escalas, através de filtros sintonizados para cada uma, e agrupar os elementos resultantes ao longo dessas diferentes escalas. No que respeita à detecção de orlas, fazem-se, naquele trabalho, duas observações importantes: uma, é que as variações de intensidade em imagens existem numa vasta gama de escalas espaciais e outra, é que essas variações têm uma extensão limitada. Deste modo, cada filtro deve ter um espectro de frequências de banda limitada, para detectar eficazmente as orlas numa dada escala, e a sua resposta deve ser localizada no domínio espacial.

A vasta gama de escalas de variação, aliada ao ruído inerente aos processos automáticos de aquisição e digitalização de imagens, dificulta a detecção de orlas por meios computacionais, na medida em que não é possível projectar um filtro único óptimo, para todas as escalas, e que sirva também em todos os casos - o que numa aplicação constitui uma variação acentuada de intensidade e pode ser classificado como orla, noutra aplicação pode já não o ser. No entanto, na maioria dos casos práticos, há apenas uma gama limitada de variações de interesse, bastando então um único filtro adaptado para detectar as orlas correspondentes, sem necessidade de recorrer à multiplicidade e aos agrupamentos acima referidos.

O projecto do filtro óptimo para cada escala de variações é vulgarmente encarado de dois modos diferentes que, no entanto, são equivalentes. Num desses modos (seguido por Marr e Hildreth), considera-se a aplicação de um filtro de suavização $h(x)$, para reduzir a gama de escalas das variações e o ruído na imagem $I(x)$, sendo a detecção de orlas realizada depois pela procura dos extremos em $D[I(x) * h(x)]$ ou das passagens por zero em $D^2[I(x) * h(x)]$ em que D e D^2 representam operadores diferenciais de primeira e segunda ordem, respectivamente. O outro modo (seguido por Canny em [Canny, 1986]) consiste em determinar a expressão do filtro óptimo para cada escala de variações com base na resposta que se pretende. E é aqui que surge a equivalência destes dois modos. Em primeiro lugar, porque a resposta do filtro de detecção óptimo a uma orla não poderá ser outra coisa que não um extremo ou uma passagem por zero e, em segundo lugar, porque pelas propriedades da convolução temos que $D[I(x) * h(x)] = I(x) * D[h(x)]$ e $D^2[I(x) * h(x)] = I(x) * D^2[h(x)]$, ou seja, as primeira e segunda derivadas do filtro de suavização constituem filtros de detecção cuja resposta a uma orla é, respectivamente, um máximo ou uma passagem por zero.

A detecção de orlas de intensidade pode assim ser realizada pela procura dos máximos de amplitude na resposta a um filtro com as características de uma derivação de primeira ordem ou pela procura das passagens por zero na resposta de um filtro que apresente características de uma derivação de segunda ordem (ver Fig. 8).

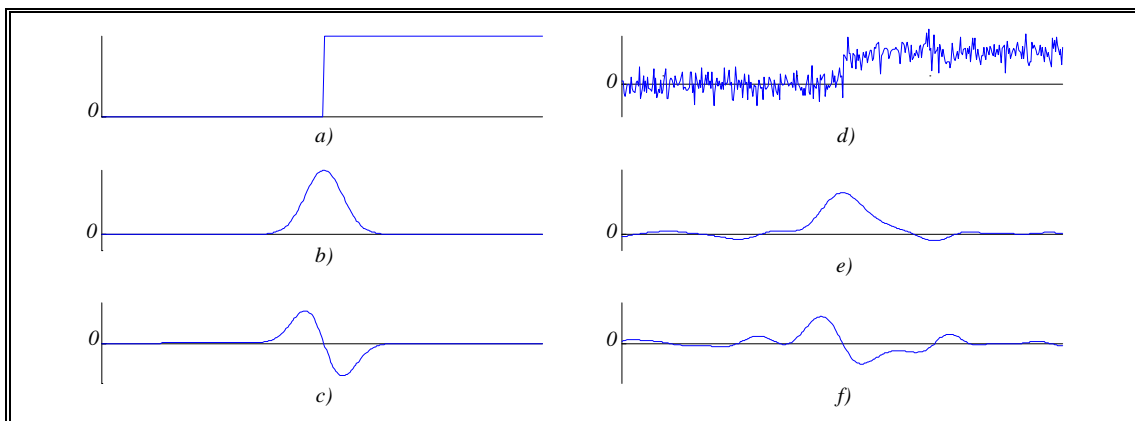


Fig. 8 - a) Orla em degrau; b) detecção de máximos de amplitude; c) detecção de zeros; d), e) e f), idem com ruído Gaussiano.

1.2.1 - Critérios de desempenho

A detecção de orlas em sinais é um problema comum a muitas aplicações, sendo, por isso, de todo o interesse dispor de critérios de desempenho genéricos para os detectores de orlas que sejam independentes das suas possíveis aplicações. Esses critérios são: a baixa probabilidade de erro, isto é, devem ser detectadas correctamente todas e apenas as orlas existentes no sinal e a sua localização deve ser tão precisa quanto possível. Sendo inicialmente referidos, na perspectiva do processamento digital de imagens, por Marr e Hildreth [Marr, 1980], a sua formulação matemática foi desenvolvida por Canny em [Canny, 1986] e formam a base de vários trabalhos e desenvolvimentos na detecção de orlas em imagens (ver por exemplo: [Deriche, 1987], [Sarkar, 1991], [Shen, 1992]).

Um filtro, caracterizado pela sua resposta impulsional, constitui um bom detector de orlas se, através da convolução com o sinal original, se conseguir:

- 1) **detecção eficaz** - baixas probabilidades de não assinalar orlas existentes e de assinalar orlas não existentes. Dado que estas probabilidades decrescem monotonamente com o aumento da relação Sinal-Ruído, um critério adequado consiste na maximização dessa relação.
- 2) **localização precisa** - as orlas devem ser assinaladas com precisão, o mais próximo possível do centro da orla real, e para isso, deve-se minimizar uma medida que represente essa distância.
- 3) **unicidade de resposta** - deve existir uma relação unívoca entre orlas reais e orlas assinaladas ou, por outras palavras, uma orla real deve ser assinalada uma única vez.

1.2.2 - O filtro utilizado no trabalho desta dissertação: Filtro de Deriche

Deriche em [Deriche, 1987] propõe um filtro idealizado com vista à detecção óptima de variações de intensidade em degrau. Assim, modifica as condições fronteira do filtro de Canny [Canny, 1986] para as de um filtro de resposta impulsional infinita, o que conduz à equação:

$$f_D = -c e^{-\alpha|x|} \sin(\omega x). \quad (\text{Eq. 3})$$

Deste modo, este filtro fica totalmente definido por apenas dois parâmetros α e ω , sendo o valor de c determinado pela condição de ganho que, normalmente, se pretende unitário.

Este filtro de Deriche, para detecção de orlas de intensidade, foi o utilizado ao longo do trabalho desta dissertação. Esta escolha deveu-se a vários factores, nomeadamente:

- a sua implementação ser recursiva, sendo assim eficiente;
- o tempo de execução depender apenas das dimensões da imagem na qual se pretende empregar;
- a afinação conseguida para diferentes tipos de imagens, pela utilização dos parâmetros α e ω .

A afinação conseguida por este filtro foi responsável pela sua escolha em detrimento do filtro de Shen e Castan¹: neste filtro apenas existe um parâmetro de ajuste sendo, assim, a mesma afinação mais difícil a diferentes tipos de imagem.

Na *Fig. 9* apresenta-se o resultado da detecção de orlas em imagens pelo filtro de Deriche. É de notar a diferença entre a imagem obtida e as resultantes da aplicação dos operadores de Sobel e do Laplaciano apresentadas na *Fig. 7*, nas quais é manifestamente mais elevada a amplitude do ruído em toda a imagem.

¹ Ver, por exemplo [Shen, 1992].

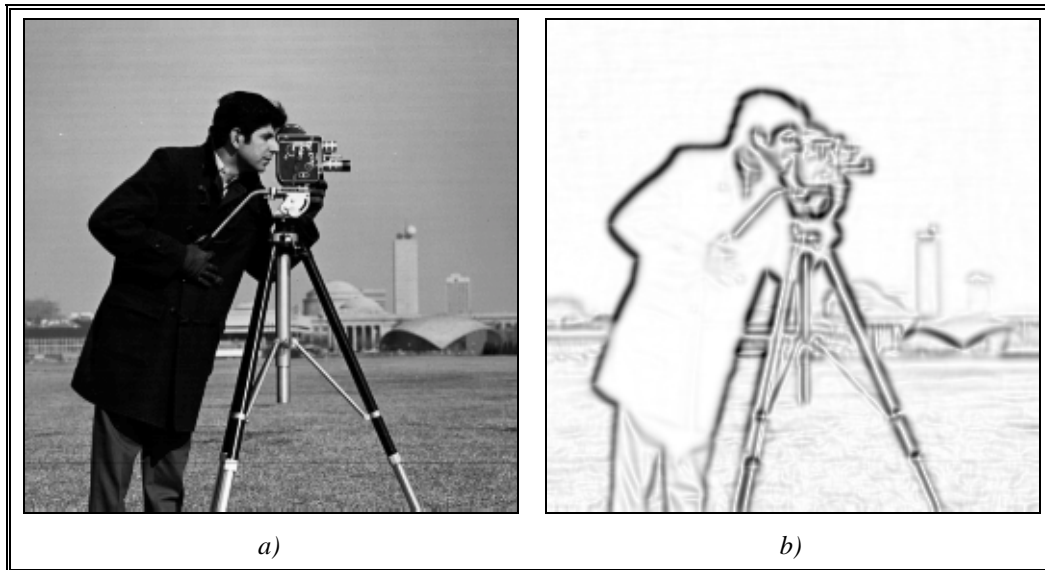


Fig. 9 - Resultado da detecção de orlas numa imagem real a) pelo filtro de Deriche, sem operações complementares, com $\alpha = 1.0$ e $\omega = 0.0001$, b).

1.3 - Operações complementares

A aplicação de um filtro para detecção de orlas de intensidade a uma imagem constitui de facto apenas um estágio de processamento para a realização desta operação.

Estes filtros são óptimos segundo os critérios que estão na base da sua formulação e, no entanto, a imagem resultante da filtragem está longe de ser uma representação fidedigna e precisa das orlas de intensidade na imagem original. Em particular, os problemas que surgem na diferenciação de imagens (mencionados no ponto 1.1) subsistem, embora em menor escala, nas imagens resultantes da aplicação de qualquer um dos filtros para detecção de orlas de intensidade, quer seja numa configuração de detecção por máximos (como no caso do filtro de Deriche), quer seja numa configuração de detecção por passagens-por-zero. Por este motivo é comum realizar operações que validam e melhoram aquela representação (ver, por exemplo, [Chen, 1989]), algumas das quais se dá conta de seguida. O tipo de filtragem utilizada - detecção por máximos ou detecção por passagens-por-zero - determina a ou as operações aplicáveis. Nomeadamente, as operações de supressão de não-máximos e de limiarização só são aplicáveis a imagens resultantes de uma filtragem por detector de máximos de amplitude do gradiente, enquanto que a operação de validação de passagens-por-zero, como o nome indica, é aplicável a imagens de orlas detectadas por derivações de segunda ordem.

1.3.1 - Supressão de não-máximos

Esta operação tem por objectivo remover todos os pontos que, não sendo máximos de amplitude na resposta do filtro à imagem original, se situam nas suas imediações e apresentam ainda um nível elevado que poderá conduzir a uma localização imprecisa da orla correspondente. O que se pretende é realizar uma espécie de adelgaçamento direccionado segundo o gradiente de intensidade em cada ponto da imagem.

Note-se que a implementação do filtro de Deriche usada no trabalho desta dissertação inclui a supressão de não-máximos.

1.3.2. - Limiarização

A limiarização de uma imagem de orlas de intensidade destina-se a transformá-la numa representação gráfica ou mapa das orlas mais significativas presentes na imagem original, removendo-se todas aquelas que correspondem a pequenas transições de intensidade, vulgarmente associadas a ruído nos processos de aquisição e digitalização de imagens. Esta operação não está, todavia, isenta de

dificuldades. O ruído na imagem de amplitudes das orlas, após as operações de filtragem, dá origem a flutuações de amplitude ao longo de uma mesma orla, de tal modo que a escolha de um limiar global de valor elevado poderá levar à quebra dos contornos detectados. Por outro lado, a escolha de um limiar de valor reduzido aumenta a sensibilidade havendo por isso maior probabilidade de assinalar as orlas causadas pelo ruído.

1.3.3 - Validação das passagens por zero

No que diz respeito à detecção de orlas pelas passagens-por-zero de um operador de derivação de segunda ordem, há a ter em conta que surgem com frequência detecções falsas. Estas não têm nenhuma correspondência directa com variações nos sinais originais e as suas causas são principalmente duas:

- o ruído nos sinais ou imagens originais, introduzido possivelmente pelos dispositivos de captação, e que dão origem a desvios na resposta do operador;
- as variações no sinal ou imagem serem tais que, devido às suas interferências, surjam mínimos locais na amplitude da primeira derivada.

Torna-se assim necessário validar as passagens-por-zero que correspondem efectivamente às variações existentes e remover as passagens-por-zero que surgiram indevidamente.

2 - Seguimento de linhas e aproximação poligonal utilizando faixas

Nesta secção são apresentados dois métodos de aproximação poligonal de linhas utilizando faixas de aproximação: um, que utiliza faixas estáticas de aproximação, e um segundo que utiliza faixas dinâmicas.

Também é apresentado um método para o seguimento de linhas e conseqüente aproximação poligonal, utilizando faixas dinâmicas. Trata-se do método concebido e implementado no âmbito desta dissertação e que inclui, como se depreende da designação utilizada, o método de aproximação poligonal de linhas utilizando faixas dinâmicas; o método utiliza, para o seguimento das diversas linhas, critérios baseados na direcção e em dois níveis de amplitude. Considera-se que todos os *pixels* da imagem estão previamente classificados em termos de direcção e amplitude, como resultado dum detector de orlas de intensidade; a metodologia foi aplicada a imagens produzidas pelo filtro de Deriche.

2.1 - Métodos de aproximação poligonal de linhas utilizando faixas

Uma linha pode ser aproximada por uma sequência de segmentos de recta, de tal modo que a distância de qualquer ponto da linha ao segmento de recta de aproximação (que abranja o troço da linha em que se encontra o ponto) esteja dentro de uma tolerância especificada. Tal processo é útil na descrição de forma e na redução dos dados. Muitos métodos têm sido propostos; alguns, tentam minimizar o número de segmentos de recta de aproximação em detrimento do tempo dispendido em computação; outros, tentam minimizar o tempo dispendido, com menor cuidado no número de segmentos resultantes.

Roberge propôs um método rápido que, experimentalmente, se tem verificado ser o mais rápido de todos os propostos [Leung, 1990]. Este método tem um mecanismo interno para tratar contornos com o uso de faixas; contudo, gera demasiados segmentos de recta de aproximação, quando comparado com a solução óptima. Em alguns casos, gera mais do dobro, o que o torna numa solução não muito atraente. Roberge introduziu um factor de extensão, para reduzir o número de segmentos de recta resultantes, mas tal modificação é “*ad hoc*” e o referido factor não pode ser predeterminado.

De seguida, são apresentados dois métodos de aproximação poligonal utilizando faixas; o método proposto por Roberge e apresentado em [Leung, 1990], e o proposto por Leung e Yang em [Leung, 1990]. Este último é mais lento, contudo produz um número menor de segmentos de recta de aproximação.

2.1.1 - Método de aproximação poligonal de linhas utilizando faixas estáticas (Roberge)

Como representado na Fig. 10, uma faixa é definida por uma linha crítica e por duas linhas de fronteira.

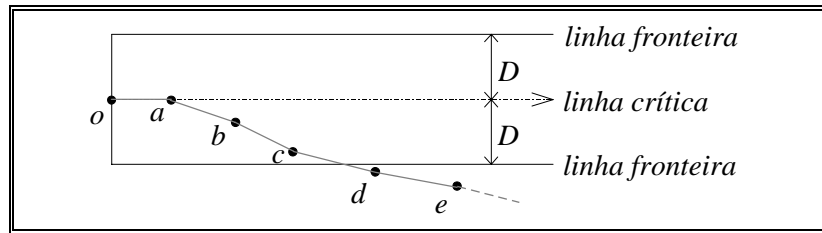


Fig. 10 - Definição de uma faixa de aproximação.

A linha crítica é definida por dois pontos de referência, nomeadamente, o primeiro e o segundo ponto (por exemplo, os pontos *o* e *a*, na Fig. 10) da linha que se pretende aproximar, e as duas linhas fronteira como sendo paralelas à linha crítica e a uma distância *D* desta. Estas duas linhas, formam uma faixa para restrição do processo de aproximação por um segmento de recta. A linha que se pretende aproximar é então percorrida ponto a ponto. O processo pára no primeiro ponto que sai da faixa (por exemplo, o ponto *d* na Fig. 10). Um segmento de recta de aproximação, é então definido pelos pontos *o* e *c*. Este último ponto, é novamente utilizado como ponto inicial de uma nova faixa de aproximação cuja linha crítica fica definida por ele e pelo ponto que saiu da faixa anterior (*d*, na Fig. 10).

O maior problema com este método dá-se quando o segundo ponto de referência está posicionado de tal forma que o terceiro ponto da linha fica fora da faixa de aproximação. O segmento de recta resultante será muito curto, o que geralmente não é o desejado. Um exemplo de tal problema é verificado na Fig. 11 pela faixa *I*. Pode-se observar, na mesma figura, que a faixa *II* é mais desejável, pois inclui mais pontos da linha que se pretende aproximar.

Roberge modificou o seu método de aproximação original, escolhendo o ponto *a* (o segundo ponto de referência) de tal forma que a sua distância ao primeiro ponto de referência seja pelo menos de $d \times EF$, onde *EF* é o factor de extensão que tem um valor igual ou superior a 1. A sua ideia baseava-se na observação de que quanto maior a distância do segundo ao primeiro ponto de referência, melhor se aproxima a direcção da linha crítica à da linha que se pretende aproximar. Contudo, o valor de *EF* não pode ser predeterminado, e nenhum método para a sua determinação foi proposto. Em vez disso, utilizou-se um valor arbitrário, o qual foi demonstrado fornecer geralmente resultados pobres. Como já foi referido, experimentalmente verifica-se que este algoritmo é o mais rápido, mas gera demasiados segmentos de linha [Leung, 1990]. Em alguns casos, gera mais do dobro da solução óptima, o que torna este método não muito atractivo.

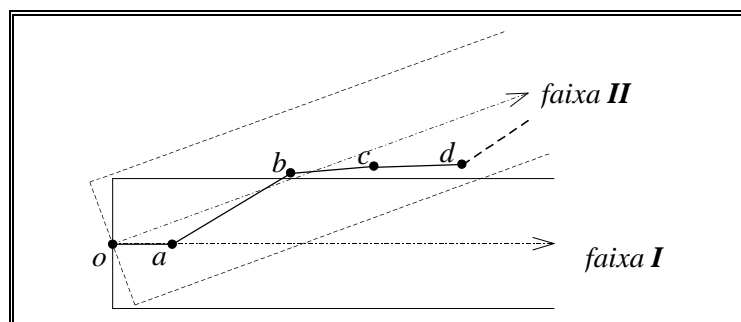


Fig. 11 - Uma má escolha e outra mais apropriada, para uma faixa de aproximação de uma dada linha.

2.1.2 - Método de aproximação poligonal de linhas utilizando faixas dinâmicas (Leung e Yang)

A ideia básica deste método de aproximação consiste em rodar a faixa de aproximação utilizando o primeiro ponto como *pivot*; de forma a incluir o maior número possível de pontos da linha que se

pretende aproximar. Um exemplo para ilustrar a vantagem deste método de aproximação pode ser observado na Fig. 12; nesta figura, o caso a) ilustra a melhor faixa de aproximação possível sem rotação desta e o caso b) ilustra a melhor faixa quando é permitida rotação.

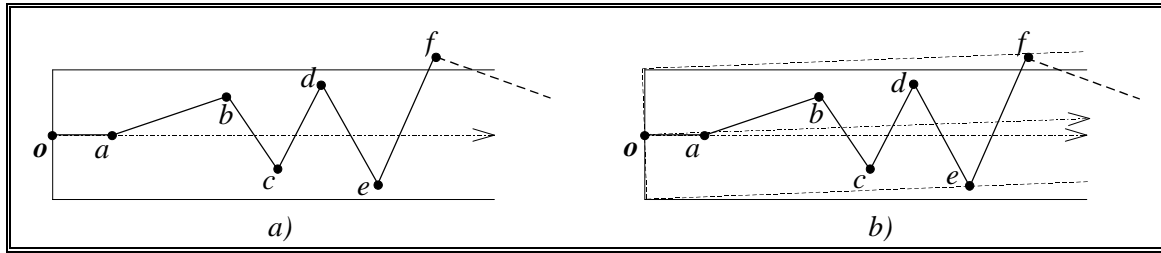


Fig. 12 - Melhor faixa de aproximação possível: a) sem rotação permitida e b) com rotação permitida.

A orientação da faixa de aproximação é o único parâmetro que varia neste método.

Com o intuito de permitir a rotação da faixa de aproximação, é necessário determinar o valor dos dois ângulos máximos de que a faixa pode ser rodada para cima e para baixo (*rotateUp* e *rotateDown*, respectivamente), sem excluir nenhum ponto que já se encontra no seu interior. Na Fig. 12a, os pontos *o*, *a*, *b*, *c*, *d*, e *e*, estão incluídos na faixa enquanto o ponto *f* não está. Para se incluir o ponto *f* na faixa, esta é rodada para cima de forma a não excluir nenhum dos pontos previamente incluídos do seu interior. Esta rotação pode ser realizada até que a linha de fronteira inferior toque o ponto *e*. Desta forma, a rotação da linha de fronteira inferior até ao ponto *e* define o máximo ângulo de rotação permitido para cima. Formalmente, o ângulo *rotateUp* é igual à soma dos ângulos *A* e *B*, como representado na Fig. 13. O ângulo *A*, que é sempre positivo, pode ser definido por $\sin^{-1}(D/\overline{oX})$. Por outro lado, o ângulo *B* pode ser definido como $\tan^{-1}(\overline{PX}/\overline{oP})$, podendo ser positivo ou negativo dependendo de \overline{PX} estar acima (ver Fig. 13a) ou abaixo (ver Fig. 13b) da linha crítica, respectivamente. O ângulo *rotateDown* pode ser calculado de forma semelhante.

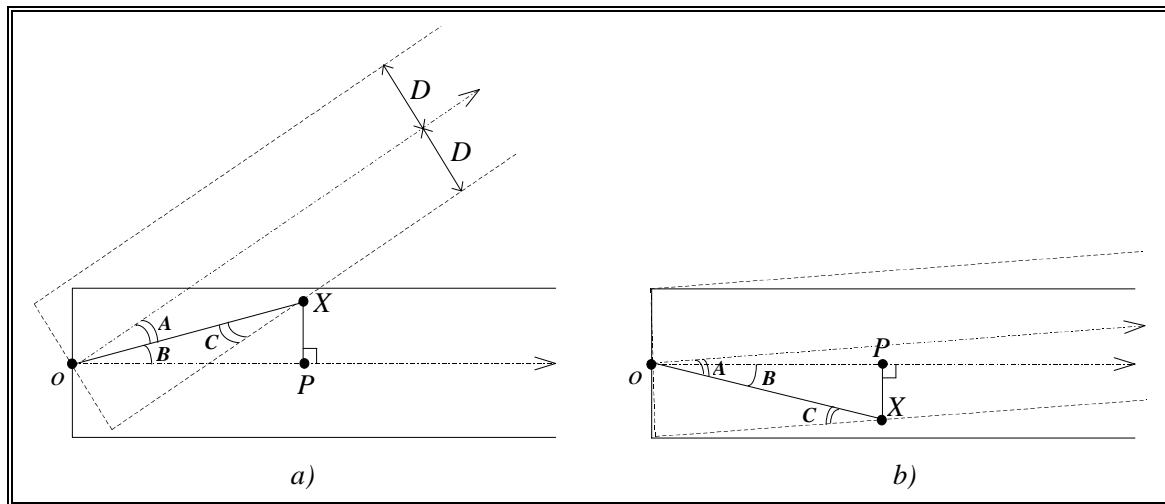


Fig. 13 - Ângulo de rotação permitido, para cima, da faixa de aproximação quando o ponto *X* está localizado na metade: a) superior e b) inferior, desta.

Quando surge um ponto que se apresenta acima da faixa de aproximação, o mínimo ângulo *rotateUp'* que esta deverá rodar de forma a incluí-lo pode ser definido formalmente como igual à diferença entre os ângulos *B* e *A*, Fig. 14a. Se este ângulo for inferior ao ângulo *rotateUp* até então determinado, a faixa pode ser rodada para cima com um ângulo igual a *rotateUp*. Depois desta rotação, o ângulo *rotateUp* é igual a zero, enquanto o ângulo *rotateDown* é igual a (*rotateUp* - *rotateUp'*), Fig. 14b.

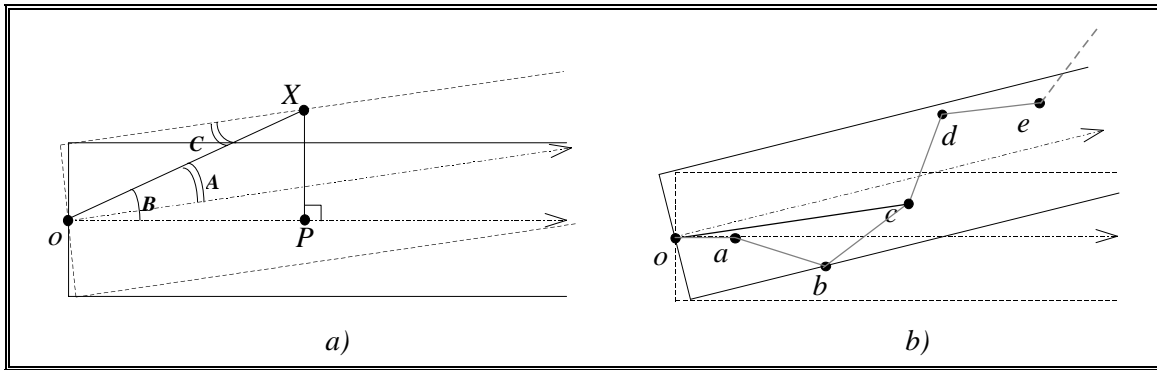


Fig. 14 - a) Ângulo que a faixa de aproximação deverá rodar para incluir o ponto X que se encontra acima da mesma. b) Rotação da faixa de aproximação de forma a incluir o ponto d .

Se, pelo contrário o referido ponto exterior não puder ser incluído, Fig. 15a, então um segmento de recta de aproximação é definido pelo primeiro e último ponto no interior da faixa, e uma nova faixa é definida por este último ponto e pelo ponto que não foi possível incluir, Fig. 15b.

As mesmas considerações são efectuadas quando surge um ponto que se apresenta abaixo da faixa de aproximação.

Pela observação dos casos apresentados, pode-se concluir que não é necessário determinar os ângulos *rotateDown* e *rotateUp* utilizando todos os pontos no interior da respectiva faixa. Assim, por exemplo, observando a Fig. 12a conclui-se que apenas é necessário considerar na determinação do ângulo *rotateDown* os pontos a , b e d , e da mesma forma, para o ângulo *rotateUp* são apenas considerados os pontos a , c e e . Deste modo, é possível aumentar-se consideravelmente a eficiência do método de aproximação.

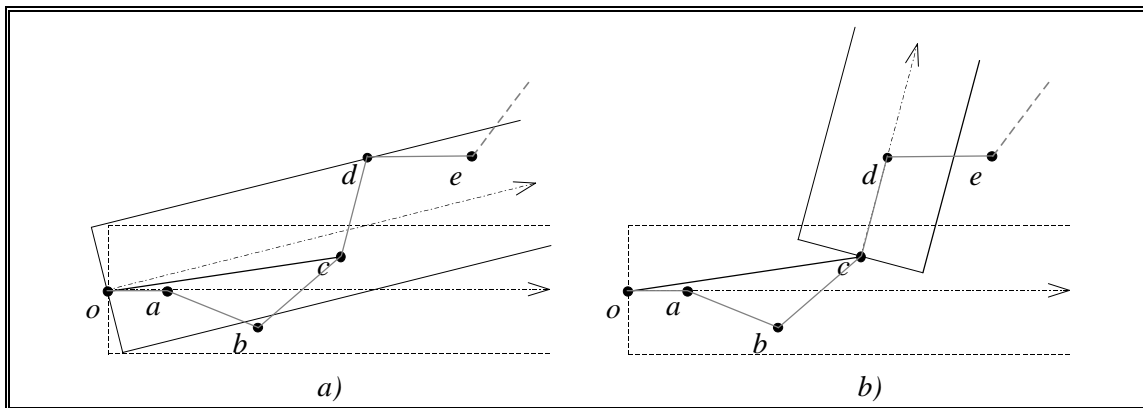


Fig. 15 - a) A rotação da faixa de aproximação de forma a incluir o ponto d não é possível. b) Definição de uma nova faixa de aproximação.

Um algoritmo para este método de aproximação por faixas dinâmicas pode ser:

- I) Considerar o primeiro ponto da linha a aproximar.
- II) Considerar o segundo ponto da mesma linha.
- III) Definir uma faixa de aproximação considerando o primeiro e o segundo ponto da referida linha, de forma a determinar a linha crítica da faixa.
- IV) Inicializar os ângulos *rotateUp* e *rotateDown*. Tal inicialização pode ser efectuada a partir da largura admissível para as faixas de aproximação.
- V) Considerar o próximo ponto da linha.
 - Se não existir mais nenhum ponto, então definir o segmento de recta de aproximação pelo primeiro e segundo ponto no interior da faixa de aproximação, e terminar.

- Se existir, então:
 - a) Determinar a distância deste ponto à linha crítica da faixa que está a ser considerada.
 - b) Se esta distância for inferior ou igual a metade da largura admissível para as faixas de aproximação, então este ponto pode ser incluído na faixa, e actualizam-se os ângulos *rotateUp* e *rotateDown* do seguinte modo:
 - Se o referido ponto se situa na parte superior da faixa, então é actualizado o ângulo *rotateDown*; se ainda não existir nenhum ponto na parte inferior da faixa, então o referido ponto é também utilizado para actualizar o ângulo *rotateUp*.
 - Se o referido ponto se situa na parte inferior da faixa, então é actualizado o ângulo *rotateUp*; se ainda não existir nenhum ponto na parte superior da faixa, então o referido ponto é também utilizado para actualizar o ângulo *rotateDown*.
 - c) Se não, então:
 - i) Se está localizado acima da faixa de aproximação, então determinar o ângulo *rotateUp*'.
 - ii) Caso este ângulo seja inferior ao ângulo *rotateUp*, então rodar a faixa para cima de um ângulo igual a *rotateUp*, passando este ângulo a ser igual a zero e o ângulo *rotateDown* igual a (*rotateUp-rotateUp*').
 - iii) Caso *rotateUp*' seja superior ao ângulo *rotateUp*, então este ponto não pode ser incluído na faixa de aproximação. Neste caso:
 - um segmento de recta de aproximação é definido pelo primeiro e pelo último ponto no interior da faixa de aproximação;
 - é definida uma nova faixa de aproximação, utilizando-se o último ponto no interior da última faixa considerada e este ponto;
 - voltar ao ponto *V*) deste algoritmo.
 - iv) Se está localizado abaixo da faixa de aproximação, então determinar o ângulo *rotateDown*'.
 - v) Caso este ângulo seja inferior ao ângulo *rotateDown*, então rodar a faixa para baixo de um ângulo igual a *rotateDown*, passando este ângulo a ser igual a zero e o ângulo *rotateUp* igual a (*rotateDown-rotateDown*').
 - vi) Caso *rotateDown*' seja superior ao ângulo *rotateDown*, então este ponto não pode ser incluído na faixa de aproximação. Neste caso:
 - um segmento de recta de aproximação é definido pelo primeiro e pelo último ponto no interior da faixa de aproximação;
 - é definida uma nova faixa de aproximação, utilizando-se o último ponto no interior da última faixa considerada e este ponto;
 - voltar ao ponto *V*) deste algoritmo.

2.2 - Um método para o seguimento de linhas e conseqüente aproximação poligonal utilizando faixas dinâmicas

Apresenta-se agora um método para o seguimento de linhas em imagens produzidas por um detector de orlas que determine a amplitude e direcção do gradiente. O método faz o seguimento dos máximos de amplitude com histerese², sob condições de consistência na direcção. No seguimento é integrada a

² Tal como é realizado no filtro de Canny para detecção de orlas de intensidade.

aproximação poligonal por faixas dinâmicas, o que permite dispensar o adelgaçamento dos resultados do seguimento. Deste método destacam-se as seguintes características:

- efectua o seguimento e a aproximação de todas as linhas presentes numa imagem;
- contempla o seguimento e a aproximação de linhas fechadas, permanecendo fechadas após a aproximação;
- contempla o seguimento e a aproximação de linhas abertas, com o seguimento iniciado num dos seus pontos extremos;
- faz o seguimento e a aproximação de linhas de espessura unitária e não unitária;
- o seguimento é baseado na direcção e em dois níveis de amplitude do gradiente; pressupõe-se, por isso, que os *pixels* da imagem se encontram previamente classificados por um detector de orlas de intensidade que aproxime a primeira derivada da função-imagem;
- a aproximação poligonal por faixas dinâmicas é embebida no seguimento, pelo que o método produz um conjunto de linhas poligonais de aproximação.

Para controlar o método de seguimento são utilizados os seguintes parâmetros:

- T_s , nível de amplitude do gradiente que um ponto deverá apresentar para ser considerado como seguramente pertencente a uma linha;
- T_p , inferior a T_s , nível de amplitude do gradiente que um ponto deverá apresentar para ser considerado como provavelmente pertencente a uma linha;
- d_e , diferença máxima de direcção do gradiente entre dois pontos adjacentes para serem considerados como pertencentes a uma mesma linha;
- w_m , largura máxima de uma janela quadrada no interior da qual deverá ser detectado o primeiro ponto de uma linha para se proceder ao fecho da sua aproximação poligonal. Esta detecção é realizada apenas no caso de a linha que está a ser considerada ser composta por $(w_m + 2)$ pontos ou mais. Deste modo o problema de fechar linhas de pequeno comprimento, cujas as aproximações poligonais poderiam ficar reduzidas a um curto segmento, é evitado.

O algoritmo geral pode ser descrito do seguinte modo:

- 1) Executar, ou continuar, o varrimento da imagem de entrada para determinar uma linha aberta presente nesta e que ainda não tenha sido considerada. Esta determinação consiste em procurar um ponto que apresente nível de amplitude não inferior a T_s , e apresente apenas um ponto adjacente com nível de amplitude também igual ou superior a T_s e cuja direcção não difira em relação à sua mais do que d_e .
- 2) Caso o varrimento anterior tenha terminado, então iniciar ou continuar um novo varrimento da imagem de entrada para determinar uma linha fechada presente nesta e que ainda não tenha sido considerada. Esta determinação é realizada procurando um ponto que apresente nível de amplitude não inferior a T_s . Determinar um ponto adjacente que apresente nível de amplitude também igual ou superior a T_s e cuja direcção relativamente ao primeiro não difira mais do que d_e . Caso não seja determinado nenhum ponto adjacente com tais características, anular³ o ponto determinado, pois trata-se de um ponto isolado, e continuar o varrimento.
- 3) Considerar o ponto adjacente ao primeiro detectado como o segundo ponto da respectiva linha.
- 4) Anular o primeiro ponto detectado. Caso se estejam a considerar linhas de espessura não unitária, anular todos os pontos adjacentes ao primeiro ponto detectado que apresentem nível

³ Neste algoritmo pretende-se designar com “anular” o tornar impossível que o(s) ponto(s) especificado(s) seja(m) novamente detectado(s).

de amplitude não inferior a T_s e uma diferença de direcção relativamente à sua não superior a d_e ; caso contrário, anular apenas o segundo ponto detectado.

- 5) Verificar se a presente linha é constituída por mais pontos. Tal verificação é realizada determinando se o segundo ponto tem pelo menos um vizinho que apresente nível de amplitude não inferior a T_s e uma diferença de direcção relativamente à sua não superior a d_e .
- 6) Caso a linha que está a ser seguida não apresente mais nenhum ponto, então define-se um segmento de recta pelo primeiro e segundo ponto da referida linha e continua-se o varrimento voltando a **1)** ou a **2)**, caso se estejam a considerar linhas abertas ou linhas fechadas, respectivamente.
- 7) Se existir mais algum ponto na linha, então o ponto inicial é etiquetado como *INICIAL* e o segundo ponto é considerado como *CORRENTE*.
- 8) Definir uma faixa dinâmica de aproximação, utilizando os pontos *INICIAL* e *CORRENTE*.
- 9) Continuar a aproximação poligonal utilizando faixas dinâmicas sendo o próximo ponto da respectiva linha determinado do seguinte modo:
 - a) Determinar o ponto adjacente do *CORRENTE* que apresente nível de amplitude não inferior a T_s , uma diferença de direcção relativamente a este não superior a d_e , e que melhor mantenha a direcção da linha que está a ser seguida (isto é, mantendo o mais possível a direcção definida pelo ponto anterior ao *CORRENTE* e por este).
 - b) Caso seja determinado o próximo ponto, considerá-lo como o ponto *CORRENTE* e continuar a aproximação poligonal saltando para **h)**.
 - c) Caso contrário, determinar o ponto adjacente do *CORRENTE* que apresente um nível de amplitude não inferior a T_p , uma diferença de direcção relativamente a este não superior a d_e e melhor mantenha a direcção da linha que está a ser seguida. (Deste modo, é possível continuar a seguir a linha, que poderá pontualmente apresentar pontos de menor valor de amplitude - seguimento com histerese.)
 - d) Caso seja determinado o próximo ponto, considerá-lo como *CORRENTE*, continuar a aproximação poligonal saltando para **h)**.
 - e) Caso não seja determinado o próximo ponto da linha e esta seja já constituída por um número de pontos igual ou superior a $(w_m + 2)$, então procurar, no interior duma janela quadrada de dimensão w_m centrada no ponto *CORRENTE*, o primeiro ponto determinado para a respectiva linha (isto é, aquele que foi classificado como *INICIAL*).
 - f) Caso seja detectado o ponto inicial da respectiva linha, então anular a sua presença na imagem original, e terminar a aproximação da referida linha, considerando este ponto como o final da referida linha e continuar o varrimento da imagem original voltando a **1)** ou a **2)**, caso estejam a ser consideradas linhas abertas ou linhas fechadas, respectivamente.
 - g) Caso contrário, anular o ponto *INICIAL*, e terminar a aproximação da linha; continuar o varrimento da imagem original voltando a **1)** ou a **2)**, caso se esteja a seguir e aproximar linhas abertas ou linhas fechadas, respectivamente.
 - h) Caso se estejam a considerar linhas de espessura não unitária, anular todos os pontos adjacentes do *CORRENTE* que apresentem um nível de amplitude não inferior a T_s (ou a T_p , caso a origem do salto seja **d)**) e uma diferença de direcção relativamente a este não superior a d_e .
 - i) Caso contrário, anular unicamente o próximo ponto determinado.
 - j) Continuar o seguimento e aproximação poligonal da respectiva linha voltando a **9)**.

3 - Fusão de segmentos de recta próximos e de direcções similares

Por muito cuidadosos que sejam os métodos de detecção de linhas, a mesma entidade física na imagem original poderá ser representada por mais do que uma linha; o problema mais frequente é o da interrupção de linhas, mas também pode ocorrer a detecção de segmentos disjuntos, total ou parcialmente sobrepostos, que correspondem à mesma orla física. Um passo adicional para aproximação às entidades originais consiste na ligação e fusão dos segmentos de linha que pertençam à mesma orla física. O critério mais seguro para decidir dessa presença consiste em combinar a consistência na direcção dos segmentos com a sua proximidade. Os segmentos que satisfaçam esse critério serão combinados num só, por um método de fusão. Com este método é então possível, por exemplo, reduzir o número de entidades que possuam características comuns presentes numa mesma imagem, e a obtenção de entidades com características mais fortes (tal como o comprimento, a direcção), etc.

Nesta secção é apresentado o método concebido para fusão de segmentos que pertencem a uma mesma entidade. Trata-se de um método bastante versátil, que é apropriado para a fusão de segmentos que sejam não sobrepostos, mas também para aqueles que sejam parcialmente ou totalmente sobrepostos. O método tem a principal vantagem, em relação a outros, de entrar em consideração com o comprimento de cada segmento para definir a direcção do segmento resultante da fusão, e também para definir a sua localização entre os segmentos a fundir.

3.1 - Métodos para fusão de segmentos de recta próximos e de direcções similares

Existem vários métodos para a fusão de dois segmentos de recta próximos e que possuem direcções aproximadamente iguais. Assim, pode-se optar pelo método proposto por Liu e Huang, descrito em [Hussien, 1993]. O princípio deste método está representado na Fig. 16, e consiste em definir o segmento resultante da fusão pelos dois pontos extremos dos segmentos que definem a maior distância possível entre si. Este método apresenta contudo três desvantagens, nomeadamente:

- para o caso de segmentos a fundir totalmente sobrepostos, este método é inadequado pois o segmento de fusão poderá coincidir com um dos segmentos a fundir, Fig. 17;
- para o caso de segmentos a fundir parcialmente sobrepostos, a fusão resultante poderá não ser a mais adequada, Fig. 18;
- a fusão não é devidamente influenciada pelos comprimentos dos segmentos a ligar, l_i e l_j .

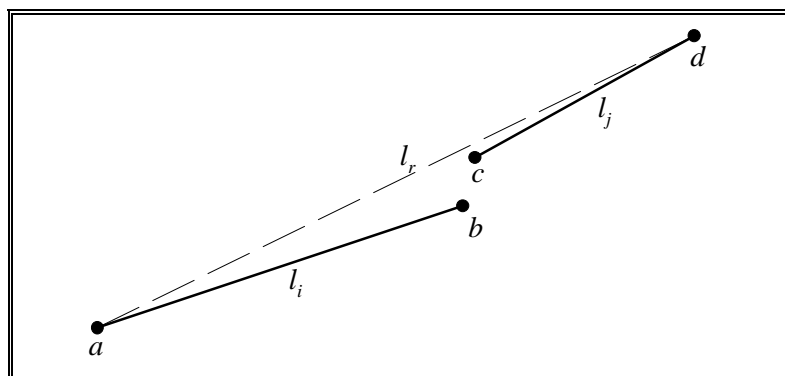


Fig. 16 - Fusão dos segmentos pelos dois pontos extremos que definem a maior distância possível entre si (método de Liu e Huang).

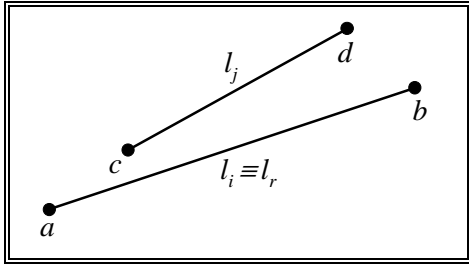


Fig. 17 - Fusão de dois segmentos que são totalmente sobrepostos pelo método proposto por Liu e Huang.

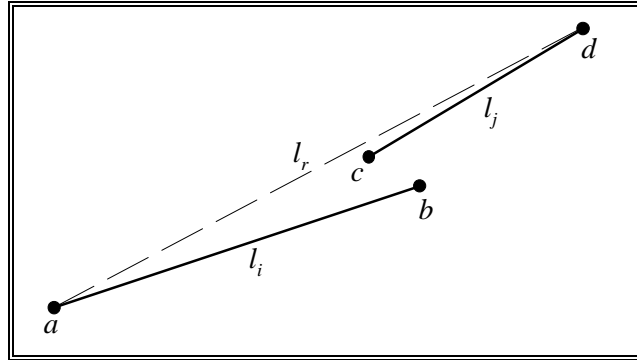


Fig. 18 - Fusão de dois segmentos que são parcialmente sobrepostos pelo método proposto por Liu e Huang.

Outro método possível para proceder à fusão de dois segmentos de recta próximos e que possuem direcções aproximadas, é proposto por Hussien e Sridhar em [Hussien, 1993]. Este método consiste em:

- 1) determinar as perpendiculares a cada segmento a fundir que passam pelos dois pontos extremos que definem a maior distância possível entre si;
- 2) definir o segmento de fusão resultante como passante pelos pontos médios de cada segmento a fundir e pela sua intersecção com as perpendiculares definidas anteriormente, Fig. 19.

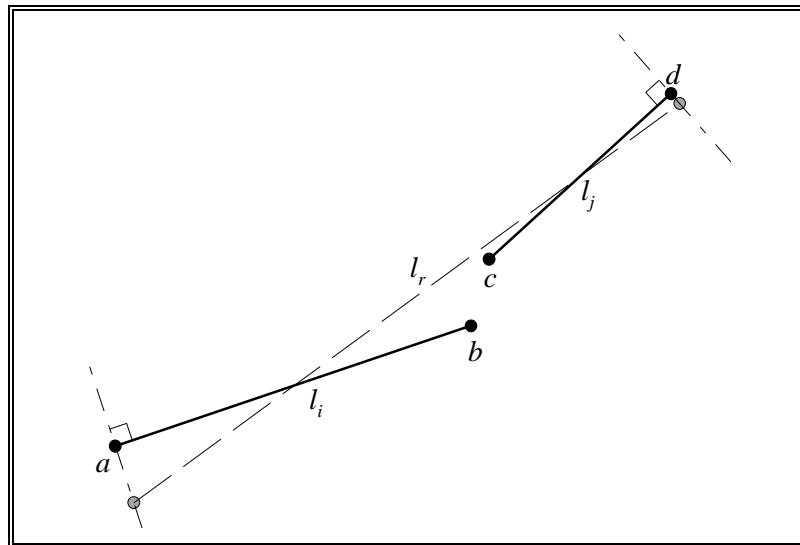


Fig. 19 - Segmento resultante da fusão pelo método proposto por Hussien e Sridhar.

Como é possível observar das Figs. 16 e 19, este segundo método apresenta resultados de melhor qualidade, quando comparado com o proposto por Liu e Huang. No entanto, este método apresenta as seguintes desvantagens:

- não é adequado para os casos de sobreposição parcial dos segmentos a fundir, Fig. 20;
- quando os dois segmentos a fundir são totalmente sobrepostos a sua utilização é bastante inadequada, Fig. 21;
- a fusão não é devidamente influenciada pelos comprimentos dos segmentos a fundir, l_i e l_j .

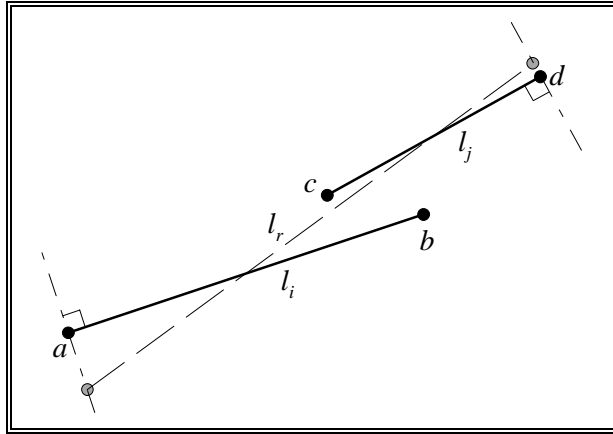


Fig. 20 - Segmento de fusão resultante pelo método proposto por Hussien e Sridhar para o caso de sobreposição parcial dos segmentos a fundir.

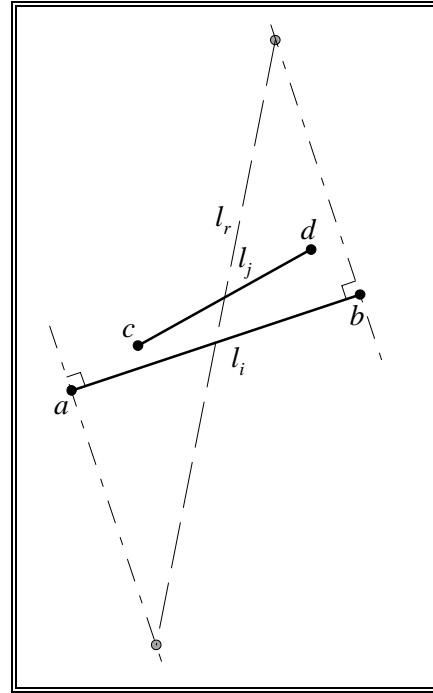


Fig. 21 - Segmento de fusão resultante pelo método proposto por Hussien e Sridhar para o caso dos segmentos a fundir serem totalmente sobrepostos.

Das figuras anteriores, conclui-se que ambos os métodos descritos não são de âmbito geral; isto é, não são aplicáveis do mesmo modo para os casos de não existência de sobreposição, existência de sobreposição total ou de sobreposição parcial; além disso, não consideram a influência dos comprimentos dos segmentos a fundir, l_i e l_j .

Assim, foi concebido um novo método para a fusão de segmentos. Este método foi implementado no âmbito desta dissertação e consiste em:

- 1) Definir as coordenadas do centróide (x_G, y_G) , do conjunto formado pelos quatro pontos extremos dos dois segmentos a fundir considerando os respectivos comprimentos l_i e l_j , como as “massas” pontuais a considerar:

$$x_G = \frac{l_i(a_x + b_x) + l_j(c_x + d_x)}{2(l_i + l_j)}, \quad y_G = \frac{l_i(a_y + b_y) + l_j(c_y + d_y)}{2(l_i + l_j)}, \quad (\text{Eqs. 4a e 4b})$$

onde (a, b) são os pontos extremos do segmento a ligar i e (c, d) são do segmento a ligar j .

Por este ponto, cuja localização é influenciada pelos comprimentos dos segmentos a fundir, deverá passar o segmento de fusão resultante.

- 2) Definir a direcção do segmento de fusão resultante como a média pesada das direcções dos segmentos a fundir, considerando os comprimentos destes como os respectivos pesos. Assim a direcção do segmento de fusão resultante θ_r , é definida como:

$$\text{Se } |\theta_i - \theta_j| \leq \frac{\pi}{2} \text{ então } \theta_r = \frac{l_i \theta_i + l_j \theta_j}{l_i + l_j}, \text{ se não } \theta_r = \frac{l_i \theta_i + l_j \left(\theta_j - \pi \frac{\theta_j}{|\theta_j|} \right)}{l_i + l_j}, \quad (\text{Eqs. 5a e 5b})$$

onde l_k e θ_k representam o comprimento e o ângulo de declive do segmento a fundir k , respectivamente.

- 3) Definir um referencial (X_G, Y_G) centrado no centróide, de coordenadas (x_G, y_G) , e com o eixo X_G paralelo à direcção θ_r do segmento de fusão.
- 4) Definir as coordenadas dos pontos extremos a, b, c e d , dos dois segmentos a fundir i e j , no referencial (X_G, Y_G) . A transformação de coordenadas⁴ respectiva é uma translação seguida de uma rotação:

$$\delta_{X_G} = (\delta_y - y_G)\sin\theta_r + (\delta_x - x_G)\cos\theta_r, \quad \delta_{Y_G} = (\delta_y - y_G)\cos\theta_r - (\delta_x - x_G)\sin\theta_r, \quad (\text{Eqs. 6a e 6b})$$

onde $(\delta_{X_G}, \delta_{Y_G})$ são as coordenadas do ponto δ no referencial (X_G, Y_G) , (x_G, y_G) são as coordenadas do centróide no referencial original e θ_r é o ângulo de declive do segmento de fusão.

- 5) As duas projecções mais afastadas, sobre o eixo X_G , dos quatro pontos extremos a, b, c e d , dos dois segmentos a fundir, definem os pontos extremos do segmento resultante da fusão.

Nas *Figs. 22, 23 e 24*, estão representadas as fusões resultantes pelo novo método proposto, nos casos de os segmentos a fundir serem não sobrepostos, parcialmente sobrepostos e totalmente sobrepostos, respectivamente.

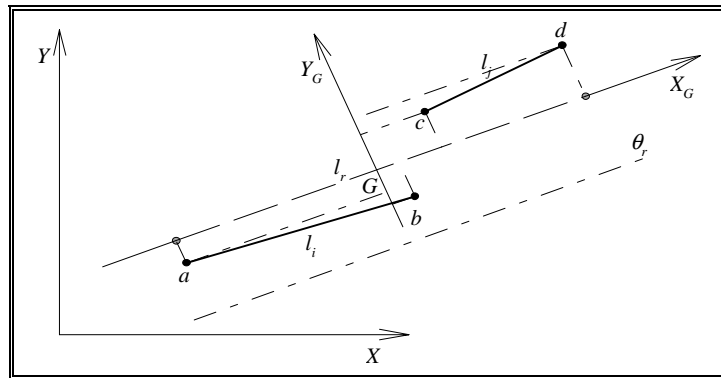


Fig. 22 - Fusão de dois segmentos que são não sobrepostos pelo novo método apresentado nesta secção.

Como é possível observar nas figuras anteriores, o novo método apresenta bons resultados quer os segmentos sejam totalmente sobrepostos, parcialmente sobrepostos ou não sobrepostos. No entanto, surge a necessidade de definir o conceito de proximidade dos dois segmentos a ligar para cada tipo de disposição relativa destes. Analisando as projecções dos quatro pontos extremos a, b, c e d , dos dois segmentos a ligar i e j , no referencial (X_G, Y_G) , torna-se bastante simples determinar se os segmentos estão ou não suficientemente próximos de modo a ser viável a sua fusão, assim como avaliar a disposição relativa destes. Desta forma, para determinar se dois segmentos são suficientemente próximos para se proceder à fusão e para conhecer o tipo de disposição entre os segmentos, pode-se comparar o comprimento do segmento de fusão resultante l_r com o dos segmentos definidos pelas projecções dos extremos a, b, c e d , no referencial (X_G, Y_G) . Deste modo, é possível a existência dos três casos seguintes:

⁴ Ver o anexo desta publicação: *Transformações Geométricas em 2D e em 3D*.

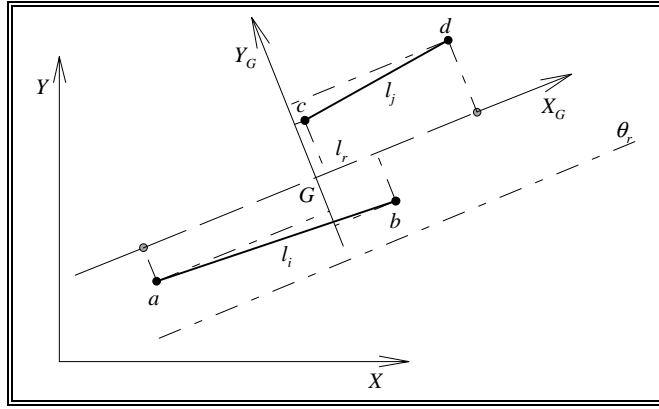


Fig. 23 - Fusão de dois segmentos que são parcialmente sobrepostos pelo novo método apresentado nesta secção.

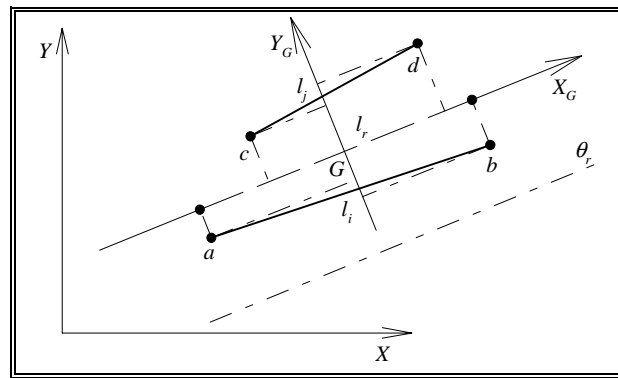


Fig. 24 - Fusão de dois segmentos que são totalmente sobrepostos pelo novo método apresentado nesta secção.

Caso 1:

$$\text{Se } |l_r| \geq \left(\left| \overline{a_{X_G} b_{X_G}} \right| + \left| \overline{c_{X_G} d_{X_G}} \right| \right), \quad (\text{Eq. 7})$$

então os segmentos em questão são não sobrepostos. Neste caso, o critério de proximidade para verificar se se deve ou não proceder à respectiva fusão é:

$$\left(\left| l_r \right| - \left| \overline{a_{X_G} b_{X_G}} \right| + \left| \overline{c_{X_G} d_{X_G}} \right| \right) \leq d_{MAX_{X_G}} \text{ e } \left| \overline{\delta_{Y_G} \kappa_{Y_G}} \right| \leq d_{MAX_{Y_G}}, \quad (\text{Eqs. 8a e 8b})$$

onde $d_{MAX_{X_G}}$ é a máxima distância admitida para se proceder à fusão, segundo o eixo X_G , δ_{Y_G} e κ_{Y_G} são as duas projecções mais afastadas sobre o eixo Y_G dos quatro pontos extremos a , b , c e d , e $d_{MAX_{Y_G}}$ é a máxima distância admitida para se proceder à fusão, segundo o eixo Y_G .

Caso 2:

$$\text{Se } |l_r| = \left| \overline{a_{X_G} b_{X_G}} \right| \text{ ou } |l_r| = \left| \overline{c_{X_G} d_{X_G}} \right|, \quad (\text{Eqs. 9a e 9b})$$

então os segmentos em questão são totalmente sobrepostos. Neste caso, o critério de proximidade para verificar se se deve ou não proceder à fusão é dado novamente pela Eq. 8b.

Caso 3:

$$\text{Se } |l_r| < \left(\left| \frac{a_{x_g} b_{x_g}}{c_{x_g} d_{x_g}} \right| + \left| \frac{c_{x_g} d_{x_g}}{a_{x_g} b_{x_g}} \right| \right) \quad (\text{Eq. 10})$$

e ambas as Eqs. 9a e 9b não são verificadas, então os segmentos em questão são parcialmente sobrepostos. Neste caso, o critério de proximidade para verificar se se deve ou não proceder à fusão é dado novamente pela Eq. 8b.

Das Eqs. 7 a 10 verifica-se que este método de fusão integra, de forma natural, os critérios para a determinação da disposição relativa dos segmentos de recta a fundir e, para cada tipo de disposição, as condições a respeitar para proceder à fusão.

Deve-se referir que, como facilmente é demonstrado, os resultados obtidos por este método de fusão são equivalentes aos obtidos realizando a regressão linear de todos os pontos dos segmentos de recta a fundir e, portanto, minimizando o erro quadrático da aproximação.

4 - Implementações desenvolvidas*

Nesta secção são apresentadas, de forma sucinta⁵, três implementações desenvolvidas no âmbito deste capítulo, para o seguimento de linhas e consequente aproximação poligonal utilizando faixas dinâmicas, e para a fusão de segmentos de recta próximos e de direcções similares. Estas implementações foram desenvolvidas em linguagem C em ambiente UNIX numa estação de trabalho SunSparc. As implementações integram-se no ambiente de processamento de imagem XITE⁶ do qual utilizam algumas funções e tipos de variáveis já previamente definidas. Assim, são apresentadas as seguintes três implementações:

- ✓ Uma implementação para o seguimento de linhas e consequente aproximação poligonal utilizando faixas dinâmicas: *dstrip*.
- ✓ Uma implementação para o desenho, numa imagem, dos segmentos de recta definidos num ficheiro: *showlines*.
- ✓ Uma implementação para a fusão de segmentos de recta próximos e de direcções similares: *linelink*.

O desenvolvimento da implementação *showlines* prendeu-se com o facto dos segmentos de recta a considerar pela implementação *linelink* serem introduzidos na forma dum ficheiro. Deste modo, a sua utilização torna-se desejável quando se pretende comparar a imagem obtida por *linelink* com a constituída pelo conjunto dos segmentos que possuem mais de um certo número de pontos e cujas coordenadas iniciais e finais estão definidas num dado ficheiro de entrada.

4.1 - Seguimento de linhas e consequente aproximação poligonal utilizando faixas dinâmicas: *dstrip*

Neste ponto, é apresentada de forma sumária uma implementação para o seguimento de linhas e consequente aproximação poligonal utilizando faixas dinâmicas, segundo o método descrito na secção

* Esta secção foi introduzida por razões de completude, mas não é essencial à compreensão das secções seguintes.

⁵ Em [Tavares, 1995] estas implementações estão descritas de forma mais detalhada.

⁶ X-based Image processing Tools an Environmment, [Lønnestad, 1992].

2. A referida implementação é constituída pelos módulos *principal*, *strip*, *find*, *finddir*, *closing* e *dda*. Apresenta-se de seguida a descrição de cada um destes módulos.

a) **Módulo *principal***, responsável pela interface implementação/utilizador. A seu cargo está:

- ✓ Leitura dos argumentos opcionais:
 - *title*, título da imagem de saída. Por defeito, é “*Dynamic Strip*”.
 - *filename*, nome do ficheiro para escrita dos resultados obtidos (coordenadas dos pontos que definem cada segmento de recta resultante). Por defeito é “*NULL*”, o que implica a não abertura do ficheiro.
 - *tsure*, valor mínimo de amplitude do gradiente para que um determinado ponto seja considerado como seguramente pertencente a uma dada linha. Por defeito, o seu valor é 100.
 - *tprobable*, valor mínimo de amplitude do gradiente para que um determinado ponto seja considerado como provavelmente pertencente a uma dada linha. Por defeito, o seu valor é 50.
 - *erdirection*, diferença máxima de direcção para um determinado ponto ser considerado como pertencente a uma dada linha. Por defeito, o seu valor é 5.
 - *dadm*, metade da largura da faixa utilizada para a aproximação poligonal, isto é, a tolerância de erro. Por defeito, o seu valor é 1.0.
 - *wmclose*, largura da janela quadrada no interior da qual deverá ser detectado o vizinho de um determinado ponto para fecho de uma dada linha que seja constituída por mais de (*wmclose* + 2) pontos. Por defeito, é igual a 3.
 - *IPixWid*, quando especificado como igual a 1 pelo utilizador, a implementação considera que a imagem de entrada contém linhas de espessura unitária. Por defeito, a implementação considera que a imagem de entrada contém linhas de espessura não unitária.
 - *linecolor*, valor que deverá ser atribuído aos pontos pertencentes aos segmentos de recta resultantes da aproximação na imagem de saída. Por defeito, o seu valor é 0, correspondente ao preto.
 - *backcolor*, valor que deverão ter todos os pontos na imagem de saída que não pertençam aos segmentos de recta resultantes da aproximação, isto é, os pontos que constituem o fundo da imagem. Por defeito, o seu valor é 255, correspondente ao branco.
- ✓ Leitura dos argumentos passados à implementação pela linha de comando:
 - *input*, nome da imagem de entrada.
 - *output*, nome da imagem de saída.
- ✓ Chamada do módulo *strip*.
- ✓ Comunicação da existência de erros em argumentos de entrada ou ao longo do decurso da execução.

Como restrições à utilização deste módulo têm-se:

- ⊗ A imagem de entrada tem de ser do tipo “*UNS_BYTE*”, com duas bandas de igual tamanho, resultantes dum detector de orlas de intensidade. Este detector, deverá classificar cada *pixel*, na primeira banda quanto à amplitude e, na segunda banda quanto à direcção do gradiente.
- ⊗ O valor para *wmclose* deverá ser ímpar.
- ⊗ *linecolor* e *backcolor* deverão ter valores compreendidos entre 0 e 255.

⊗ *tprobable* deverá ter um valor compreendido entre 2 e 255.

⊗ *tsure* deverá ter um valor igual ou superior ao valor de *tprobable* e menor ou igual a 255.

- b) **Módulo *strip***, responsável pelo seguimento e aproximação poligonal de linhas, segundo o método concebido no âmbito desta dissertação e descrito anteriormente na secção 2. Este módulo, durante a sua execução, utiliza os módulos *find*, *finddir*, *closing* e *dda*.
- c) **Módulo *find***, responsável pela determinação de um ponto adjacente a um dado ponto que apresenta um valor de amplitude superior a um determinado limiar e direcção o mais próxima possível à deste.
- d) **Módulo *finddir***, responsável pela determinação de um ponto adjacente a um dado ponto que apresenta um valor de amplitude superior a um determinado limiar, direcção aproximadamente igual à deste e que melhor mantém a direcção da linha que está a ser seguida.
- e) **Módulo *closing***, responsável pela localização no interior de uma janela quadrada de dimensão especificada e centrada num dado ponto, do ponto vizinho deste que deverá apresentar um valor especificado numa dada banda e seja o mais próximo possível.
- f) **Módulo *dda***, responsável pelo desenho na imagem de saída dos segmentos de recta resultantes do seguimento e aproximação poligonal, efectuada pelo módulo *strip*. Trata-se de uma implementação do algoritmo de Bresenham (também designado por versão meio ponto do segmento) que tem como grande vantagem a não utilização de multiplicações, de divisões e de funções de arredondamento [Foley, 1991]. Neste módulo, também é efectuada a escrita das coordenadas dos pontos que definem cada segmento de recta no ficheiro de resultados, caso seja pretendido pelo utilizador.

Na Fig. 25, estão representados os módulos que constituem esta implementação, assim como a relação existente entre estes.

4.2 - Desenho numa imagem dos segmentos de recta definidos num ficheiro: *showlines*

Neste ponto, é apresentada de forma sumária uma implementação que tem como principal objectivo o desenho numa imagem de saída dos segmentos de recta definidos num ficheiro de entrada. Algumas características desta implementação são:

- ✓ A possibilidade de eliminação de falsos segmentos de recta; isto é, segmentos que sejam constituídos por um número de pontos inferior a um dado valor, definido pelo utilizador, não são considerados.
- ✓ A possibilidade de saída dos resultados obtidos (as coordenadas dos pontos inicial e final, que definem cada segmento de recta) num ficheiro para futura utilização.

A referida implementação é constituída pelos módulos *principal*, *showl* e *dda*. Apresenta-se de seguida a descrição de cada um destes módulos.

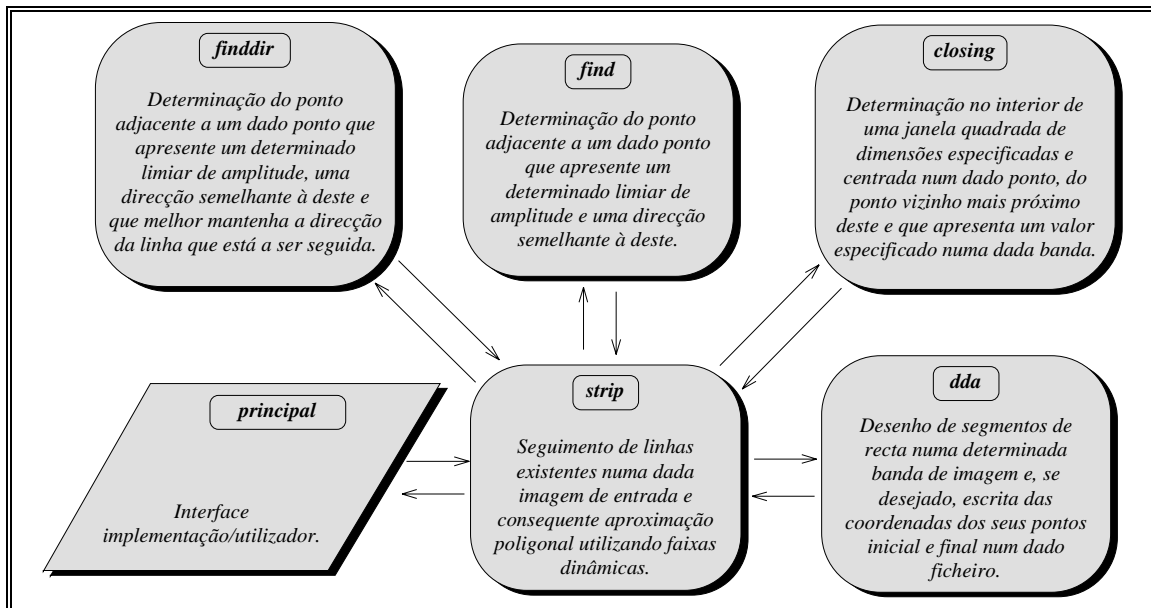


Fig. 25- Módulos integrantes da implementação *dstrip*.

a) **Módulo principal** responsável pela interface implementação/usuario. A seu cargo está:

- ✓ Leitura dos argumentos opcionais:
 - *title*, título da imagem de saída. Por defeito, é “*Lines in file*”.
 - *lineColor*, valor que deverá ser atribuído aos pontos pertencentes aos segmentos de recta determinados pela implementação. Por defeito, o seu valor é 0, correspondente ao preto.
 - *backColor*, valor que deverão ter todos os pontos na imagem de saída que não pertençam aos segmentos de recta determinados pela implementação, isto é, os pontos que constituem o fundo da imagem. Por defeito, o seu valor é 255, correspondente ao branco.
 - *xsize*, dimensão segundo *x* da imagem de saída. Por defeito, o seu valor é 256.
 - *ysize*, dimensão segundo *y* da imagem de saída. Por defeito, o seu valor é 256.
 - *nPixGoLine*, mínimo número de pontos que um dado segmento de recta deverá ter para não ser considerado como ruído. Por defeito, o seu valor é 20.
 - *outputFilename*, nome do ficheiro para escrita dos resultados obtidos (coordenadas dos pontos que definem cada segmento de recta determinado). Por defeito é “*NULL*”, o que implica a não abertura do ficheiro.
- ✓ Leitura dos argumentos passados à implementação pela linha de comando:
 - *inputFile*, nome do ficheiro de entrada que contém as coordenadas dos pontos inicial e final de cada segmento de recta a ser considerado.
 - *outputImage*, nome da imagem de saída que apresentará os segmentos de recta determinados pela implementação.
- ✓ Chamada do módulo *showl*.
- ✓ Comunicação da existência de erros em argumentos de entrada ou ao longo do decurso da execução.

Como restrições à utilização deste módulo, tem-se que *lineColor* e *backColor* deverão ter valores compreendidos entre 0 a 255.

b) **Módulo *showl*** responsável pela determinação dos segmentos de recta constituídos por mais de *nPixGoLine* pontos definidos no ficheiro de entrada, e consequente passagem das suas coordenadas iniciais e finais ao módulo *dda*.

c) **Módulo *dda*** responsável por:

- ✓ desenho dos segmentos de recta determinados pelo módulo *showl* na imagem de saída;
- ✓ se desejado pelo utilizador, escrita das coordenadas dos pontos inicial e final de cada segmento de recta determinado no ficheiro de saída.

Como já referido anteriormente, este módulo é uma implementação do algoritmo de Bresenham.

Na Fig. 26, estão representados os módulos que constituem esta implementação assim como a relação existente entre estes.

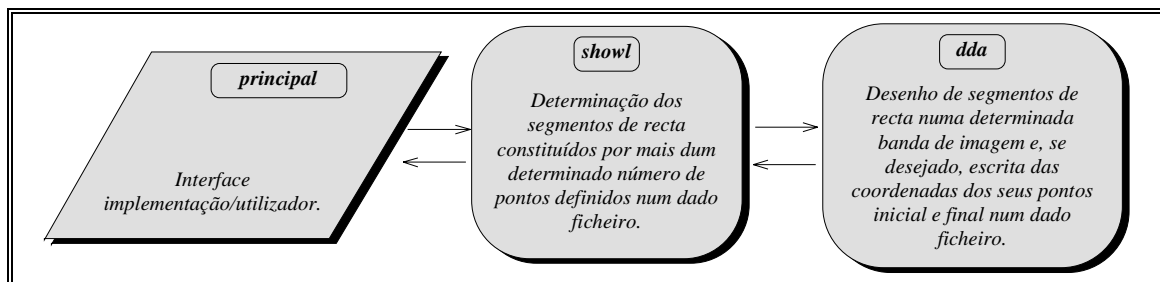


Fig. 26 - Módulos integrantes da implementação *showlines*.

4.3 - Fusão de segmentos de recta próximos e de direcções similares: *linelink*

Neste ponto, é apresentada de forma sumária uma implementação capaz de proceder à fusão de segmentos de recta próximos e de direcções similares, segundo o apresentado na secção anterior. Algumas características desta implementação são:

- ✓ A implementação procede em primeiro lugar à fusão dos segmentos de recta que sejam não sobrepostos; de seguida, procede à dos parcialmente sobrepostos; e por último, procede à dos totalmente sobrepostos; utiliza em cada fase os respectivos critérios de ligação.
- ✓ A ordem de fusão é imposta pelo comprimento dos segmentos de recta, sendo fundidos em primeiro lugar os de maior comprimento.
- ✓ A fusão de dois segmentos só é efectuada se nenhum dos dois já foi fundido com outro.
- ✓ A possibilidade de eliminação de falsos segmentos de recta, isto é, segmentos que sejam constituídos por um número de pontos inferior a um mínimo admissível, definido pelo utilizador.
- ✓ A possibilidade de saída dos resultados obtidos (as coordenadas dos pontos inicial e final de cada segmento de recta resultante da fusão) num ficheiro, para futura utilização.
- ✓ A possibilidade de utilização da implementação iterativamente, isto é, aplicar novamente a fusão a um conjunto de segmentos que já sofreram fusão.

A implementação é constituída pelos módulos: *principal*, *linking* e *dda*. Apresenta-se de seguida a descrição de cada um destes módulos.

a) **Módulo *principal***, responsável pela interface implementação/utilizador. A seu cargo está:

- ✓ Leitura dos argumentos opcionais:
 - *title*, título da imagem de saída. Por defeito, é “*Lines after linking*”.

- *lineColor*, valor que deverá ser atribuído, na imagem de saída, aos pontos pertencentes aos segmentos de recta resultantes da fusão. Por defeito, o seu valor é 0, correspondente ao preto.
 - *backColor*, valor que deverão ter todos os pontos, na imagem de saída, que não pertençam aos segmentos de recta resultantes da fusão, isto é, os pontos que constituem o fundo da imagem. Por defeito, o seu valor é 255, correspondente ao branco.
 - *xsize*, dimensão segundo x da imagem de saída. Por defeito, o seu valor é 256.
 - *ysize*, dimensão segundo y da imagem de saída. Por defeito, o seu valor é 256.
 - *maxDirDif*, máxima diferença de direcção entre dois segmentos de recta para que seja possível proceder à sua fusão. Por defeito, o seu valor é 5 grau.
 - *maxDist*, máxima distância segundo o eixo X_G entre dois segmentos de recta que sejam não sobrepostos, para que seja possível proceder à sua fusão. Por defeito, o seu valor é 8.
 - *maxWstrip*, máxima distância segundo o eixo Y_G entre dois segmentos de recta que sejam não sobrepostos para que seja possível proceder à sua fusão. Por defeito, o seu valor é 6.
 - *maxWstripParOverlap*, máxima distância segundo o eixo Y_G entre dois segmentos de recta que sejam parcialmente sobrepostos para que seja possível proceder à sua fusão. Por defeito, o seu valor é 4.
 - *maxWstripToOverlap*, máxima distância segundo o eixo Y_G entre dois segmentos de recta que sejam totalmente sobrepostos para que seja possível proceder à sua fusão. Por defeito, o seu valor é 2.
 - *nPixGoLine*, mínimo número de pontos que um dado segmento de recta deverá ter para não ser considerado como ruído. Por defeito, o seu valor é 20.
 - *outputFilename*, nome do ficheiro para escrita dos resultados obtidos (coordenadas dos pontos que definem cada segmento de recta resultante) e descrição de como foram obtidos. Por defeito é "NULL", o que implica a não abertura do ficheiro.
- ✓ Leitura dos argumentos passados à implementação pela linha de comando:
- *inputFile*, nome do ficheiro de entrada que contém as coordenadas dos pontos inicial e final de cada segmento de recta a ser considerado pela implementação.
 - *outputImage*, nome da imagem de saída que apresentará os segmentos de recta obtidos após a respectiva fusão.
- ✓ Chamada do módulo *linking*.
- ✓ Comunicação da existência de erros em argumentos de entrada ou ao longo do decurso da execução.

Como restrições à utilização deste módulo tem-se que *lineColor* e *backColor* deverão ter valores compreendidos entre 0 a 255.

b) **Módulo *linking***, responsável pelo processo de fusão de segmentos de recta próximos, de direcções similares e constituídos pelo menos por um determinado número de pontos, segundo o método apresentado na secção anterior. Para desenho na imagem de saída, e escrita no ficheiro de saída (quando desejado pelo utilizador) das coordenadas dos pontos iniciais e finais dos segmentos de recta resultantes, este módulo utiliza o módulo *dda*.

Este módulo tem como única restrição quanto à sua utilização a existência de memória suficiente.

c) **Módulo *dda***, responsável por:

- ✓ desenho dos segmentos de recta resultantes do módulo **linking** na imagem de saída;
- ✓ se desejado pelo utilizador, escrita das coordenadas dos pontos inicial e final de cada segmento de recta resultante no ficheiro de saída.

Como já anteriormente referido, este módulo é uma implementação do algoritmo de Bresenham.

Na Fig. 27, estão representados os módulos que constituem esta implementação, assim como a relação existente entre estes.

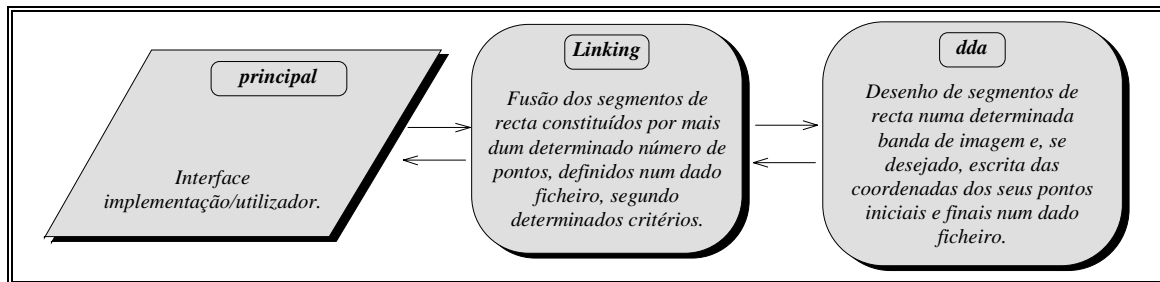


Fig. 27 - Módulos integrantes da implementação **linelink**.

5 - Alguns resultados experimentais obtidos pelas implementações desenvolvidas

Nesta secção, são apresentados alguns resultados experimentais⁷ obtidos pelas várias implementações descritas de forma sumária na secção anterior. Assim serão apresentados:

- a) resultados experimentais obtidos na utilização da implementação **dstrip** para o seguimento de linhas e consequente aproximação poligonal utilizando faixas dinâmicas;
- b) resultados experimentais obtidos na utilização da implementação **showlines** para o desenho numa imagem dos segmentos de recta definidos num ficheiro;
- c) resultados experimentais obtidos da implementação **linelink** para a fusão de segmentos de recta próximos e de direcções similares.

5.1 - Utilizando a implementação **dstrip** para o seguimento de linhas e consequente aproximação poligonal utilizando faixas dinâmicas

Neste ponto são apresentados alguns resultados experimentais obtidos pela implementação **dstrip** apresentada no ponto 4.1. A imagem de teste utilizada está representada na Fig. 28. Trata-se de uma imagem de dimensões 256×256 de 8 bits que posteriormente foi submetida a um detector de orlas de intensidade, nomeadamente ao detector de Deriche. Nas Figs. 29 e 30, estão representadas as duas bandas resultantes (previamente negadas, para uma mais fácil visualização). Assim, na Fig. 29 está representada a banda que contém os valores de amplitude para cada *pixel*, e na Fig. 30 está representada a banda que contém os valores de direcção para cada *pixel*. Como é possível observar das Fig. 29 e 30, as linhas que as constituem não têm necessariamente espessura unitária, tratando-se assim de um teste bastante exigente à implementação **dstrip**.

Nas Figs. 31, 32, 33, 34, 35, 36 e 37, estão representadas as imagens obtidas pela implementação **dstrip** com os parâmetros: *isure* igual a 50, *tprobable* igual a 40, *erdirection* igual a 10 e *dadm* igual a 0.1, 0.5, 1.0, 1.5, 2.0, 3.0 e 4.0, respectivamente. Os restantes parâmetros assumiram os seus valores por defeito. Na Fig. 38 é possível visualizar a influência do parâmetro *dadm* na aproximação poligonal obtida.

Na Fig. 39 está representada a soma da imagem de teste, representada na Fig. 28, com a imagem obtida pela implementação **dstrip** representada na Fig. 37; nesta figura, é possível verificar-se a

⁷ No capítulo V desta dissertação são apresentados outros resultados obtidos por estas implementações.

simplificação atingida por esta implementação quando o parâmetro *dadm* tem um valor já consideravelmente elevado.



Fig. 28 - Imagem de teste da implementação *dstrip*.



Fig. 29 - Primeira banda (amplitude) obtida pelo detector de Deriche a partir da imagem de teste.



Fig. 30 - Segunda banda (direcção) obtida pelo detector de Deriche a partir da imagem de teste.



Fig. 31 - Imagem obtida com os parâmetros: *tsure* igual a 50, *tprobable* igual a 40, *erdirection* igual a 10 e *dadm* igual a 0.1.

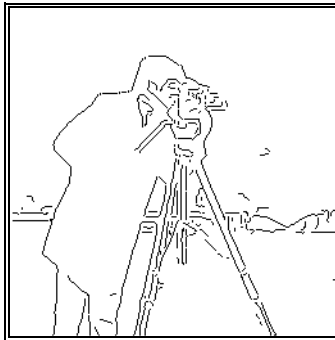


Fig. 32 - ... *dadm* igual a 0.5.

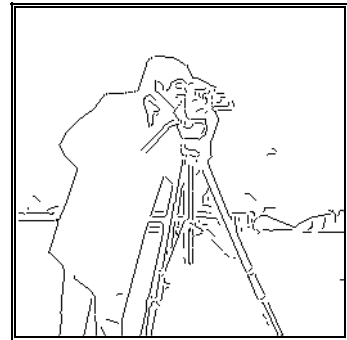


Fig. 33 - ... *dadm* igual a 1.0.

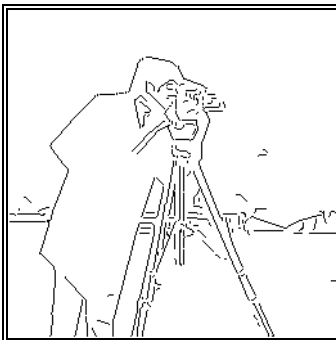


Fig. 34 - ... *dadm* igual a 1.5.

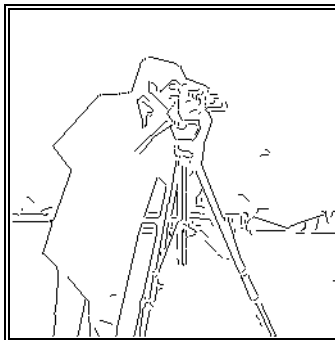


Fig. 35 - ... *dadm* igual a 2.0.

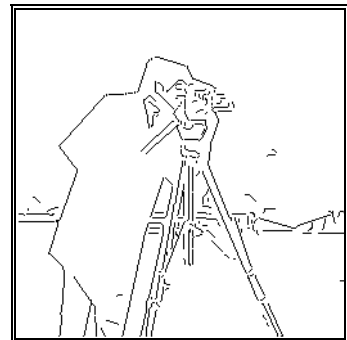


Fig. 36 - ... *dadm* igual a 3.0.

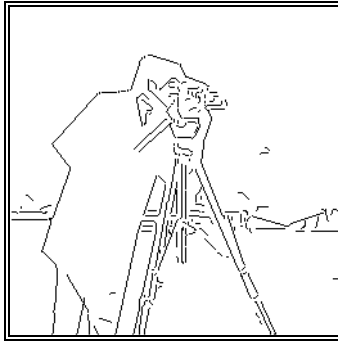


Fig. 37 - ... *dadm* igual a 4.0.

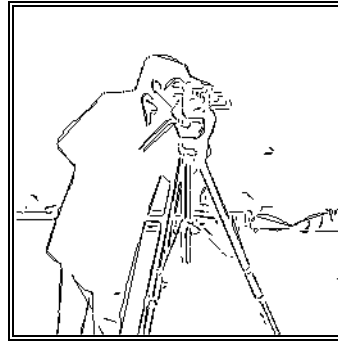


Fig. 38 - Soma da imagem representada na Fig. 31 com a representada na Fig. 37.



Fig. 39 - Soma da imagem representada na Fig. 28 com a representada na Fig. 37

Com o objectivo de demonstrar a influência dos parâmetros *tsure* e *tprobable* no seguimento das entidades que constituem a imagem original, está representada na Fig. 40 a imagem obtida com estes parâmetros iguais a 50 e a 40, e na Fig. 41 a imagem obtida com estes parâmetros iguais a 30 e a 10, respectivamente. Na obtenção destas imagens utilizaram-se os parâmetros *dadm* igual a 1.0, *erdirection* igual a 10, e os restantes com os respectivos valores por defeito.

Com o objectivo de demonstrar a influência do parâmetro *erdirection*, está representado na Fig. 42 a imagem obtida com este parâmetro igual a 5 grau e os restantes coincidentes com os utilizados na obtenção da imagem representada na Fig. 41.

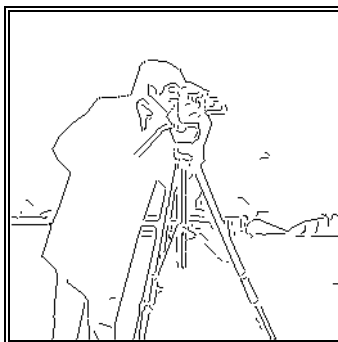


Fig. 40 - Imagem obtida com os parâmetros: *tsure* igual a 50, *tprobable* igual a 40, *erdirection* igual a 10 e *dadm* igual a 1.0.

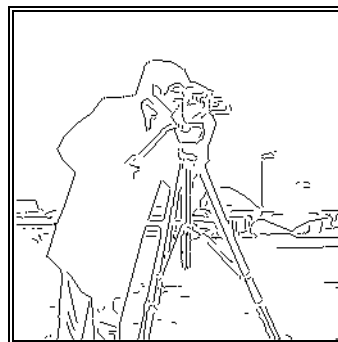


Fig. 41 - Imagem obtida com os parâmetros: *tsure* igual a 30, *tprobable* igual a 10, *erdirection* igual a 10 e *dadm* igual a 1.0.

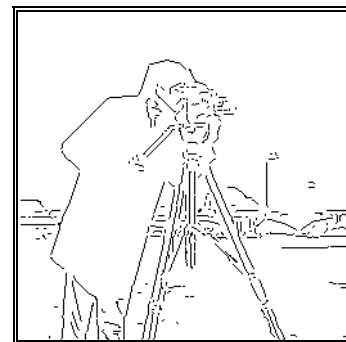


Fig. 42 - Imagem obtida com os parâmetros: *tsure* igual a 30, *tprobable* igual a 10, *erdirection* igual a 5 e *dadm* igual a 1.0.

Na Fig. 43, está representado o número de segmentos de recta obtidos pela aproximação poligonal versus o parâmetro *dadm*, com os parâmetros: *tsure* igual a 50, *tprobable* igual a 40, *erdirection* igual a 10 e os restantes assumindo os respectivos valores por defeito.

Na Fig. 44, está representado, a título de exemplo, parte do conteúdo do ficheiro de saída obtido pela implementação *dstrip* para os parâmetros: *dadm* igual a 4.0, *tsure* igual a 50, *tprobable* igual a 40, *erdirection* igual a 10 e os restantes assumindo os respectivos valores por defeito.

Das figuras apresentadas pode-se concluir que a implementação *dstrip* obtém bons resultados, mesmo para imagens bastante complexas como é o caso da imagem de teste utilizada, sendo adequada às situações nas quais se pretende reduzir o número de entidades presentes numa imagem sem, contudo, reduzir de forma drástica a informação contida. Como é lógico, o número de segmentos de recta de aproximação obtidos é tanto menor quanto maior é o parâmetro *dadm*, afastando-se, no entanto, cada vez mais a imagem obtida da original. Como é possível observar na Fig. 43, este decréscimo do número de segmentos de recta obtidos não é linear. Assim, o decréscimo é bastante acentuado para valores pequenos do parâmetro *dadm* (no caso apresentado, até *dadm* igual a 1.0) e depois é muito baixo.

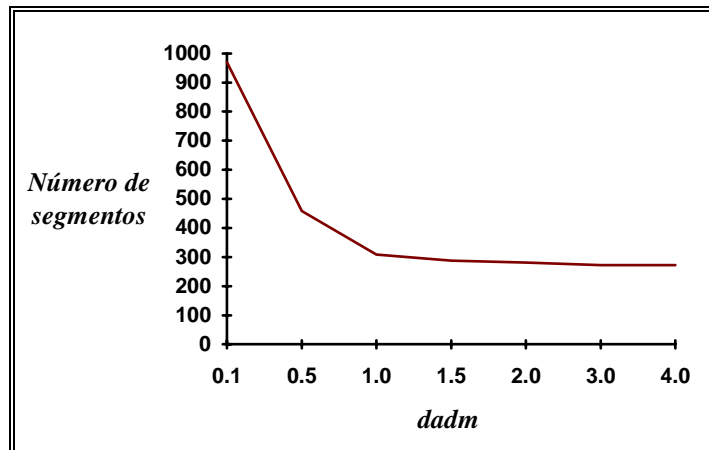


Fig. 43 - Número de segmentos de recta obtidos pela aproximação *dstrip* versus o parâmetro *dadm*.

```

Results for dstrip in camera_d.img with:
Sure threshold level for amplitude: 50.
Probable threshold level for amplitude: 40.
Error tolerance for direction: 10.
Error tolerance for distance: 4.
Width of mask for close line: 5.

101 43 94 65
133 45 137 54
137 55 153 61
127 56 118 64
...
102 226 101 228
35 234 35 236
34 237 34 238
    
```

Fig. 44 - Parte do conteúdo de ficheiro de saída obtido para os parâmetros: *dadm* igual a 4.0, *tsure* igual a 50, *tprobable* igual a 40, *erdirection* igual a 10 e *erdirection* igual a 10.

5.2 - Utilizando a implementação *showlines* para o desenho numa imagem dos segmentos de recta definidos num ficheiro

Neste ponto é apresentado um exemplo de resultados obtidos pela implementação *showlines*, apresentada no ponto 4.2. Assim, na Fig. 45, é representada a imagem constituída pelos segmentos de recta que se encontram definidos num dado ficheiro de entrada pelos seus pontos iniciais e finais, representado em parte na Fig. 46; na Fig. 47 é representada a imagem que se obteve como resultado da implementação *showlines*, com o parâmetro *nPixGoLine* igual a 20. Desta forma, esta imagem é constituída apenas por segmentos de recta formados por 20 pontos ou mais.

Como se depreende do resultado apresentado, uma aplicação desta implementação, além da já referida de representar os segmentos de recta definidos num dado ficheiro que irão posteriormente ser fundidos pela aplicação *linelink*, poderá ser a remoção de falsos segmentos - por exemplo, devidos ao ruído presente - definidos no ficheiro de entrada.

5.3 - Utilizando a implementação *linelink* para a fusão de segmentos de recta próximos e de direcções similares

Neste ponto, são apresentados alguns resultados experimentais obtidos pela implementação *linelink* apresentada no ponto 4.3. Os segmentos de recta para teste da referida implementação foram os resultantes da aplicação da implementação *dstrip* à imagem representada na Fig. 48 e estão representados na Fig. 49. Da imagem representada na Fig. 49 consideraram-se apenas os segmentos de recta formados pelo menos por 20 pontos, Fig. 50 (imagem obtida utilizando-se para tal a

implementação *showlines*). Os segmentos de recta desta última figura serão sujeitos à implementação *linelink*.



Fig. 45 - Imagem constituída pelos segmentos de recta que se encontram definidos num dado ficheiro de entrada.

101 43	90 66
90 66	66 71
66 71	33 107
133 45	137 54
137 55	142 56
127 56	118 63
...	
35 234	35 236
34 237	35 238
53 248	52 255

Fig. 46 - Parte do conteúdo do ficheiro de entrada que define os segmentos de recta a considerar.

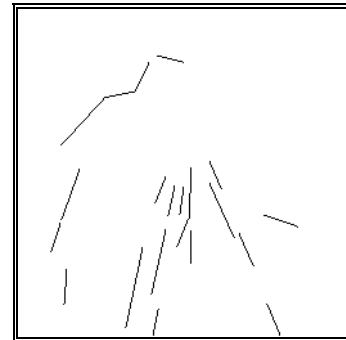


Fig. 47 - Imagem constituída por segmentos de recta formados por 20 pontos ou mais.



Fig. 48 - Imagem sujeita à implementação *dstrip*.

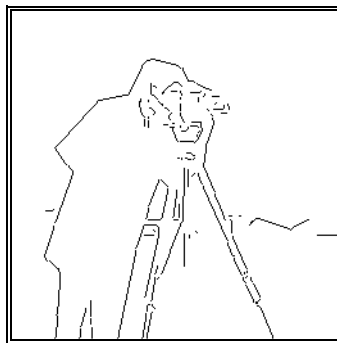


Fig. 49 - Imagem obtida pela implementação *dstrip*.

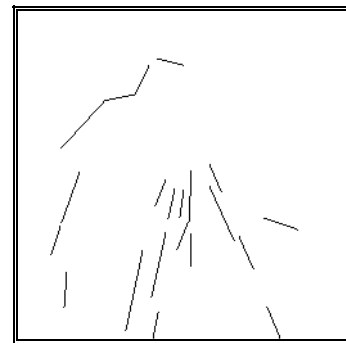


Fig. 50 - Imagem constituída por segmentos de recta formados pelo menos por 20 pontos.

Nas Figs. 51, 52, 53, e 54, são representadas as imagens obtidas pela implementação *linelink* a partir do ficheiro constituído pelos segmentos de recta visíveis na imagem da Fig. 50 com, respectivamente: todos os parâmetros assumindo os respectivos valores por defeito; parâmetro *maxWstripParOverlap* igual a 8 e os restantes assumindo os respectivos valores por defeito; parâmetro *maxDirDif* igual a 30, *maxWstrip* igual a 10, *maxWstripParOverlap* igual a 8 e os restantes assumindo os respectivos valores por defeito; parâmetro *maxDirDif* igual a 30, *maxWstrip* igual a 10, *maxWstripParOverlap* igual a 25 e os restantes assumindo os respectivos valores por defeito.

Na Fig. 55, está representada a imagem resultante da soma das imagens representadas nas Fig. 50 e 54.

O conteúdo do ficheiro constituído pelos segmentos de recta visíveis na imagem representada na Fig. 54 foi posteriormente submetido à implementação *linelink* com os parâmetros: *maxDirDif* igual a 40, *maxDist* igual a 12, *maxWstrip* igual a 15, *maxWstripParOverlap* igual a 10 e os restantes assumindo os respectivos valores por defeito. Nas Figs. 56, 57 e 58, são representadas: a imagem resultante; a soma da imagem resultante com a imagem equivalente ao conteúdo do ficheiro de entrada (Fig. 54); a soma da imagem resultante com a representada na Fig. 50.

O conteúdo do ficheiro constituído pelos segmentos de recta visíveis na Fig. 56 foi posteriormente submetido à implementação *linelink* com os parâmetros *maxDirDif* igual a 35, *maxWstrip* igual a 20, *maxWstripParOverlap* igual a 10 e os restantes assumindo os respectivos valores por defeito. Nas Figs. 59, 60 e 61, são representadas: a imagem resultante; a soma da imagem resultante com a representada na Fig. 56; a soma da imagem resultante com a representada na Fig. 50.

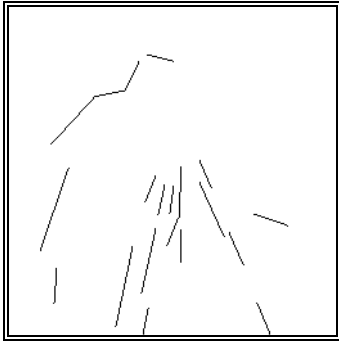


Fig. 51 - Imagem obtida com todos os parâmetros assumindo os respectivos valores por defeito.

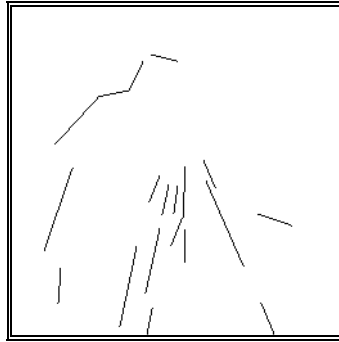


Fig. 52 - Imagem obtida com o parâmetro $\text{maxWstripParOverlap}$ igual a 8.

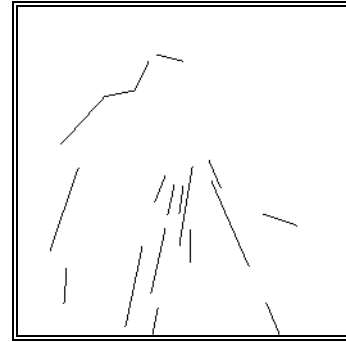


Fig. 53 - Imagem obtida com os parâmetros: maxDirDif igual a 30, maxWstrip igual a 10 e $\text{maxWstripParOverlap}$ igual a 8.

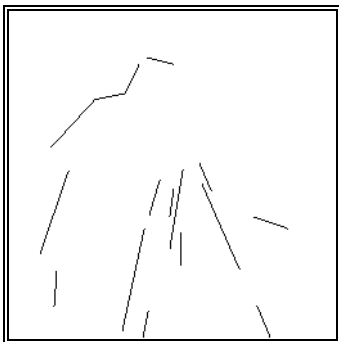


Fig. 54 - Imagem obtida com os parâmetros: maxDirDif igual a 30, maxWstrip igual a 10 e $\text{maxWstripParOverlap}$ igual a 25

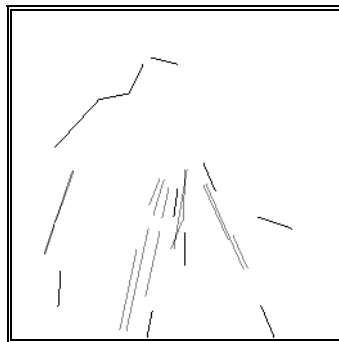


Fig. 55 - Soma das imagens representadas na Figs. 50 e 54.

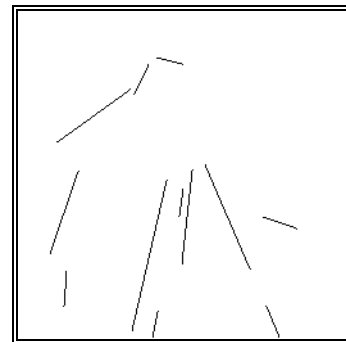


Fig. 56 - Imagem resultante a partir dos segmentos visíveis na Fig. 54 com os parâmetros: maxDirDif igual a 40, maxDist igual a 12, maxWstrip igual a 15 e $\text{maxWstripParOverlap}$ igual a 10.



Fig. 57 - Soma da imagem representada na Fig. 54 com a representada na Fig. 56.

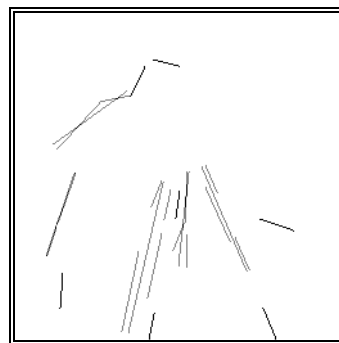


Fig. 58 - Soma da imagem representada na Fig. 50 com a representada na Fig. 56.

A título de exemplo, é representado na Fig. 62 o conteúdo dum ficheiro resultante da implementação *linelink*.

Como é possível deprender, a partir dos resultados experimentais apresentados, a implementação *linelink* é de utilização bastante flexível, sendo possível, por exemplo, numa primeira fase fundir apenas os segmentos de recta que não sejam sobrepostos, numa segunda fase fundir apenas os que sejam parcialmente sobrepostos e numa terceira e última fase, fundir apenas os que sejam totalmente sobrepostos. Independentemente do tipo de distribuição relativa dos segmentos de recta a fundir, a

implementação *linelink*, com os parâmetros adequados à respectiva situação, obtém resultados de boa qualidade.

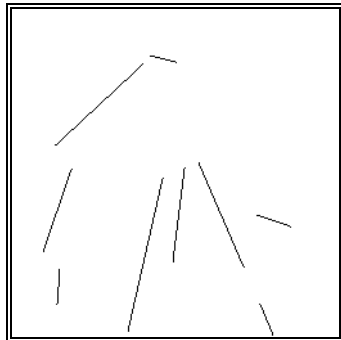


Fig. 59 - Imagem obtida a partir dos segmentos de recta visíveis na Fig. 56 com os parâmetros: *maxDirDif* igual a 35, *maxWstrip* igual a 20 e *maxWstripParOverlap* igual a 10.

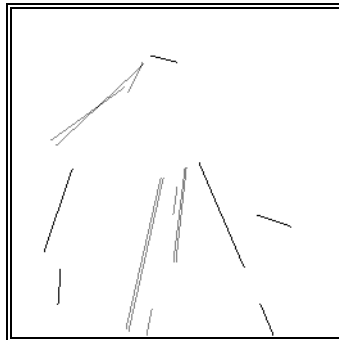


Fig. 60 - Soma da imagem representada na Fig. 59 com a representada na Fig. 56.

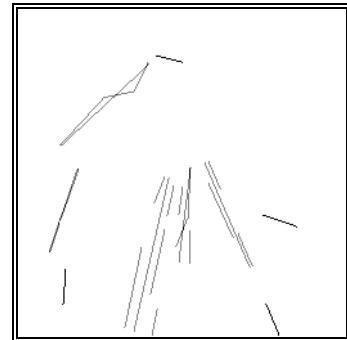


Fig. 61 - Soma da imagem representada na Fig. 59 com a representada na Fig. 50.

```

Results for linelink in test.txt with:
Maximum difference in direction for linking: 40
Maximum length of strip for linking without overlap: 12
Maximum width of strip for linking without overlap: 15
Maximum width of strip for linking with partially overlap: 10
Maximum width of strip for linking with total overlap: 2
Minimum number of pixels to be considered as a line: 20.

89 250 116 133
30 103 87 62
146 121 181 202
128 198 136 125
25 190 47 126
191 162 217 171
37 204 35 231
203 255 193 231
101 43 90 66
128 140 125 161
108 38 128 43
105 255 109 235
    
```

Fig. 62 - Conteúdo dum ficheiro resultante da implementação *linelink*. (Equivalente à Fig. 56)

6 - Algumas conclusões finais

Neste capítulo aborda-se de forma introdutória o problema da detecção de orlas de intensidade em imagens. Apresentam-se os operadores de diferenciação numérica - nomeadamente, o Gradiente e o Laplaciano - e a sua aplicação a imagens em tons de cinzento como método de detecção de transições acentuadas de intensidade. Estes operadores, embora de grande utilidade pela sua facilidade de realização, revelam-se pouco eficazes na detecção de orlas em imagens captadas de cenas reais, uma vez que estas imagens, de um modo geral, apresentam pequenos detalhes e ruído que lhes é induzido pelos dispositivos e processos de aquisição e digitalização.

De modo a fazer face àquela dificuldade apresenta-se sucintamente o método baseado na aplicação de filtros óptimos de detecção, adaptados às formas e escalas das variações de interesse. É também apresentado, de forma sucinta, o filtro utilizado ao longo do trabalho desta dissertação, o filtro de

Deriche, sendo esta utilização devidamente justificada. Um exemplo de resultados obtidos pela utilização deste filtro numa imagem real é também apresentado. Critérios comuns de desempenho são apresentados, assim como as operações normalmente efectuadas após a aplicação de um filtro para detecção de orlas de intensidade.

Neste capítulo são também apresentados métodos para aproximação poligonal de linhas utilizando faixas de aproximação; assim, é apresentado um método que utiliza faixas estáticas, e um segundo que utiliza faixas dinâmicas; este último tem a liberdade de rodar a faixa de ajustamento de modo a incluir mais pontos no seu interior.

Um método inovador para o seguimento de linhas, e consequente aproximação poligonal utilizando faixas dinâmicas, é também apresentado. Trata-se do método implementado no âmbito desta dissertação e que utiliza no seguimento a direcção e dois níveis de amplitude, e na aproximação o já referido método de aproximação poligonal utilizando faixas dinâmicas. Com este método torna-se possível seguir e aproximar convenientemente linhas abertas e linhas fechadas, que poderão existir em conjunto numa mesma imagem de entrada, sem a necessidade de serem de espessura unitária. Como a detecção do ponto seguinte é baseada nos valores de amplitude e de direcção, não surgem erros em pontos de ancoragem de várias linhas. Neste método considera-se que os pixels que constituem a imagem original se encontram previamente classificados em termos de amplitude e de direcção, por meio de um detector de orlas de intensidade.

Também se analisa um método de fusão de segmentos de recta próximos e de direcções similares, que define o segmento de fusão resultante pelos dois pontos extremos mais afastados dos segmentos a fundir. Um outro método analisado, define o segmento de fusão resultante como passante pelos pontos médios dos dois segmentos a fundir, estando os seus pontos extremos sobre as perpendiculares cujos pés são os dois pontos extremos mais afastados entre si. Este segundo método obtém melhores resultados que o anterior. Contudo, nenhum dos métodos é de âmbito geral, isto é, não são adequados para todos os casos de segmentos a fundir: não sobrepostos, parcialmente sobrepostos ou totalmente sobrepostos. Além do mais, a localização do segmento de fusão, assim como a sua direcção, não são devidamente influenciadas pelos comprimentos dos segmentos a fundir. Assim, neste capítulo é também apresentado um novo método, concebido no âmbito desta dissertação. Trata-se de um método de âmbito geral, no qual a localização e a direcção do segmento de fusão resultante é devidamente influenciada pelos comprimentos dos segmentos a fundir. O segmento de fusão resultante é definido como:

- passante pelo centróide, determinado a partir do conjunto dos quatro pontos extremos dos segmentos a fundir, considerando os respectivos comprimentos como as massas pontuais;
- a sua direcção é a média pesada das direcções dos segmentos a fundir, sendo os comprimentos destes os respectivos pesos;
- as projecções dos pontos extremos dos segmentos a fundir sobre o referencial centrado no centróide e com um eixo paralelo à direcção de fusão, definem os dois pontos extremos do segmento de fusão, assim como concretizam os critérios de admissibilidade da fusão.

São também apresentadas, de forma sumária, as implementações desenvolvidas para o seguimento de linhas e consequente aproximação poligonal utilizando faixas dinâmicas, para fusão de segmentos de recta próximos e de direcções similares, assim como uma implementação para o desenho numa imagem dos segmentos de recta (constituídos pelo menos por um dado número de pontos) definidos pelos seus pontos iniciais e finais, num dado ficheiro de entrada.

Alguns resultados destas implementações são também apresentados neste capítulo. Pode-se concluir, a partir destes, que as referidas implementações produzem resultados de boa qualidade e que são de utilização bastante flexível.

Capítulo IV

SEGUIMENTO DE SEGMENTOS DE RECTA EM IMAGENS SUCESSIVAS E OBTENÇÃO DE INFORMAÇÃO TRIDIMENSIONAL

Continuando a abordagem proposta para esta dissertação, após a definição do problema da calibração de câmaras no contexto da visão tridimensional e a sua abordagem no capítulo II, e depois da solução global apresentada no capítulo anterior para a determinação numa dada imagem das entidades que irão ser objecto do seguimento, é neste capítulo explanada uma abordagem global para o seguimento ao longo duma sequência e consequente obtenção de informação tridimensional dos seus pontos iniciais e finais.

Para o seguimento é apresentado um método que utiliza uma parametrização das entidades baseada na posição do ponto médio, na direcção e no comprimento, três filtros de Kalman independentes (um para cada característica de parametrização) para a estimação das características das entidades ao longo da sequência e, como medida do emparelhamento, distâncias normalizadas de Mahalanobis ou, quando tal não é conseguido, restrições geométricas.

Para a obtenção de informação tridimensional, é apresentado neste capítulo um método que utiliza o princípio da triangulação estereoscópica e que determina, previamente, a verdadeira localização dos pontos inicial e final de um segmento de recta visível numa dada imagem sobre o segmento de recta visível na anterior a esta e com o qual foi emparelhado.

São ainda apresentadas as implementações desenvolvidas, assim como alguns resultados experimentais.

Introdução

O seguimento de segmentos de recta em imagens sucessivas desempenha um papel muito importante em várias aplicações de visão por computador. Exemplos de tais aplicações são o seguimento de objectos quando estes são definidos por uma estrutura em arame, a determinação das características do movimento de um dado objecto quando este é definido como constituído por um determinado conjunto de segmentos de recta, a obtenção de informação tridimensional a partir dos segmentos de recta que constituem as diversas imagens de uma sequência, etc. Neste capítulo é apresentado um método, concebido e implementado no âmbito desta dissertação, para o referido seguimento. Trata-se de um método que utiliza:

- uma parametrização das entidades baseada na posição do ponto médio, na direcção e no comprimento;
- três filtros de Kalman independentes para a estimação das características de parametrização de cada entidade - um para cada uma das características de parametrização a estimar;
- distâncias normalizadas de Mahalanobis ou, quando desta forma o emparelhamento não é conseguido com êxito, restrições geométricas na fase do emparelhamento das diversas entidades.

A utilização do seguimento de segmentos de recta em imagens sucessivas pode, como já referido, prender-se com a necessidade da obtenção de informação tridimensional a partir das imagens de vários segmentos de recta obtidas por uma câmara com uma determinada geometria interna e de movimento conhecido. Tal obtenção de informação tridimensional pode ser necessária em várias aplicações de visão por computador como, por exemplo, no guiamento de veículos autónomos - torna-se indispensável que este disponha de informação acerca da constituição do ambiente em que se situa -, no reconhecimento de objectos - torna-se útil a utilização de informação tridimensional de certas características destes, como por exemplo as coordenadas 3D dos vértices -, etc.

No método desenvolvido, a obtenção das coordenadas tridimensionais é conseguida pela resolução do respectivo sistema de projecção estéreo após a determinação da verdadeira localização dos pontos inicial e final do segmento de recta visível numa dada imagem sobre o segmento de recta visível na anterior a esta, e ao qual foi devidamente emparelhado. Tal determinação é realizada utilizando linhas epipolares ou, quando deste modo não é possível, analisando as posições dos referidos pontos nas duas imagens a considerar. De referir que se optou por apenas determinar as coordenadas 3D dos pontos inicial e final de cada segmento de recta, sendo obtida uma aproximação da profundidade dos restantes pontos por interpolação linear.

Assume-se que os diversos segmentos de recta que constituem as várias imagens da respectiva sequência estão previamente determinados por um qualquer processo¹, assim como se considera que a câmara utilizada está previamente calibrada².

Apresentam-se ainda, de forma sumária, as implementações desenvolvidas neste domínio. São também apresentados alguns resultados experimentais obtidos por estas implementações, bem como algumas conclusões finais.

Na *Fig. 1* está representada a abordagem global induzida neste capítulo para o seguimento dos segmentos de recta que constituem as diversas imagens de uma dada sequência e para a consequente obtenção das coordenadas 3D dos pontos de cada um destes segmentos.

1 - Seguimento de segmentos de recta em imagens sucessivas

Nesta secção é apresentado o método concebido e implementado para o seguimento de segmentos de recta em imagens sucessivas. No referido método considera-se que:

¹ Ver capítulo III.

² Ver capítulo II.

- As entidades a seguir são parametrizados pela posição do ponto médio, pela direcção e pelo comprimento.
- Para o seguimento são utilizados três filtros de Kalman³ independentes; um para a posição do ponto médio, outro para a direcção e um último para o comprimento; todos utilizam um modelo cinemático de aceleração constante localmente. Como medida do emparelhamento são utilizadas distâncias normalizadas de Mahalanobis³ ou restrições geométricas. A utilização dos três filtros independentes deve-se ao facto de as variáveis de parametrização não apresentarem qualquer relação entre si.
- É utilizado um modelo 2D de entidades; assim, sempre que uma nova entidade se torna visível, ela é inserida neste com um factor de confiança de valor médio; sempre que uma entidade é novamente visível na imagem seguinte, este factor é aumentado de uma unidade até ao valor máximo admissível; se, pelo contrário, não é visível na imagem seguinte (isto é, não é possível o seu emparelhamento), este factor é diminuído de uma unidade até ao valor mínimo admissível, após o que é removida do modelo 2D. Desta forma, dispõe-se de um modelo 2D cujos elementos estão continuamente actualizados.

Esta apresentação está dividida em duas fases: na primeira, é descrita a parametrização utilizada para as entidades e, na segunda, é descrita a filosofia utilizada para o seguimento das entidades em imagens sucessivas.

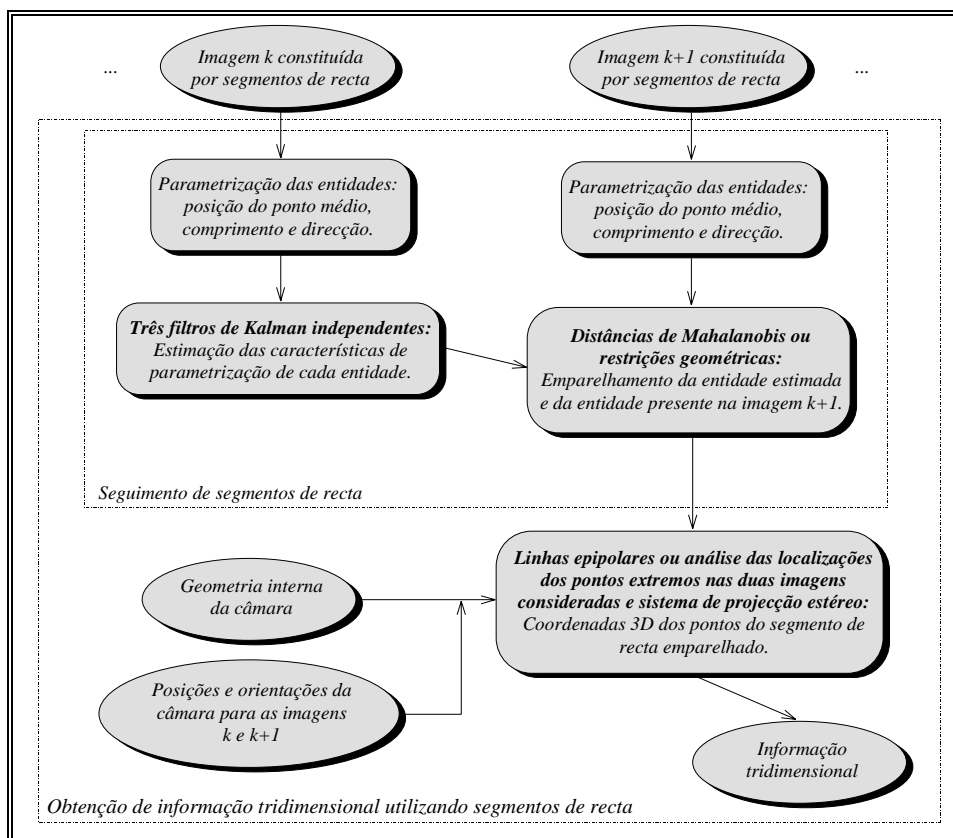


Fig. 1 - Abordagem global induzida neste capítulo para o seguimento de segmentos de recta em imagens sucessivas e para a conseqüente obtenção de informação tridimensional.

³ Ver, por exemplo [Maybeck, 1979].

1.2 - Parametrização utilizada para as entidades

As entidades utilizadas necessitam de ser devidamente parametrizadas; uma possível parametrização dos segmentos de recta poderia consistir nos dois pontos extremos; no entanto, uma grande desvantagem desta parametrização é a possibilidade de os referidos segmentos poderem ser visíveis diferentemente em imagens sucessivas [Deriche, 1990]. Uma outra solução possível poderia ser a orientação θ , o comprimento l , a distância c da origem ao segmento de recta e a distância d do pé da perpendicular ao segmento de recta que passa pela origem ao ponto médio do mesmo, Fig. 2.

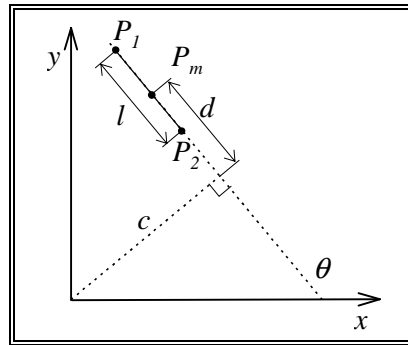


Fig. 2 - Parametrização θ , l , c e d , para as entidades.

No entanto, esta parametrização apresenta também algumas desvantagens, nomeadamente os parâmetros são correlacionados, o que implica a utilização de um único filtro de Kalman, de maior dimensão, com o conseqüente custo na eficiência; os parâmetros c e d dependem fortemente da posição da respectiva entidade na imagem, o que dificulta o emparelhamento das entidades [Checa, 1993]. Deste modo, uma possível parametrização, que não apresenta tais desvantagens, é a posição do ponto médio, a direcção e o comprimento. Devido a estes parâmetros não estarem correlacionados entre si, esta parametrização permite a utilização de três filtros de Kalman independentes: um, para as coordenadas do ponto médio x_m e y_m ; outro, para a direcção θ ; e um último, para o comprimento l . Desta possibilidade resulta um aumento de eficiência [Deriche, 1990; Checa, 1993]. Assim, esta parametrização é a adoptada no método implementado para o seguimento dos segmentos de recta que constituem as diversas imagens de uma dada sequência.

1.3 - Filosofia utilizada para o seguimento das entidades em imagens sucessivas

No seguimento das entidades entre duas imagens sucessivas utilizam-se três filtros de Kalman independentes, como se referiu antes. O modelo cinemático associado a cada filtro é um modelo de aceleração constante localmente, sendo a aceleração considerada como um processo Gauss-Markov de primeira ordem e igual a uma fracção α da aceleração anterior. Deste modo, a aceleração estimada tende para zero; isto significa que, se uma entidade desaparece (isto é, não é emparelhada com êxito em imagens sucessivas provocando uma incerteza elevada), o filtro diminuirá a última aceleração conhecida gradualmente até zero, através de potências de α . A utilização de um modelo cinemático deste tipo torna este método mais flexível e adaptável a diferentes tipos de aplicação.

Como medida do emparelhamento são utilizadas distâncias normalizadas de Mahalanobis ou, quando desta forma o emparelhamento não é conseguido com êxito, restrições geométricas. Na Fig. 3 está representada a filosofia geral utilizada para o seguimento das entidades em imagens sucessivas.

Seguidamente são descritos o filtro de Kalman, os elementos do filtro de Kalman utilizado para a posição do ponto médio, os elementos do utilizado para a direcção, os elementos do utilizado para o comprimento, a distância normalizada de Mahalanobis, as restrições geométricas e a fase de medição e emparelhamento.

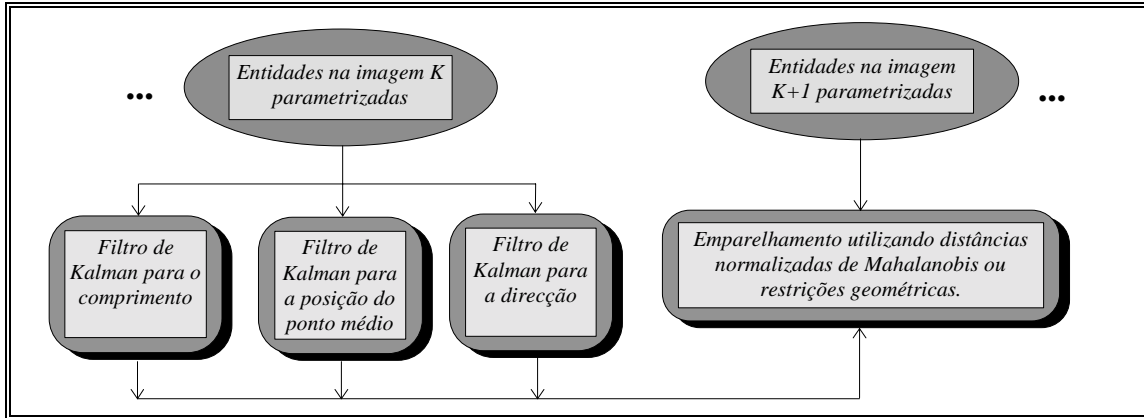


Fig. 3 - Filosofia para o seguimento de entidades em imagens sucessivas.

1.3.1 - Filtro de Kalman

O filtro de Kalman é utilizado para a estimativa do comportamento das características das diversas entidades ao longo do tempo, isto é, na sequência de imagens. Trata-se de uma abordagem estatística para estimar um vector de características que variam ao longo do tempo, \hat{x}_t , a partir de medidas que incluem ruído, \hat{z}_t . Considere-se a estimativa de \hat{x}_{t+k} a partir da medição até ao instante k ; o filtro de Kalman é um esquema recursivo desenvolvido para descrever o modelo dinâmico do sistema, a estatística do erro entre o modelo e a realidade, e a incerteza associada à medição. O filtro de Kalman pode ser interpretado como constituído pelas seguintes fases:

- **Previsão:**

A estimativa para o vector de características \hat{x} no instante t é:

$$\hat{x}_t^- = \Phi \hat{x}_{t-1}^+ \quad (\text{Eq. 1})$$

onde Φ é a matriz cinemática (isto é, uma matriz constante que envolve as características consideradas e todas as suas derivadas entre instantes sucessivos) e os índices + e - indicam a estimativa posterior e a anterior à fase de medição, respectivamente.

No instante $t=0$ (isto é, na inicialização), considera-se que as características desconhecidas são nulas e às restantes atribui-se o respectivo valor, determinado por medição.

A estimativa para a incerteza associada ao vector de características \hat{x} no instante t é:

$$P_t^- = \Phi P_{t-1}^+ \Phi^T + Q, \quad (\text{Eq. 2})$$

onde Q é uma matriz diagonal, cujos elementos são determinados em função da aplicação em causa, que traduz a inclusão na modelação da variância do ruído Gaussiano de média nula. Neste método, tornou-se essencial a modelação da variância do ruído Gaussiano de média nula de modo a evitar que os filtros de Kalman utilizados convergissem em demasia.

No instante $t=0$, considera-se que a matriz de incerteza é diagonal com os respectivos elementos assumindo valores arbitrados; estes valores são melhor determinados pela experiência acumulada.

- **Medição e emparelhamento:**

Após a estimação do vector de características e da sua incerteza, deve-se proceder à medição das características no interior de uma dada área. De notar que não é necessário proceder-se à medição das derivadas das características de parametrização consideradas, pois um aspecto interessante do filtro de Kalman é a possibilidade de estimar recursivamente as derivadas das características usando apenas a medição destas. Além do mais, devido a estas estimativas serem desenvolvidas por integração, elas são mais imunes ao ruído do que as derivadas instantâneas calculadas por simples diferença [Crowley, 1992].

De seguida deve-se seleccionar, entre as várias entidades candidatas ao emparelhamento, aquela que apresenta características que melhor se aproximam das estimadas. Após o emparelhamento, o vector de características medidas, \hat{z}_t , assume os valores das características da entidade emparelhada, e a matriz de incerteza associada a este vector, R_t , assume as variâncias medidas. Como se depreende, esta fase é fortemente dependente da aplicação.

• **Actualização:**

Após a medição e o emparelhamento, deve-se calcular o ganho do filtro de Kalman, K_t ; este ganho traduz o peso relativo que se deve atribuir à medição e à estimativa, sendo o seu cálculo baseado nas incertezas relativas. A equação do ganho do filtro de Kalman é:

$$K_t = P_t^- H^T [H P_t^- H^T + R_t]^{-1}, \quad (Eq. 3)$$

onde H é a matriz que transforma o sistema de coordenadas do vector de características estimado, \hat{x}_t , no sistema do vector de características medido, \hat{z}_t , e R_t é a matriz de variância medida.

Após o cálculo do ganho do filtro de Kalman, pode-se realizar a actualização da estimativa, utilizando a medição efectuada. Assim, a actualização do vector de características estimado é:

$$\hat{x}_t^+ = [-K_t H] \hat{x}_t^- + K_t \hat{z}_t, \text{ onde } I \text{ é a matriz identidade.} \quad (Eq. 4)$$

A correcção da matriz de incerteza associada ao vector de características estimado é:

$$P_t^+ = [-K_t H] P_t^- [-K_t H]^T + K_t R_t K_t^T. \quad (Eq. 5)$$

Observando as duas equações anteriores, depreende-se que, na actualização, quanto maior for o ganho do filtro de Kalman maior será o peso atribuído à medição e, conseqüentemente, o menor peso associado à estimativa efectuada pelo filtro. Assim, se a incerteza associada à medição for menor do que a associada à estimativa, o filtro de Kalman atribui maior peso à medição, e vice-versa.

De notar que as duas equações anteriores podem ser escritas de forma diferente; no entanto, as formas apresentadas são aquelas que traduzem um cálculo computacionalmente melhor condicionado, menos sensível aos erros associados ao filtro de Kalman, e menos sensível à truncagem numérica [Maybeck, 1979].

Na Fig. 4 está representado o diagrama de funcionamento do filtro de Kalman.

Como se depreende da Fig. 4, sempre que se inicializa uma entidade deve-se realizar a actualização da mesma, pois só assim é mantida a ordem medição-emparelhamento-actualização-previsão. Além do mais, desta forma diminuem-se os erros associados às atribuições realizadas na inicialização de cada entidade considerada.

1.3.1.1 - Elementos do filtro de Kalman utilizado para a posição do ponto médio das entidades

Para o filtro de Kalman utilizado para a posição do ponto médio das entidades, considera-se que as coordenadas do mesmo ponto são dependentes. Os diferentes elementos que estão presentes nas equações para este filtro de Kalman são a seguir definidos:

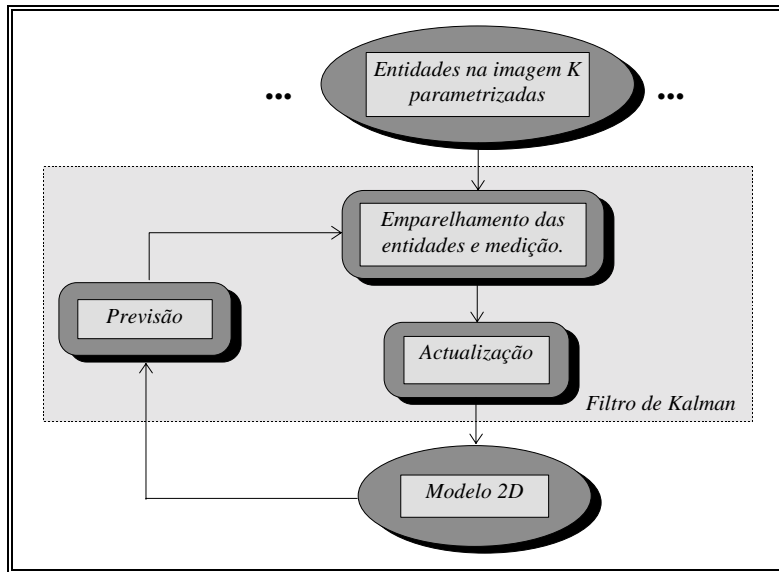


Fig. 4 - Diagrama de funcionamento do filtro de Kalman.

A matriz cinemática:

$$\Phi = \begin{bmatrix} 1 & 0 & \Delta t & 0 & \frac{\Delta t^2}{2} & 0 \\ 0 & 1 & 0 & \Delta t & 0 & \frac{\Delta t^2}{2} \\ 0 & 0 & 1 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & \alpha \\ 0 & 0 & 0 & 0 & 0 & \alpha \end{bmatrix}$$

onde Δt é o intervalo de tempo considerado e α é a fracção a considerar da última aceleração conhecida.

O vector de características a estimar:

$$\hat{x}_t^T = [x \quad y \quad \dot{x} \quad \dot{y} \quad \ddot{x} \quad \ddot{y}]$$

onde x e y são as coordenadas do ponto médio segundo os eixos xx e yy , \dot{x} e \dot{y} são as velocidades do mesmo ponto segundo os referidos eixos, \ddot{x} e \ddot{y} são as acelerações do mesmo ponto segundo os referidos eixos.

No instante $t = 0$, este vector é $\hat{x}_{t=0}^T = [x_{t=0} \quad y_{t=0} \quad 0 \quad 0 \quad 0 \quad 0]$

A matriz de incerteza associada ao vector de características estimado:

$$C_t = \begin{bmatrix} xx & xy & x\dot{x} & x\dot{y} & x\ddot{x} & x\ddot{y} \\ yx & yy & y\dot{x} & y\dot{y} & y\ddot{x} & y\ddot{y} \\ \dot{x}x & \dot{x}y & \dot{x}\dot{x} & \dot{x}\dot{y} & \dot{x}\ddot{x} & \dot{x}\ddot{y} \\ \dots & \dots & \dots & \dots & \dots & \dots \\ \dot{y}x & \dot{y}y & \dot{y}\dot{x} & \dot{y}\dot{y} & \dot{y}\ddot{x} & \dot{y}\ddot{y} \end{bmatrix}$$

No instante $t = 0$, esta matriz é:

$$C_t = \begin{bmatrix} xx_{t=0} & 0 & 0 & 0 & 0 & 0 \\ 0 & yy_{t=0} & 0 & 0 & 0 & 0 \\ 0 & 0 & \dot{x}\dot{x}_{t=0} & 0 & 0 & 0 \\ 0 & 0 & 0 & \dot{y}\dot{y}_{t=0} & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddot{x}\ddot{x}_{t=0} & 0 \\ 0 & 0 & 0 & 0 & 0 & \ddot{y}\ddot{y}_{t=0} \end{bmatrix}$$

Os valores dos elementos da diagonal desta matriz são determinados pela experiência desenvolvida.

A variância do ruído Gaussiano de média nula:

$$Q = \begin{bmatrix} xx_G & 0 & 0 & 0 & 0 & 0 \\ 0 & yy_G & 0 & 0 & 0 & 0 \\ 0 & 0 & \dot{x}\dot{x}_G & 0 & 0 & 0 \\ 0 & 0 & 0 & \dot{y}\dot{y}_G & 0 & 0 \\ 0 & 0 & 0 & 0 & \ddot{x}\ddot{x}_G & 0 \\ 0 & 0 & 0 & 0 & 0 & \ddot{y}\ddot{y}_G \end{bmatrix}$$

Os valores dos elementos da diagonal desta matriz são determinados pela experiência desenvolvida, e deverão traduzir os efeitos da digitalização, da determinação das entidades, assim como o facto de as velocidades e acelerações das coordenadas do ponto médio não serem medidas.

A matriz de medição:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Na filosofia seguida considera-se que apenas as coordenadas do ponto médio das entidades são medidas, isto é, as velocidades e as acelerações não são medidas.

1.3.1.2 - Elementos do filtro de Kalman utilizado para a direcção das entidades

Os diferentes elementos que estão presentes nas equações do filtro de Kalman utilizado para a direcção das entidades são definidos como:

A matriz cinemática:

$$\Phi = \begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} \\ 0 & 1 & \Delta t \\ 0 & 0 & \alpha \end{bmatrix}$$

O vector de características a estimar:

$$\hat{x}_t^T = [\theta \quad \dot{\theta} \quad \ddot{\theta}]$$

onde θ é a direcção, $\dot{\theta}$ a velocidade da direcção e $\ddot{\theta}$ a aceleração da mesma.

No instante $t = 0$, este vector é $\hat{x}_{t=0}^T = [\theta_{t=0} \quad 0 \quad 0]$

A matriz de incerteza associada ao vector de características estimado:

$$C_t = \begin{bmatrix} \theta\theta & \theta\dot{\theta} & \theta\ddot{\theta} \\ \dot{\theta}\theta & \dot{\theta}\dot{\theta} & \dot{\theta}\ddot{\theta} \\ \ddot{\theta}\theta & \ddot{\theta}\dot{\theta} & \ddot{\theta}\ddot{\theta} \end{bmatrix}$$

No instante $t = 0$, esta matriz é:

$$C_{t=0} = \begin{bmatrix} \theta\theta_{t=0} & 0 & 0 \\ 0 & \dot{\theta}\dot{\theta}_{t=0} & 0 \\ 0 & 0 & \ddot{\theta}\ddot{\theta}_{t=0} \end{bmatrix}.$$

Os valores dos elementos da diagonal desta matriz são determinados pela experiência desenvolvida.

A variância do ruído Gaussiano de média nula:

$$C_t = \begin{bmatrix} \theta\theta_G & 0 & 0 \\ 0 & \dot{\theta}\dot{\theta}_G & 0 \\ 0 & 0 & \ddot{\theta}\ddot{\theta}_G \end{bmatrix}$$

Os valores dos elementos da diagonal desta matriz são determinados pela experiência desenvolvida, e deverão traduzir os efeitos da digitalização, da determinação das entidades, assim como o facto de a velocidade e a aceleração da direcção não serem medidas.

O vector de medição:

$$H = [1 \ 0 \ 0]$$

Na filosofia seguida considera-se que apenas a direcção das entidades é medida, isto é, a velocidade e a aceleração da direcção não são medidas.

1.3.1.3 - Elementos do filtro de Kalman utilizado para o comprimento das entidades

Os diferentes elementos que estão presentes nas equações do filtro de Kalman utilizado para o comprimento das entidades são definidos como:

A matriz cinemática:

$$\Phi = \begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} \\ 0 & 1 & \Delta t \\ 0 & 0 & \alpha \end{bmatrix}$$

O vector de características a estimar:

$$\hat{x}_t^T = [l \ \dot{l} \ \ddot{l}]$$

Onde l é o comprimento, \dot{l} a velocidade do comprimento e \ddot{l} a aceleração do mesmo.

No instante $t = 0$, este vector é $\hat{x}_{t=0}^T = [l_{t=0} \ 0 \ 0]$

A matriz de incerteza associada ao vector de características estimado:

$$C_t = \begin{bmatrix} l & \dot{l} & \ddot{l} \\ \dot{l} & \ddot{l} & \dddot{l} \\ \ddot{l} & \dddot{l} & \ddddot{l} \end{bmatrix}$$

No instante $t = 0$, esta matriz é:

$$C_{t=0} = \begin{bmatrix} l_{t=0} & 0 & 0 \\ 0 & \dot{l}_{t=0} & 0 \\ 0 & 0 & \ddot{l}_{t=0} \end{bmatrix}.$$

Os valores dos elementos da diagonal desta matriz são determinados pela experiência desenvolvida.

Variância do ruído Gaussiano de média nula:

$$C_t = \begin{bmatrix} l_G & 0 & 0 \\ 0 & \dot{l}_G & 0 \\ 0 & 0 & \ddot{l}_G \end{bmatrix}$$

Os valores dos elementos da diagonal desta matriz são determinados pela experiência desenvolvida, e deverão traduzir os efeitos da digitalização, da determinação das entidades, assim como o facto de a velocidade e a aceleração do comprimento não serem medidas.

Vector de medição:

$$H = [1 \ 0 \ 0]$$

Na filosofia seguida, considera-se que apenas o comprimento das entidades é medido, isto é, a velocidade e a aceleração do comprimento não são medidas.

1.3.2 - Distância normalizada de Mahalanobis

No interior da área de emparelhamento, resultante do filtro de Kalman para a posição do ponto médio, é necessário associar uma medida do grau de emparelhamento de cada possível entidade candidata com a entidade cujas características foram estimadas pelos três filtros de Kalman utilizados. Na primeira tentativa de emparelhamento são utilizadas distâncias normalizadas de Mahalanobis como medida do mesmo. A distância normalizada de Mahalanobis é definida como a diferença de características normalizada pelas suas variâncias, sendo a sua definição matemática:

$$d_{x_n^2} = \frac{(X_m - X_e)^T (V_m + V_e)^{-1} (X_m - X_e)}{2} \quad (Eq. 6)$$

onde X_m é o vector de características e V_m a matriz de variância da entidade candidata ao emparelhamento; X_e é o vector de características e V_e a matriz de variância estimados pelo filtro respectivo.

Para o caso de uma única característica escalar, a equação anterior simplifica-se resultando:

$$d_{x_i^2} = \frac{(x_m - x_e)^2}{2(v_m + v_e)} \quad (Eq. 7)$$

onde x_m é a característica e v_m a variância da entidade candidata ao emparelhamento; x_e é a característica e v_e a variância estimadas pelo filtro respectivo.

Esta distância tem uma distribuição χ^2 , com o número de graus de liberdade N igual ao número de características consideradas. Assim, por exemplo, quando se pretende uma probabilidade de 95% de um emparelhamento com êxito e um grau de liberdade, o limiar para d_{χ^2} deverá ser 3.841458.

1.3.3 - Restrições geométricas

Quando o emparelhamento não é conseguido com êxito pela utilização das distâncias normalizadas de Mahalanobis, este é novamente tentado utilizando-se, para o mesmo efeito, restrições geométricas. As restrições geométricas utilizadas são:

- A diferença d_l entre o comprimento l_m da entidade candidata ao emparelhamento e o estimado pelo filtro de Kalman respectivo, l_e ; não deve ser superior a um dado valor admissível MAX_{ladm} , isto é:

$$d_l = |l_m - l_e| \leq MAX_{ladm}. \quad (Eq. 8)$$

- A diferença d_θ entre a direcção θ_m da entidade candidata ao emparelhamento e a estimada pelo filtro de Kalman respectivo, θ_e ; não deve ser superior a um dado valor admissível MAX_{\thetaadm} , isto é:

$$d_\theta = |\theta_m - \theta_e| \leq MAX_{\thetaadm}. \quad (Eq. 9)$$

- A diferença d_p entre a posição (X_m, Y_m) do ponto médio da entidade candidata ao emparelhamento e a estimada pelo filtro de Kalman respectivo, (X_e, Y_e) ; não deve ser superior a um dado valor admissível MAX_{padm} , isto é:

$$d_p = \sqrt{(X_m - X_e)^2 + (Y_m - Y_e)^2} \leq MAX_{padm}. \quad (Eq. 10)$$

Na Fig. 5 estão ilustradas as três restrições geométricas utilizadas nesta filosofia.

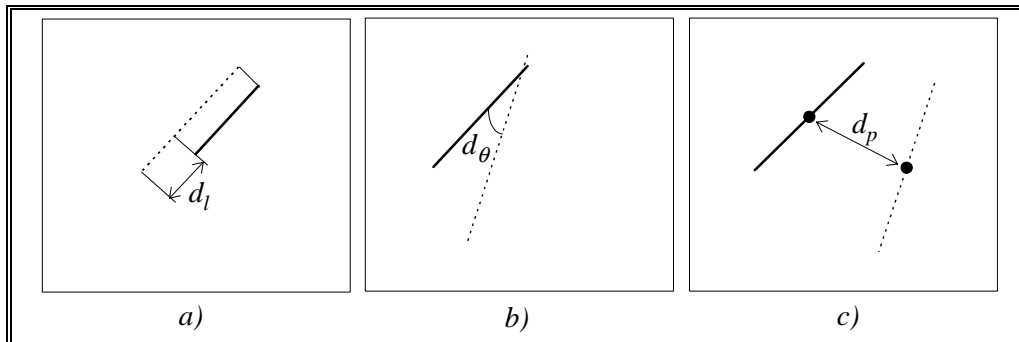


Fig. 5 - Restrições geométricas utilizadas: a) diferença entre comprimentos, b) diferença entre direcções e c) diferença entre as posições dos pontos médios. A traço cheio a característica da entidade candidata ao emparelhamento, e a interrompido a prevista pelo filtro respectivo.

1.3.4 - Fase de medição e emparelhamento

A estimação da matriz de incerteza associada ao vector de características estimado pelo filtro de Kalman utilizado para a posição do ponto médio das entidades determina uma área, em forma de elipse, no interior da qual a entidade cujas características foram estimadas pelos três filtros de Kalman deverá encontrar-se. Assim definindo-se a matriz M segundo a equação:

$$M = HP_t^T \quad (Eq. 11)$$

onde, para o filtro de Kalman para a posição do ponto médio, H é a matriz que transforma o sistema de coordenadas do vector de características estimado, \hat{x}_t , no sistema do vector de características medido, \hat{z}_t , e P_t^- é a matriz de incerteza estimada.

Desta forma, é possível determinar a referida área elíptica de emparelhamento do seguinte modo: os valores próprios da matriz M definem os raios maior e menor da elipse de emparelhamento, e os vectores próprios da mesma matriz definem os eixos maior e menor da mesma elipse. Esta elipse está naturalmente centrada na posição estimada para a entidade em causa pelo filtro utilizado para a posição do ponto médio. Definida esta elipse, *Fig. 6*, é necessário realizar o emparelhamento com a entidade cujas características melhor se aproximem das estimadas pelos filtros de Kalman utilizados.

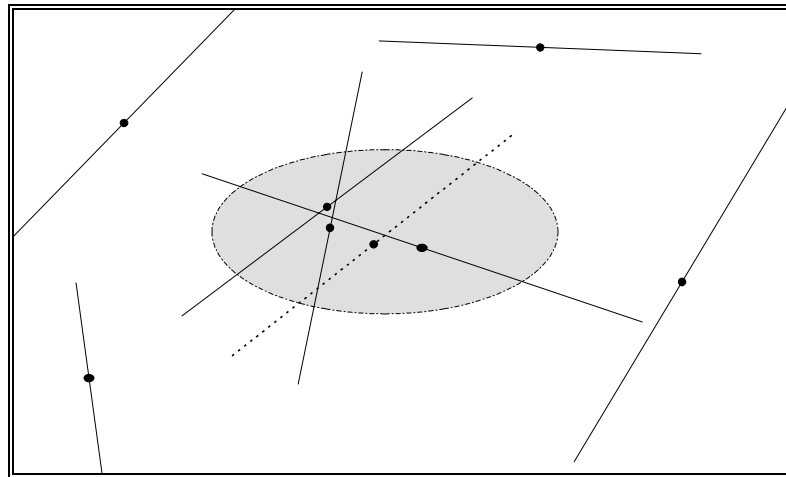


Fig. 6 - Elipse resultante do filtro de Kalman utilizado para a posição do ponto médio. A traço continuo as entidades que realmente existem na imagem, e a interrompido a "entidade" estimada pelos três filtros de Kalman utilizados.

Numa primeira fase, o emparelhamento é tentado do seguinte modo:

- I)** Considerar como entidades candidatas ao emparelhamento todas aquelas cujo ponto médio é interior à elipse de emparelhamento e que ainda não foram emparelhadas.
- II)** Se não existe nenhuma candidata ao emparelhamento, então terminar.
- III)** Fazer para todas as candidatas ao emparelhamento:
 - a)** Calcular a distância de Mahalanobis $d_{x_i}(\theta_i, \theta_e)$ entre a direcção θ_i da candidata ao emparelhamento e a prevista pelo filtro de Kalman respectivo θ_e .
 - b)** Se a distância calculada anteriormente é maior que um dado limiar, então voltar ao ponto **III)** considerando a próxima candidata.
 - c)** Calcular a distância de Mahalanobis $d_{x_i}(l_i, l_e)$ entre o comprimento l_i da candidata ao emparelhamento e o previsto pelo filtro de Kalman respectivo l_e .
 - d)** Se a distância calculada anteriormente é maior que um dado limiar, então voltar ao ponto **III)** considerando a próxima candidata.
 - e)** Calcular a distância de Mahalanobis $d_{x_i}((x, y)_i, (x, y)_e)$ entre a posição do ponto médio $(x, y)_i$ da candidata ao emparelhamento e a prevista pelo filtro de Kalman respectivo $(x, y)_e$.
 - f)** Se a distância calculada anteriormente é maior que um dado limiar, então voltar ao ponto **III)**, considerando a próxima candidata.
 - g)** Calcular o produto das três distâncias calculadas anteriormente.

- h) Se o produto calculado anteriormente é menor que o mínimo encontrado até esta altura, então a melhor entidade ao emparelhamento é esta candidata⁴.
- i) Caso ainda existam mais candidatas ainda não consideradas, então voltar ao ponto **III**), considerando a próxima candidata.

Se o emparelhamento é conseguido com êxito, então o estado medido é o seguinte:

Para o caso do filtro de Kalman para a direcção:

O primeiro elemento do vector de características medido assume o valor correspondente à direcção da entidade determinada pelo emparelhamento θ_i , o elemento $(1, 1)$ da matriz de incerteza associada à medição assume o valor correspondente à variância da direcção calculada no interior da área elíptica, var_θ . Os elementos do estado medido são então:

$$\hat{z}_i^T = [\theta_i \ 0 \ 0] \text{ e } R_i = \begin{bmatrix} var_\theta & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \text{ com } var_\theta = \frac{\sum_{j=1}^n (\theta_j - \bar{\theta})^2}{n},$$

onde n é o número de entidades no interior da área de medição, θ_j é a direcção da candidata j ao emparelhamento cujo ponto médio é interior à mesma área, e $\bar{\theta}$ é a direcção média das candidatas cujo ponto médio é interior à mesma área.

Para o caso do filtro de Kalman para o comprimento:

O primeiro elemento do vector de características medido assume o valor correspondente ao comprimento da entidade determinada pelo emparelhamento l_i , o elemento $(1, 1)$ da matriz de incerteza associada à medição assume o valor correspondente à variância do comprimento calculada no interior da área elíptica, var_l . Os elementos do estado medido são então:

$$\hat{z}_i^T = [l_i \ 0 \ 0] \text{ e } R_i = \begin{bmatrix} var_l & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \text{ com } var_l = \frac{\sum_{j=1}^n (l_j - \bar{l})^2}{n},$$

onde n é o número de entidades no interior da área de medição, l_j é o comprimento da candidata j ao emparelhamento cujo ponto médio é interior à mesma área, e \bar{l} é o comprimento médio das candidatas cujo ponto médio é interior à mesma área.

Para o caso do filtro de Kalman para a posição do ponto médio:

O primeiro e terceiro elementos do vector de características medido assumem os valores correspondentes às coordenadas do ponto médio da entidade determinada pelo emparelhamento, x_i e

⁴ A minimização do produto das três distâncias de Mahalanobis, para determinação da melhor entidade ao emparelhamento, é baseada no facto de que a probabilidade de três eventos independentes se verificarem em simultâneo ser equivalente ao produto das probabilidades individuais. Verificou-se, ao longo das várias experiências realizadas, que esta minimização era a que, de entre várias utilizadas, obtinha, sistematicamente, os melhores resultados. Contudo esta minimização poderá, em certas situações, não originar os melhores emparelhamentos possíveis. Por exemplo, se alguma das distâncias é bastante reduzida quando comparada com as restantes o seu valor absorverá o produto. Desta forma, o emparelhamento resultante poderá ser com uma entidade que, em termos globais, não é a melhor. A ocorrência de tais situações pode ser diminuída por uma atribuição cuidada dos valores máximos admissíveis das respectivas distâncias normalizadas. Esta minimização necessita de ser alvo de um estudo mais extenso e, porventura, em termos futuros melhorada.

y_i ; os elementos $(1, 1)$, $(1, 2)$, $(2, 1)$ e $(2, 2)$, da matriz de incerteza associada à medição assumem os valores das variâncias calculadas no interior da área elíptica, var_{xx} , var_{xy} , var_{yx} e var_{yy} , respectivamente. Os elementos do estado medido são então:

$$\hat{z}_i^T = [x_i \ 0 \ y_i \ 0 \ 0 \ 0] e R_i = \begin{bmatrix} var_{xx} & var_{xy} & 0 & 0 & 0 & 0 \\ var_{yx} & var_{yy} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

com $var_{xx} = \frac{1}{m_{00}} \left(m_{20} - \frac{m_{10}^2}{m_{00}} \right)$, $var_{xy} = var_{yx} = \frac{1}{m_{00}} \left(m_{11} - \frac{m_{10}m_{01}}{m_{00}} \right)$, $var_{yy} = \frac{1}{m_{00}} \left(m_{02} - \frac{m_{01}^2}{m_{00}} \right)$, onde m_{ij} são os momentos de ordem ij calculados no interior da área de medição do seguinte modo:

$$m_{00} = \sum_{j=1}^n f(\theta_j, l_j), \quad m_{10} = \sum_{j=1}^n f(\theta_j, l_j)x_j, \quad m_{01} = \sum_{j=1}^n f(\theta_j, l_j)y_j, \quad m_{11} = \sum_{j=1}^n f(\theta_j, l_j)x_jy_j,$$

$$m_{20} = \sum_{j=1}^n f(\theta_j, l_j)x_j^2, \quad m_{02} = \sum_{j=1}^n f(\theta_j, l_j)y_j^2, \quad f(\theta_j, l_j) = e^{-\left(d_{\chi_i^2}(\theta_j, \theta_e) + d_{\chi_i^2}(l_j, l_e)\right)},$$

onde n é o número de entidades no interior da área de medição, l_j é o comprimento e θ_j é a direcção da candidata j ao emparelhamento cujo ponto médio, de coordenadas x_j e y_j , é interior à mesma área.

Quando o emparelhamento não é conseguido com êxito na tentativa anterior, este é tentado de novo utilizando-se restrições geométricas como medida do mesmo, do seguinte modo:

I) Fazer para todas as entidades ainda não emparelhadas:

- a) Calcular a diferença d_θ entre a direcção θ_i da candidata ao emparelhamento e a prevista pelo filtro Kalman respectivo θ_e .
- b) Se a distância d_θ for maior que o respectivo limiar, então voltar ao ponto **I)**, considerando a próxima candidata.
- c) Calcular a diferença d_l entre o comprimento l_i da candidata ao emparelhamento e o previsto pelo filtro Kalman respectivo l_e .
- d) Se a distância d_l for maior que o respectivo limiar, então voltar ao ponto **I)**, considerando a próxima candidata.
- e) Calcular a diferença d_p entre a posição do ponto médio p_i da candidata ao emparelhamento e a prevista pelo filtro Kalman respectivo p_e .
- f) Se a distância d_p for maior que o respectivo limiar, então voltar ao ponto **I)**, considerando a próxima candidata.
- g) Calcular o produto $d_\theta d_l d_p$.

- h) Se o produto calculado anteriormente for menor que o mínimo encontrado até esta altura, então a melhor entidade ao emparelhamento é esta candidata⁵.
- i) Caso ainda exista alguma candidata ainda não considerada, então voltar ao ponto **I**), considerando a próxima candidata.

Quando o emparelhamento é, desta feita, conseguido com êxito, o estado medido é calculado da forma descrita anteriormente; no entanto, a área elíptica de emparelhamento é, neste caso, centrada no ponto médio da entidade emparelhada pela utilização das restrições geométricas. Com este procedimento, os filtros de Kalman para a entidade em questão são reinicializados.

Se, após esta tentativa, o emparelhamento não é conseguido com êxito, uma causa possível pode ser o desaparecimento definitivo ou temporário da entidade; o estado medido é então assumido como:

Para o caso do filtro de Kalman para a direcção:

O primeiro elemento do vector de características medido assume o valor correspondente à direcção estimada pelo filtro de Kalman utilizado para a direcção θ_e ; o elemento $(1, 1)$ da matriz de incerteza associada à medição assume o valor correspondente à variância da direcção atribuída na fase de inicialização $\theta\theta_{t=0}$. Os elementos do estado medido são então:

$$\hat{z}_t^T = [\theta_e \ 0 \ 0] \text{ e } R_t = \begin{bmatrix} \theta\theta_{t=0} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Para o caso do filtro de Kalman para o comprimento:

O primeiro elemento do vector de características medido assume o valor correspondente ao comprimento estimado pelo filtro de Kalman utilizado para o comprimento l_e ; o elemento $(1, 1)$ da matriz de incerteza associada à medição assume o valor correspondente à variância do comprimento atribuída na fase de inicialização $ll_{t=0}$. Os elementos do estado medido são então:

$$\hat{z}_t^T = [l_e \ 0 \ 0] \text{ e } R_t = \begin{bmatrix} ll_{t=0} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

Para o caso do filtro de Kalman para a posição do ponto médio:

O primeiro e terceiro elementos do vector de características medido assumem os valores estimados pelo filtro de Kalman utilizado para a posição do ponto médio x_e e y_e ; os elementos $(1, 1)$ e $(2, 2)$ da matriz de incerteza associada à medição assumem os valores correspondentes às variâncias da posição atribuídas na fase de inicialização $xx_{t=0}$ e $yy_{t=0}$, respectivamente. Os elementos do estado medido são então:

⁵ Para resultados no emparelhamento o mais correctos possível (ver nota anterior), deve-se ter especial atenção ao valor que se atribui aos valores máximos admissíveis das respectivas distâncias; doutro modo, podem ocorrer erros no emparelhamento devido à propriedade absorvente da multiplicação.

$$\hat{z}_t^T = [x_e \ 0 \ y_e \ 0 \ 0 \ 0] e R_t = \begin{bmatrix} xx_{t=0} & 0 & 0 & 0 & 0 & 0 \\ 0 & yy_{t=0} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Com esta filosofia, quando o emparelhamento de uma dada entidade não é conseguido com êxito, aumenta-se a incerteza associada à medição de tal maneira que os filtros de Kalman aumentam o peso atribuído à previsão através da diminuição do ganho. Desta forma, na posterior previsão, a incerteza virá aumentada, provocando um conseqüente aumento da área elíptica de emparelhamento.

Naturalmente que, quanto menos as imagens sucessivas diferirem entre si, mais facilmente serão obtidos emparelhamentos com êxito e, conseqüentemente, mais fácil será o respectivo seguimento. Isto significa que a seqüência deverá ser constituída por um grande número de imagens e que o movimento dos segmentos de recta não deve ser grande entre imagens sucessivas, principalmente após os filtros de Kalman já terem convergido. No entanto, é de salientar que a utilização das restrições geométricas, como medida de emparelhamento, atenua tais requisitos.

2 - Determinação das coordenadas 3D dos pontos de cada entidade previamente emparelhada em duas imagens distintas

Como anteriormente referido, a obtenção de informação tridimensional de uma cena é muitas vezes desejável ou até indispensável; assim, surgem várias soluções para tentar responder adequadamente a esta necessidade. As duas soluções mais comuns são:

- duas ou mais câmaras obtêm imagens distintas de uma mesma cena; conhecendo-se os parâmetros intrínsecos e extrínsecos de cada câmara, é possível obter-se informação tridimensional a partir da resolução do sistema de projecção estéreo para as entidades previamente emparelhadas nas imagens de cada câmara;
- apenas uma câmara vai obtendo uma seqüência de imagens distintas de uma mesma cena; conhecendo-se os parâmetros intrínsecos e extrínsecos desta câmara, é possível obter-se informação tridimensional a partir da resolução do sistema de projecção estéreo para as entidades previamente emparelhadas nas diferentes imagens. Neste caso, existem duas situações possíveis: uma onde o movimento da câmara não é conhecido, e outra onde é conhecido.

Nesta secção é apresentado um método para este último caso, isto é, para a obtenção de informação tridimensional a partir do movimento conhecido de uma câmara. As características principais deste método são:

- o modelo utilizado para a câmara é o proposto por Roger Tsai, apresentado inicialmente em [Tsai, 1987] e descrito no capítulo II desta dissertação;
- a obtenção das coordenadas tridimensionais dos pontos inicial e final de cada segmento de recta é conseguida pela resolução do respectivo sistema de projecção estéreo, após a determinação da verdadeira localização dos pontos inicial e final do segmento de recta, visíveis numa imagem, sobre o segmento de recta visível na imagem anterior a esta; os dois segmentos de recta terão sido previamente emparelhados;
- uma aproximação para a profundidade dos restantes pontos dos segmentos de recta é obtida utilizando-se interpolação linear a partir das profundidades dos pontos inicial e final.

Seguidamente são apresentados: o princípio da triangulação estereoscópica, que permite determinar o sistema de projecção estéreo; a abordagem concebida para a determinação da verdadeira localização dos pontos inicial e final de um segmento de recta, visíveis numa dada imagem, sobre o segmento de recta visível na imagem anterior a esta.

2.1 - Princípio da triangulação estereoscópica

Quando se pretende obter informação tridimensional de uma cena a partir de imagens distintas obtidas por uma câmara, torna-se indispensável o prévio conhecimento do modelo da câmara utilizada. Assim, é necessário o conhecimento prévio de como pontos no espaço tridimensional são transformados em pontos no plano imagem. Esta determinação do modelo da câmara, isto é, a determinação da geometria interna e das características ópticas (parâmetros intrínsecos), e/ou determinação da orientação e da posição 3D da câmara relativamente a um certo sistema de coordenadas mundo (parâmetros extrínsecos), é designada por calibração da câmara. A calibração, geralmente, é realizada apenas uma vez e independentemente. Como já referido anteriormente, nesta abordagem o modelo utilizado para a câmara é o proposto por Roger Tsai.

Um ponto P no sistema de coordenadas 3D mundo é, então, transformado num ponto 2D no sistema do plano imagem por uma dada matriz de transformação global (designada, por vezes, por simplificação, como matriz de transformação perspectiva) determinada através da calibração da câmara. A equação que traduz esta transformação global para um ponto P , em coordenadas homogêneas, para uma dada câmara para a posição e orientação k , é:

$$\begin{bmatrix} wX_u^k \\ wY_u^k \\ w \end{bmatrix} = M_{pp_k} \begin{bmatrix} x_w \\ y_w \\ z_w \\ I \end{bmatrix} \text{ com } M_{pp_k} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix}, \quad (Eq. 12)$$

onde (wX_u^k, wY_u^k, w) são as coordenadas homogêneas não distorcidas do ponto P no sistema 2D do plano imagem da câmara para a posição e orientação k , (x_w, y_w, z_w, I) são as coordenadas homogêneas do mesmo ponto no sistema 3D mundo, e M_{pp_k} é a matriz não invertível da transformação global da mesma câmara para a posição e orientação k .

Como se pode observar da Eq. 12, não é possível determinar-se as coordenadas 3D do ponto P no sistema mundo a partir unicamente das coordenadas do mesmo no sistema 2D do plano imagem. Desta forma, é necessário obter-se as coordenadas do mesmo ponto no sistema 2D do plano imagem para outra posição e orientação da câmara, isto é:

$$\begin{bmatrix} wX_u^{k+1} \\ wY_u^{k+1} \\ w \end{bmatrix} = M_{pp_{k+1}} \begin{bmatrix} x_w \\ y_w \\ z_w \\ I \end{bmatrix} \text{ com } M_{pp_{k+1}} = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \end{bmatrix}, \quad (Eq. 13)$$

onde $(wX_u^{k+1}, wY_u^{k+1}, w)$ são as coordenadas homogêneas não distorcidas do ponto P no sistema 2D do plano imagem da câmara para a posição e orientação $k+1$, (x_w, y_w, z_w, I) são as coordenadas homogêneas do mesmo no sistema 3D mundo, e $M_{pp_{k+1}}$ é a matriz não invertível da transformação global da câmara para a posição e orientação $k+1$.

Utilizando as Eqs. 12 e 13, obtém-se um sistema de equações lineares sobredeterminado, geralmente designado por sistema de projecção estéreo:

$$\begin{bmatrix} a_{11} - a_{31}X_u^k & a_{12} - a_{32}X_u^k & a_{13} - a_{33}X_u^k \\ a_{21} - a_{31}Y_u^k & a_{22} - a_{32}Y_u^k & a_{23} - a_{33}Y_u^k \\ b_{11} - b_{31}X_u^{k+1} & b_{12} - b_{32}X_u^{k+1} & b_{13} - b_{33}X_u^{k+1} \\ b_{21} - b_{31}Y_u^{k+1} & b_{22} - b_{32}Y_u^{k+1} & b_{23} - b_{33}Y_u^{k+1} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \end{bmatrix} = \begin{bmatrix} a_{34}X_u^k - a_{14} \\ a_{34}Y_u^k - a_{24} \\ b_{34}X_u^{k+1} - b_{14} \\ b_{34}Y_u^{k+1} - b_{24} \end{bmatrix}. \quad (Eq. 14)$$

Trata-se de um sistema com quatro equações lineares para três incógnitas, onde intervêm apenas as coordenadas 2D não distorcidas do ponto P no plano imagem e as matrizes da transformação global da câmara para as posições e orientações k e $k + 1$. Este sistema pode ser resolvido em ordem às três incógnitas (as coordenadas do ponto P no sistema 3D mundo (x_w, y_w, z_w)) utilizando, por exemplo, o método dos mínimos quadrados com decomposição em valor único⁶ ou de Cholesky⁷.

De notar que as duas posições e orientações da câmara, k e $k + 1$, devem fazer entre si uma disparidade estereoscópica aceitável de maneira que a resolução do sistema anterior corresponda a soluções o mais correctas possível. Tal requisito é contrário ao referido na anterior secção - quanto menos as imagens sucessivas diferirem entre si, mais facilmente serão obtidos emparelhamentos com êxito e, conseqüentemente, o respectivo seguimento. Uma solução para resolver este conflito é utilizar todas as imagens para realizar o seguimento dos vários segmentos que as constituem, e obter as coordenadas 3D dos pontos que constituem estes segmentos para imagens não contíguas. Isto é, utilizar uma amostragem mais fina para o seguimento, e uma menos fina para a obtenção das coordenadas 3D.

2.2 - Determinação da verdadeira localização dos pontos inicial e final de um segmento de recta, visíveis numa dada imagem, sobre o segmento de recta visível na imagem anterior a esta

Neste método optou-se, como já anteriormente referido, pela determinação das coordenadas 3D no sistema mundo apenas dos pontos inicial e final de cada entidade; no entanto, pode acontecer que alguns dos diversos segmentos de recta considerados sejam visíveis nas duas imagens sucessivas de diferente modo, *Fig. 7*.

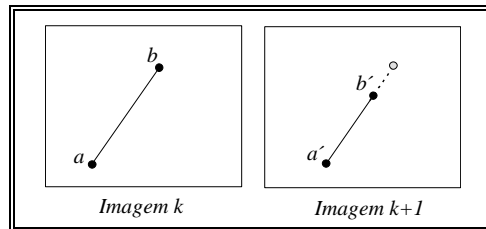


Fig. 7 - Uma entidade visível na imagem k e na $k + 1$ de diferente modo: a e a' correspondem ao mesmo ponto 3D mas b e b' correspondem, na realidade, a pontos 3D distintos.

A solução utilizada consiste na utilização de linhas epipolares para a determinação da verdadeira localização dos pontos extremos a' e b' de um segmento de recta na imagem $k + 1$ sobre o segmento na imagem k ao qual foi emparelhado, *Fig. 8*.

Deste modo, surge a necessidade da definição de linha epipolar: designa-se por linha epipolar a recta de intersecção de um plano, designado por plano epipolar, com o plano imagem da câmara. Por sua vez, o plano epipolar é definido pelo ponto P no sistema de coordenadas 3D mundo e pela localização do centro óptico para a imagem k e do centro óptico para a imagem $k + 1$, *Fig. 9*.

⁶ Ver, por exemplo [Press, 1992].

⁷ Ver, por exemplo [Chapra, 1988].

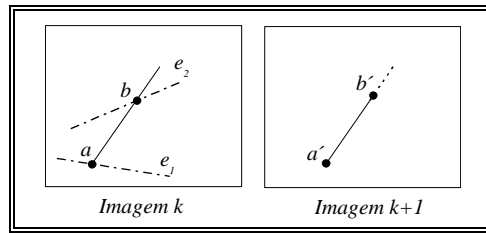


Fig. 8 - Utilização das linhas epipolares e_1 e e_2 para a determinação da verdadeira localização dos pontos a' e b' do segmento de recta na imagem $k + 1$ no segmento na imagem k .

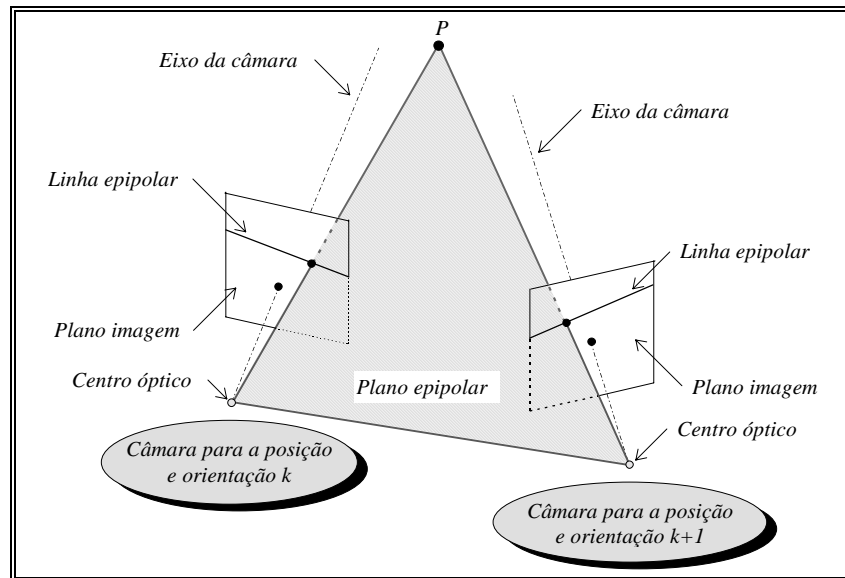


Fig. 9 - Definição de linha e plano epipolar.

Uma solução para a determinação da linha epipolar está representada na Fig. 10. Assim, conhecidas as coordenadas de um ponto P no sistema 2D do plano imagem da câmara para a posição e orientação $k + 1$, pode-se definir a recta de projecção deste ponto no mesmo plano como passante pelo ponto P e pelo centro de projecção da mesma câmara. Nesta recta de projecção, definem-se dois pontos A e B ; as projecções destes pontos no plano imagem da câmara para a posição e orientação k definem sobre este a respectiva linha epipolar. Esta solução, foi inicialmente proposta em [Mendonça, 1990], foi a adoptada.

De notar que a informação da linha epipolar poderia ser também utilizada no emparelhamento; por exemplo, poderia utilizar-se, como medida do mesmo, a distância do ponto médio da entidade candidata ao emparelhamento à linha epipolar do ponto médio da entidade que se pretende emparelhar. No entanto, durante as várias experiências realizadas, verificou-se que esta medida de emparelhamento, quando única, não era tão eficaz como as soluções anteriormente apresentadas e, quando em conjunto com estas, tornava-se redundante.

Em certas situações, pode acontecer que a linha epipolar para um dos pontos extremos seja praticamente paralela ao segmento de recta na imagem k ; tal posição relativa pode implicar que a intersecção destas duas entidades se verifique muito afastada do ponto desejado, por ampliação do erro, Fig. 11.

Este problema foi resolvido do seguinte modo:

- quando tal situação apenas se verifica para um dos extremos, utiliza-se a linha epipolar correspondente ao outro extremo para a determinação da sua verdadeira localização no segmento de recta na imagem k , e associa-se o extremo mais afastado ao outro extremo na imagem $k + 1$;

- quando tal situação se verifica para os dois pontos extremos, optou-se por representar a imagem do segmento da imagem $k + 1$ na imagem k , de forma a coincidirem os pontos médios; determina-se a distância nesta imagem k entre os vários pontos extremos e verifica-se se ocorreu ou não inversão dos pontos iniciais e finais do segmento nestas duas imagens, Fig. 12.

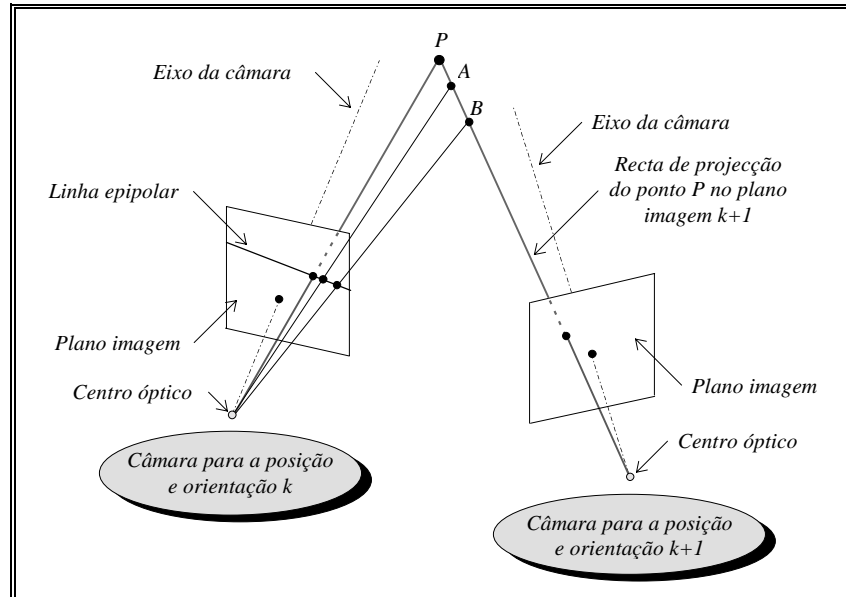


Fig. 10 - Determinação da linha epipolar no plano imagem da câmara na posição k do ponto P .

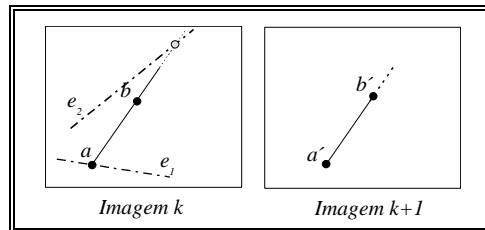


Fig. 11 - A utilização da linha epipolar e_2 para a determinação da verdadeira localização do ponto b' do segmento de recta na imagem $k + 1$ no segmento na imagem k (circulo cinzento) não define o ponto desejado (circulo preto).

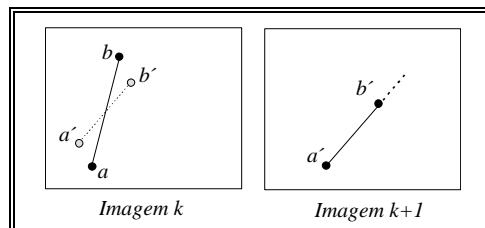


Fig. 12 - Representação da imagem do segmento de recta da imagem $k + 1$ na imagem k . Neste caso, como na imagem k a distância de b' a b é menor do que a de b' a a , os pontos b e b' são definidos como correspondentes ao mesmo ponto 3D, e os pontos a e a' são definidos como correspondentes a outro ponto 3D.

Naturalmente que esta última solução só produz resultados correctos se as imagens sucessivas do mesmo segmento de recta não diferirem muito entre si, principalmente no que respeita a rotações. Para ser possível a utilização desta solução, torna-se necessário a utilização de uma etiqueta para cada entidade considerada, para identificar que se trata de um segmento que sofreu ou não uma inversão na designação dos seus pontos extremos; esta etiqueta necessita de ser continuamente actualizada imagem após imagem (isto é, durante a amostragem mais fina utilizada no seguimento das entidades).

3 - Implementações desenvolvidas*

Nesta secção são apresentadas, de forma sucinta⁸, quatro implementações desenvolvidas no âmbito desta dissertação para o seguimento de segmentos de recta em imagens sucessivas, e para obtenção de coordenadas 3D. Estas implementações foram desenvolvidas em linguagem C em ambiente UNIX numa estação de trabalho SunSparc. As implementações integram-se no ambiente de processamento de imagem XITE⁹, do qual utilizam algumas funções e tipos de variáveis já previamente definidos. Assim, são apresentadas as seguintes quatro implementações:

- ✓ Para a determinação da área visível por uma dada câmara, sobre um plano de coordenadas especificadas: *simspace*.
- ✓ Simulador de uma dada câmara móvel para segmentos de recta no espaço tridimensional, cujas coordenadas 3D dos pontos iniciais e finais são especificadas: *simoca*.
- ✓ Para o seguimento de segmentos de recta em imagens sucessivas e posterior obtenção de coordenadas 3D dos pontos inicial e final de cada entidade emparelhada, quando o movimento da câmara é conhecido; isto é, obtenção de informação tridimensional a partir do movimento de uma câmara: *deep*.
- ✓ Para extracção de coordenadas 3D de pontos cujas coordenadas 2D em duas imagens obtidas por uma mesma câmara, são especificadas: *compdeep*.

As duas primeiras implementações *simspace* e *simoca*, têm como principal interesse a simulação de uma câmara móvel, de características conhecidas, quando as entidades a considerar são segmentos de recta no espaço tridimensional. Deste modo, torna-se possível obter dados simulados para posterior utilização com a implementação *deep*.

A quarta implementação, *compdeep*, tem como principal interesse analisar o processo de extracção de coordenadas 3D, segundo o princípio da triangulação estereoscópica.

Em todas as implementações apresentadas nesta secção, o modelo utilizado para a câmara foi o proposto por Roger Tsai.

3.1 - Determinação da área visível por uma dada câmara sobre um determinado plano: *simspace*

Neste ponto é apresentada uma implementação para a determinação da área visível por uma dada câmara sobre um plano de coordenadas especificadas. Esta implementação determina os quatro vértices da área visível por uma dada câmara sobre um dado plano.

A referida implementação é constituída pelos módulos *principal*, *compute_mr* e *frame_world*. Apresenta-se de seguida a descrição de cada um destes módulos.

a) *Módulo principal*, responsável pela interface implementação/utilizador. A seu cargo está:

- ✓ Leitura dos argumentos opcionais:
 - *xcenterbuf*, coordenada do centro da imagem na memória *frame* segundo a direcção *x*. Por defeito, é igual a metade da dimensão da memória *frame* segundo a mesma direcção.
 - *ycenterbuf*, coordenada do centro da imagem na memória *frame* segundo a direcção *y*. Por defeito, é igual a metade da dimensão da memória *frame* segundo a mesma direcção.
 - *xsizbuf*, dimensão da memória *frame* segundo a direcção *x*. Por defeito, é igual a 512.
 - *ysizbuf*, dimensão da memória *frame* segundo a direcção *y*. Por defeito, é igual a 512.

* Esta secção foi introduzida por razões de completude, mas não é essencial à compreensão das secções seguintes.

⁸ Em [Tavares, 1995] estas implementações estão descritas de forma mais detalhada.

⁹ *X-based Image processing Tools an Environmemt*, [Lønnestad, 1992].

- *rotationx*, valor em grau da rotação segundo o eixo *x*. Por defeito, é igual a *0.0* grau.
- *rotationy*, valor em grau da rotação segundo o eixo *y*. Por defeito, é igual a *0.0* grau.
- *rotationz*, valor em grau da rotação segundo o eixo *z*. Por defeito, é igual a *0.0* grau.
- *translationx*, translação segundo o eixo *x*. Por defeito, é igual a *0.0*.
- *translationy*, translação segundo o eixo *y*. Por defeito, é igual a *0.0*.
- *translationz*, translação segundo o eixo *z*. Por defeito, é igual a *0.0*.
- ✓ Leitura dos argumentos passados à implementação pela linha de comando:
 - *f*, distância focal efectiva.
 - *dx*, distância entre centros dos elementos sensores vizinhos na direcção *x*.
 - *dy*, distância entre centros dos sensores CCD vizinhos na direcção *y*.
 - *sx*, do tipo real, factor de incerteza horizontal.
 - *k*, do tipo real, factor de distorção radial da lente.
 - *zwplane*, coordenada *z* 3D mundo do plano sobre o qual se pretende determinar a área visível pela câmara.
- ✓ Chamada do módulo *compute_mr*.
- ✓ Apresentação das componentes da matriz de rotação 3D resultante.
- ✓ Chamada do módulo *frame_world*.
- ✓ Apresentação das coordenadas dos quatro vértices que definem a área visível pela câmara sobre o plano especificado.
- ✓ Comunicação da existência de erros em argumentos de entrada ou ao longo do decurso da execução.

Este módulo apresenta como restrições quanto à sua utilização:

- ⊗ O valor especificado para o factor de incerteza horizontal deve ser diferente de zero.
 - ⊗ Os valores especificados para a distância entre centros dos elementos sensores vizinhos na direcção *x* e para a distância entre centros dos sensores CCD vizinhos na direcção *y*, devem ser maiores do que zero.
 - ⊗ Os valores especificados para as rotações segundo os três eixos principais deverão estar definidos em grau e compreendidos entre *-360* e *360*.
- b) **Módulo *compute_mr***, responsável pela determinação dos elementos da matriz de rotação 3D da transformação geométrica do sistema de coordenadas mundo para o sistema câmara.
- c) **Módulo *frame_world***, responsável pela determinação das coordenadas 3D mundo *x* e *y*, de um ponto, a partir das suas coordenadas na memória *frame* e da sua coordenada *z* no mesmo sistema mundo.

Na *Fig. 13*, estão representados os módulos que constituem esta implementação assim como a relação existente entre estes.

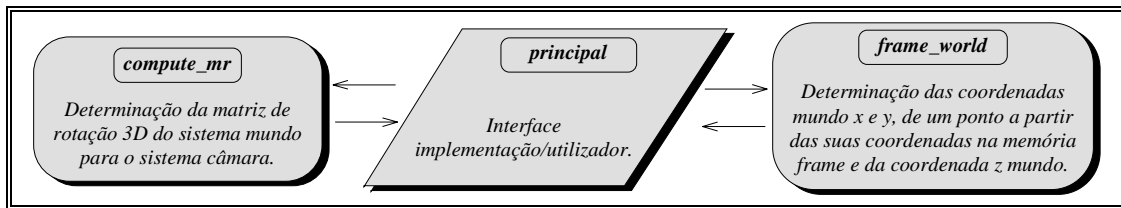


Fig. 13 - Módulos integrantes da implementação *simspace*.

3.2 - Simulador de uma dada câmara móvel para segmentos de recta definidos no espaço tridimensional: *simoca*

Neste ponto é apresentada uma implementação para simulação de uma dada câmara móvel, para segmentos de recta no espaço tridimensional cujas coordenadas 3D dos pontos iniciais e finais são especificadas. Com esta implementação torna-se possível obter as coordenadas imagem dos pontos inicial e final de cada segmento de recta definido no espaço tridimensional, para uma dada câmara e para sucessivas posições e orientações. Internamente, esta implementação inclui um algoritmo para a determinação das partes visíveis de cada segmento de recta pela câmara, em cada uma das suas sucessivas posições e orientações.

O algoritmo base desta implementação pode ser descrito do seguinte modo:

- I) Para cada posição e orientação da câmara desejada executar:
 - a) Leitura de um ficheiro de entrada da posição do referencial câmara relativamente ao referencial mundo e do nome do ficheiro de saída respectivo.
 - b) Determinação da matriz de rotação envolvida.
 - c) Para cada segmento de recta no espaço 3D executar:
 - i) Leitura das coordenadas 3D dos pontos inicial e final do segmento de recta, a partir de um segundo ficheiro de entrada.
 - ii) Determinação das coordenadas na memória *frame* dos pontos inicial e final do segmento de recta respectivo.
 - iii) Determinação das coordenadas dos pontos inicial e final da parte visível pela câmara, para a respectiva posição e orientação do segmento de recta.
 - iv) Conversão das coordenadas reais dos pontos inicial e final do segmento de recta para inteiras.
 - v) Escrita das coordenadas dos pontos inicial e final do segmento de recta em causa, no ficheiro de saída.

Esta implementação é constituída pelos módulos *principal*, *world_frame*, *compute_mr*, *clip2d*, *clipt*, *double_int* e *simulation*. Apresenta-se de seguida a descrição de cada um destes módulos.

- a) **Módulo *principal***, responsável pela interface implementação/utilizador. A seu cargo está:
 - ✓ Leitura dos argumentos opcionais:
 - *xcenterbuf*, coordenada do centro da imagem na memória *frame* segundo a direcção *x*. Por defeito, é igual a metade da dimensão da memória *frame* segundo a mesma direcção.
 - *ycenterbuf*, coordenada do centro da imagem na memória *frame* segundo a direcção *y*. Por defeito, é igual a metade da dimensão da memória *frame* segundo a mesma direcção.
 - *xsizebuf*, dimensão da memória *frame* segundo a direcção *x*. Por defeito, é igual a 512.
 - *ysizebuf*, dimensão da memória *frame* segundo a direcção *y*. Por defeito, é igual a 512.

- *verbose*, quando é especificado como igual a 1, durante a simulação, o utilizador é informado dos resultados obtidos. Por defeito, é igual a 0.
- ✓ Leitura dos argumentos passados à implementação pela linha de comando:
 - *f*, distância focal efectiva.
 - *dx*, distância entre centros dos elementos sensores vizinhos na direcção *x*.
 - *dy*, distância entre centros dos sensores CCD vizinhos na direcção *y*.
 - *sx*, factor de incerteza horizontal.
 - *k*, factor de distorção radial da lente.
 - *filepoints*, nome do ficheiro de texto que contém as coordenadas 3D dos pontos inicial e final de cada segmento de recta a considerar para a simulação.
 - *filesimulation*, nome do ficheiro de texto que contém a posição e orientação do referencial câmara relativamente ao referencial mundo e o nome do ficheiro de texto para escrita dos resultados obtidos, para cada posição e orientação da câmara a considerar.
- ✓ Chamada do módulo *simulation*.
- ✓ Comunicação da existência de erros em argumentos de entrada ou ao longo do decurso da execução.

O ficheiro de entrada *filepoints* contém as coordenadas 3D dos pontos inicial e final de cada segmento de recta a considerar e deverá ter a forma: $x_{w_k}^i y_{w_k}^i z_{w_k}^i x_{w_k}^f y_{w_k}^f z_{w_k}^f x_{w_j}^i y_{w_j}^i z_{w_j}^i x_{w_j}^f y_{w_j}^f z_{w_j}^f \dots$

O ficheiro de entrada *filesimulation* deverá conter:

- as três rotações (em grau) e as três translações da transformação geométrica do sistema de coordenadas mundo para o sistema câmara;
- o nome do ficheiro de texto para escrita dos resultados correspondentes, para cada posição e orientação da câmara a considerar.

A forma deste ficheiro de entrada deverá ser: $R_{(x, \phi)}^i R_{(y, \theta)}^i R_{(z, \psi)}^i t_x^i t_y^i t_z^i \text{ficheiro}^i R_{(x, \phi)}^j R_{(y, \theta)}^j R_{(z, \psi)}^j t_x^j t_y^j t_z^j \text{ficheiro}^j \dots$

A forma do ficheiro *ficheiro*^{*i*} de saída dos resultados respectivos é: $x_f^i y_f^i x_f^f y_f^f \dots$

Como restrições na utilização deste módulo têm-se:

- ⊗ O valor especificado para o factor de incerteza horizontal deve ser diferente de zero.
- ⊗ Os valores especificados para a distância entre centros dos elementos sensores vizinhos na direcção *x* e para a distância entre centros dos sensores CCD vizinhos na direcção *y*, devem ser maiores do que zero.

- b) **Módulo *world_frame***, responsável pela determinação das coordenadas na memória *frame* de um ponto a partir das suas coordenadas 3D mundo.
- c) **Módulo *compute_mr***, responsável pela determinação dos elementos da matriz de rotação 3D da transformação geométrica do sistema de coordenadas mundo para o sistema câmara.
- d) **Módulos *clip2d* e *clipt***, responsáveis pela determinação das coordenadas 2D dos pontos inicial e final, da parte visível de um dado segmento de recta através de um dado rectângulo de visibilidade segundo o algoritmo paramétrico de Liang-Barsky¹⁰.

¹⁰ Este algoritmo foi adoptado devido à sua elevada eficiência; ver, por exemplo [Foley, 1991].

- e) *Módulo `double_int`*, responsável pela conversão de coordenadas reais de um dado ponto para inteiras, estando estas coordenadas inteiras limitadas a certos valores.
- f) *Módulo `simulation`*, responsável pela simulação propriamente dita. Durante a sua execução este módulo utiliza os módulos *`compute_mr`*, *`world_frame`*, *`clip2d`* e *`double_int`*.

Na Fig. 14 estão representados os módulos que constituem esta implementação assim como a relação existente entre estes.

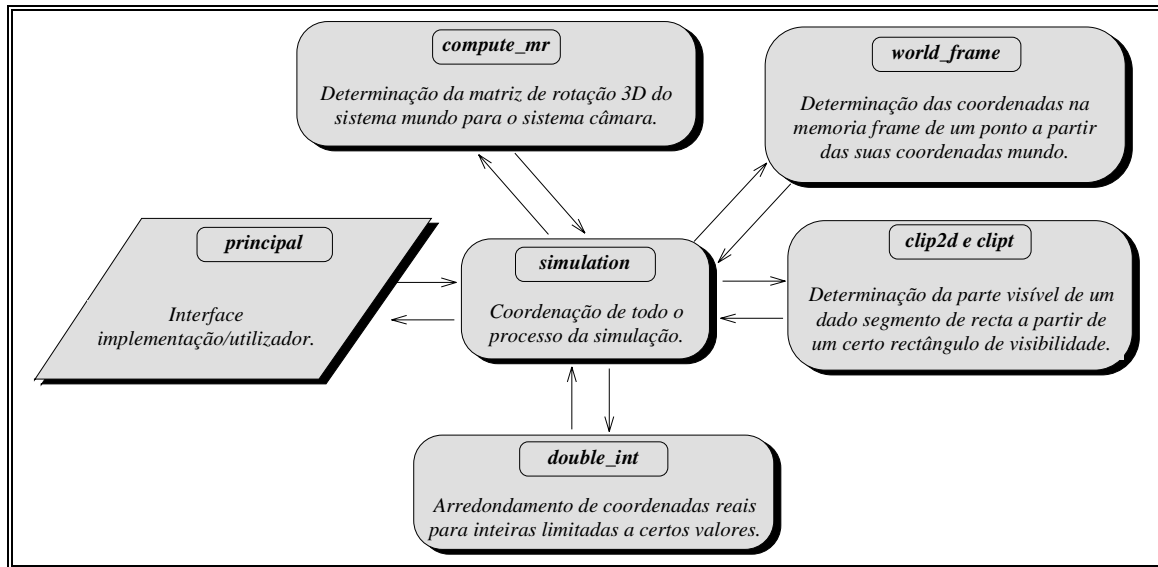


Fig. 14 - Módulos integrantes da implementação *simoca*.

3.3 - Obtenção de informação tridimensional a partir do movimento de uma câmara: *deep*

Neste ponto é apresentada a implementação da abordagem global induzida neste capítulo para o seguimento de segmentos de recta em imagens sucessivas e consequente obtenção de informação tridimensional. Os resultados obtidos por esta implementação são apresentados na forma de uma imagem constituída por uma banda para cada intervalo¹¹, de largura definida pelo utilizador, de posições e orientações sucessivas da câmara. Em cada uma destas bandas, aparecem desenhados os diferentes segmentos de recta emparelhados, com os pontos que os constituem com um nível de cinzento relacionado com a sua profundidade. Existe, do mesmo modo, a possibilidade de os resultados obtidos serem escritos num ficheiro de texto; neste caso, serão escritas as coordenadas na memória *frame* e as coordenadas 3D dos pontos inicial e final de cada entidade emparelhada.

Esta implementação é constituída pelos módulos referidos na *Tab. I*. Cada um dos grupos definidos é constituído por módulos que são comuns a um dado contexto de aplicação. Assim, o grupo **I** contém os módulos mais particulares a esta implementação, o **II** os relacionados com funções de visão tridimensional, o **III** os relacionados com resolução de sistemas lineares sobredeterminados pelo método dos mínimos quadrados utilizando decomposição em valor único¹², o **IV** os relacionados

¹¹ Como já referido, para o emparelhamento ser obtido com maior êxito as imagens sucessivas não devem estar muito espaçadas; no entanto, para o erro resultante da obtenção das coordenadas tridimensionais ser reduzido a disparidade estereoscópica deve ser razoável; assim, para ultrapassar esta contradição optou-se por considerar todas as imagens para o seguimento e determinar as coordenadas tridimensionais após um certo número de imagens de intervalo.

¹² Esta decomposição pode ser consideravelmente mais lenta do que a resolução das equações normais equivalentes. No entanto, tem a grande vantagem, pelo menos teórica, de corresponder a soluções de aproximação para as variáveis independentes do modelo de melhor qualidade; isto é, de menor erro. Desta forma, em problemas numéricos, esta decomposição apresenta vantagens sobre a eliminação de Gauss e sobre a decomposição *LU*, sendo a decomposição de

com as entidades utilizadas, o **V** os relacionados com os estados das entidades utilizados pelos filtros de Kalman, o **VI** os relacionados com operações de matrizes, o **VII** os relacionados com listas ligadas e o **VIII** os relacionados com gestão de memória. Apresenta-se de seguida apenas a descrição de cada um dos módulos dos grupos **I** e **II**, pois são os mais directamente ligados à abordagem utilizada.

<i>Grupo I</i>	<i>Grupo II</i>	<i>Grupo III</i>	<i>Grupo IV</i>
<i>principal, maching, mahalanobis, measure, ellipse, compute_deep, dda_deep</i>	<i>compute_mpp, frame_undistorted, match_line, stereo_eq</i>	<i>svdfit, svbksb, svdcmp</i>	<i>create_line, delete_line, print_line, create_mpp, delete_mpp</i>
<i>Grupo V</i>	<i>Grupo VI</i>	<i>Grupo VII</i>	<i>Grupo VIII</i>
<i>InitStateSpace, RemoveStateSpace, statePredictCovrpos, statePredict, stateUpdate, stateDistance, copyState</i>	<i>create_matrix, delete_matrix, print_matrix, matrix_opr, matrix_mult, matrix_wsqr, matrix_invert, matrix_eig</i>	<i>create_list, delete_list, insert_cell_list, delete_cell_list</i>	<i>memory_get, vector, matrix, free_vector, free_matrix</i>

Tab. 1 - Módulos que constituem a implementação *deep*.

a) **Módulo principal**, responsável pela interface implementação/utilizador. A seu cargo está:

✓ Leitura dos argumentos opcionais:

- *title*, título da imagem de saída para representação dos resultados obtidos pela implementação. Por defeito, é “*Deep extracted*”.
- *maxDeep*, máxima profundidade a considerar na representação desta na imagem de saída. Por defeito, é igual a *2800.0*.
- *minDeep*, mínima profundidade a considerar na representação desta na imagem de saída. Por defeito, é igual a *0.0*.
- *nFrameDeep*, número de imagens de intervalo a considerar para a determinação das coordenadas 3D. Por defeito, é igual a *0*, isto é, usa imagens contíguas.
- *xSize*, dimensão da memória *frame* segundo a direcção *x*. Por defeito, é igual a *512*.
- *ySize*, dimensão da memória *frame* segundo a direcção *y*. Por defeito, é igual a *512*.
- *sx*, factor de incerteza horizontal. Por defeito, é igual a *0.710935*.
- *focalDist*, distância focal efectiva. Por defeito, é igual a *50.0*.
- *radialLensDist*, factor de distorção radial da lente. Por defeito, é igual a *0.001*.
- *dx*, distância entre centros dos elementos sensores vizinhos na direcção *x*. Por defeito, é igual a *8.37765957e-3*.
- *dy*, distância entre centros dos sensores CCD vizinhos na direcção *y*. Por defeito, é igual a *8.07560136e-3*.
- *xCenterBuf*, coordenada do centro da imagem na memória *frame* segundo a direcção *x*. Por defeito, é igual a metade da dimensão da memória *frame* segundo a mesma direcção.
- *yCenterBuf*, coordenada do centro da imagem na memória *frame* segundo a direcção *y*. Por defeito, é igual a metade da dimensão da memória *frame* segundo a mesma direcção.

utilização preferencial para a resolução de problemas de aproximação linear pelo método dos mínimos quadrados [Press, 1992].

- *maxDistMedP*, máxima distância possível entre a posição do ponto médio de uma entidade e a prevista pelo filtro de Kalman respectivo, para a entidade em consideração ser declarada como candidata ao emparelhamento. Por defeito, é igual a 100.0.
 - *maxDifLength*, máxima diferença possível entre o comprimento de uma entidade e o previsto pelo filtro de Kalman respectivo, para a entidade em consideração ser declarada como candidata ao emparelhamento. Por defeito, é igual a 20.0.
 - *maxDifAng*, máxima diferença possível entre a direcção de uma entidade e a prevista pelo filtro de Kalman respectivo, para a entidade em consideração ser declarada como candidata ao emparelhamento. Por defeito, é igual a 10.0.
 - *outputFile*, nome do ficheiro de saída para escrita dos resultados obtidos. Por defeito é "NULL", o que implica a não abertura do ficheiro.
 - *verbose*, quando é especificado como igual a 1, durante a execução, o utilizador é informado de como esta está a decorrer, assim como dos resultados obtidos. Por defeito, o utilizador não é informado.
- ✓ Leitura dos argumentos passados à implementação pela linha de comando:
- *inputFile*, nome do ficheiro de texto que contém os nomes dos ficheiros para cada posição e orientação da câmara a considerar pela implementação. Cada um destes ficheiros deverá conter a posição e orientação do referencial câmara relativamente ao referencial mundo e as coordenadas na memória *frame* dos pontos inicial e final de cada segmento de recta.
 - *outputImage*, nome da imagem de saída para representação dos resultados obtidos.
- ✓ Chamada da função *matching*.
- ✓ Comunicação da existência de erros em argumentos de entrada ou ao longo do decurso da execução.

O ficheiro de entrada *inputFile* deverá ser constituído pelos nomes dos ficheiros que contêm a posição e orientação do referencial mundo relativamente ao referencial câmara e as coordenadas na memória *frame* dos pontos inicial e final de cada segmento de recta, para cada posição e orientação da câmara a considerar. Este ficheiro deverá ter a forma: *ficheiroⁱ ficheiro^j...* Cada posição e orientação da câmara, terá então um ficheiro associado. Cada um destes ficheiros contém as três rotações e as três translações da transformação geométrica 3D do sistema de coordenadas mundo para o sistema câmara e as coordenadas na memória *frame* dos pontos inicial e final de cada segmento de recta a considerar. A forma destes ficheiros deverá ser então: $R_{(x, \phi)} R_{(y, \theta)} R_{(z, \psi)} t_x t_y t_z x_{f_k}^i y_{f_k}^i x_{f_k}^f y_{f_k}^f x_{f_l}^i y_{f_l}^i x_{f_l}^f y_{f_l}^f \dots$

O ficheiro de saída dos resultados obtidos pela implementação, quando desejado pelo utilizador, terá a forma: $x_{f_k}^i y_{f_k}^i x_{w_k}^i y_{w_k}^i z_{w_k}^i x_{f_k}^f y_{f_k}^f x_{w_k}^f y_{w_k}^f z_{w_k}^f x_{f_l}^i y_{f_l}^i x_{w_l}^i y_{w_l}^i z_{w_l}^i x_{f_l}^f y_{f_l}^f x_{w_l}^f y_{w_l}^f z_{w_l}^f \dots$

Como restrições quanto à utilização deste módulo têm-se:

- ⊗ Os valores especificados para o factor de incerteza horizontal e para a distância focal efectiva, devem ser diferentes de zero.
- ⊗ Os valores especificados para a distância entre centros dos elementos sensores vizinhos na direcção *x* e para a distância entre centros dos sensores CCD vizinhos na direcção *y*, devem ser maiores do que zero.
- ⊗ Os valores especificados para as dimensões da memória *frame* segundo as direcções *x* e *y*, devem ser maiores do que zero.
- ⊗ O ficheiro de entrada *inputFile* deve conter mais do que um nome.

- b) **Módulo *matching***, responsável por toda a gestão dos restantes módulos que constituem esta implementação. Assim, este módulo coordena todo o processo de seguimento 2D e extracção de coordenadas 3D segundo a abordagem global induzida neste capítulo.

Este módulo apresenta as seguintes restrições quanto à sua utilização:

⊗ Os ficheiros de entrada devem conter pelos menos duas linhas, isto é, uma constituída pelos valores das três rotações e das três translações da transformação geométrica 3D do sistema de coordenadas mundo para o sistema câmara, e outra constituída pelas coordenadas na memória *frame* dos pontos inicial e final de pelo menos uma entidade.

⊗ A existência de memória suficiente.

- c) **Módulo *mahalanobis***, responsável pela determinação da distância normalizada de Mahalanobis entre dois elementos para uma única característica escalar.
- d) **Módulo *measure***, responsável por efectuar a medição das variâncias da posição do ponto médio, do comprimento e de direcção, no interior da elipse de emparelhamento. Do mesmo modo, este módulo é responsável pela determinação do número de entidades cujo ponto médio é interior à referida elipse. Este módulo, durante a sua execução utiliza os módulos *elipse* e *mahalanobis*.
- e) **Módulo *elipse***, responsável por determinar se um determinado ponto é ou não interior a uma dada elipse.
- f) **Módulo *compute_deep***, responsável pela determinação das coordenadas 3D dos pontos inicial e final de uma dada entidade. Durante a sua execução este módulo utiliza os módulos *match_line*, *stereo_eq* e *dda_deep*. Quando é desejado pelo utilizador, este módulo é também responsável pela escrita das coordenadas na memória *frame* e das coordenadas 3D dos pontos inicial e final da mesma entidade no ficheiro de saída.
- g) **Módulo *dda_deep***, responsável pelo desenho de segmentos de recta numa banda da imagem de saída. Trata-se de uma implementação do algoritmo de Bresenham¹³, também designado por versão meio ponto do segmento. A cada ponto que pertence à respectiva entidade é atribuído na banda da imagem de saída o valor de cinzento correspondente à sua profundidade, sendo esta determinada aproximadamente por interpolação linear das profundidades dos seus pontos inicial e final. Para uma melhor visualização a gama de cinzento disponível não é toda utilizada; assim, um certo número de elementos inferiores e superiores desta gama não são utilizados.
- h) **Módulo *compute_mpp***, responsável pela determinação da matriz homogénea da transformação geométrica 3D do sistema de coordenadas mundo para o sistema no plano imagem.
- i) **Módulo *frame_undistorted***, responsável pela determinação das coordenadas não distorcidas de um dado ponto, a partir das respectivas coordenadas na memória *frame*.
- j) **Módulo *match_line***, responsável pela determinação da linha epipolar num plano imagem de um determinado ponto, a partir das duas matrizes de transformação do sistema de coordenadas mundo para o sistema do plano imagem, e das suas coordenadas não distorcidas no outro plano imagem.
- k) **Módulo *stereo_eq***, responsável pela determinação das coordenadas 3D de um determinado ponto a partir das duas matrizes de transformação do sistema de coordenadas mundo para o

¹³ Este algoritmo tem como grande vantagem, a não utilização de multiplicações, de divisões e de funções de arredondamento. Ver, por exemplo [Foley, 1991].

sistema do plano imagem, e das suas coordenadas não distorcidas nos respectivos planos imagem. Durante a sua execução este módulo utiliza os módulos *matrix*, *vector*, *free_matrix*, *free_vector* e *svdfit*.

Na Fig. 15 estão representados os grupos que constituem esta implementação, assim como a relação existente entre estes.

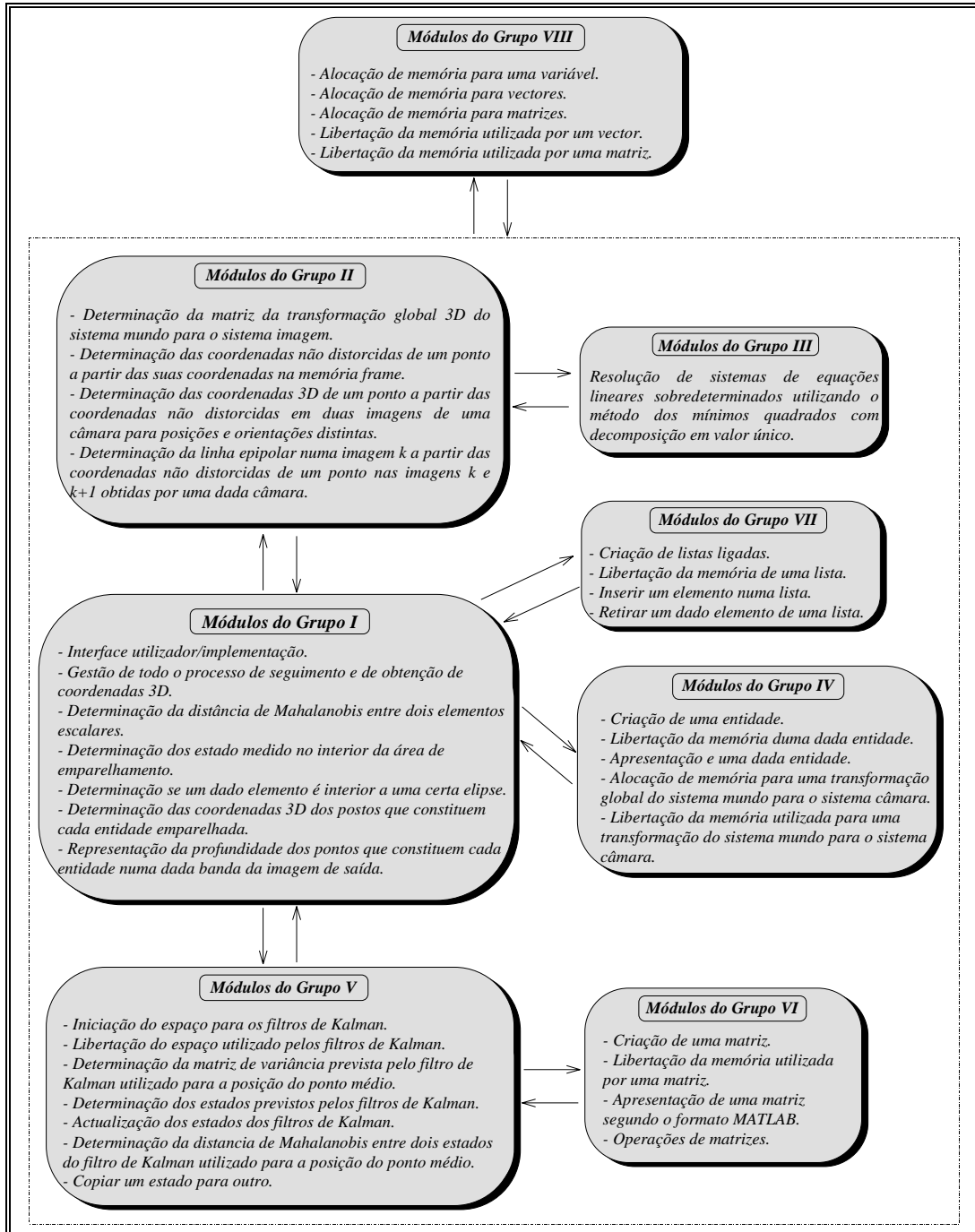


Fig. 15 - Grupos integrantes da implementação deep.

3.4 - Extracção de coordenadas 3D de pontos a partir das suas coordenadas em duas imagens distintas: *compdeep*

Neste ponto é apresentada uma implementação para extracção de coordenadas 3D de pontos cujas coordenadas 2D, em duas imagens obtidas por uma mesma câmara, são especificadas. A técnica para a extracção de coordenadas 3D utilizada nesta implementação é baseada no princípio da triangulação estereoscópica.

O algoritmo base desta implementação pode ser descrito do seguinte modo:

- I) Até o utilizador especificar o valor 9999 para a coordenada na memória *frame* segundo a direcção *x*, para uma dado ponto e para a primeira posição e orientação da câmara a considerar, executar ciclicamente:
 - a) Interrogação ao utilizador das coordenadas na memória *frame* para o ponto, segundo as direcções *x* e *y*, para a primeira posição e orientação da câmara a considerar.
 - b) Determinação das coordenadas imagem não distorcidas para o ponto, segundo as direcções *x* e *y*, para a primeira posição e orientação da câmara a considerar.
 - c) Interrogação ao utilizador das coordenadas na memória *frame* para o ponto, segundo as direcções *x* e *y*, para a segunda posição e orientação da câmara a considerar.
 - d) Determinação das coordenadas não distorcidas para o ponto, segundo as direcções *x* e *y*, para a segunda posição e orientação da câmara a considerar.
 - e) Apresentação das coordenadas não distorcidas para o ponto, segundo as direcções *x* e *y*, nos planos imagem da câmara, para a primeira e segunda posição e orientação a considerar.
 - f) Determinação das coordenadas 3D para o respectivo ponto.
 - g) Apresentação das coordenadas 3D para o respectivo ponto.

Esta implementação é constituída pelos módulos referidos na *Tab. II*.

<i>Grupo I</i>	<i>Grupo II</i>	<i>Grupo III</i>	<i>Grupo IV</i>
<i>principal</i>	<i>compute_mpp,</i> <i>frame_undistorted,</i> <i>stereo_eq</i>	<i>svdfit,</i> <i>svbksb,</i> <i>svdcmp</i>	<i>vector,</i> <i>matrix,</i> <i>free_vector,</i> <i>free_matrix</i>

Tab. II - Módulos que constituem a implementação compdeep

Cada um dos grupos definidos, é constituído por módulos que são comuns a uma dado contexto. Assim, o grupo **I** contém o módulo mais particular a esta implementação, o **II** os relacionados com funções de visão tridimensional, o **III** os relacionados com resolução de sistemas lineares sobredeterminados utilizando decomposição em valor único, e o **IV** os relacionados com gestão de memória. Apresenta-se de seguida apenas a descrição do módulo *principal*.

Módulo principal, responsável pela interface implementação/utilizador. A seu cargo está:

- ✓ Leitura dos argumentos opcionais:
 - *sx*, factor de incerteza horizontal. Por defeito, é igual a 0.710935.
 - *dx*, distância entre centros dos elementos sensores vizinhos na direcção *x*. Por defeito, é igual a 8.37765957e-3.
 - *dy*, distância entre centros dos sensores CCD vizinhos na direcção *y*. Por defeito, é igual a 8.07560136e-3.
 - *xCenterBuf*, coordenada do centro da imagem na memória *frame* segundo a direcção *x*. Por defeito, é igual a 256.

- *yCenterBuf*, coordenada do centro da imagem na memória *frame* segundo a direcção *y*. Por defeito, é igual a 256.
- ✓ Leitura dos argumentos passados à implementação pela linha de comando:
 - *focalDist*, distância focal efectiva.
 - *radialLensDist*, factor de distorção radial da lente.
- ✓ Interrogação ao utilizador dos valores, em grau, das três rotações 3D e dos valores das três translações 3D segundo os três eixos principais, da transformação geométrica do sistema de coordenadas mundo para o sistema câmara, para a primeira posição e orientação da câmara a considerar.
- ✓ Interrogação ao utilizador dos valores, em grau, das três rotações 3D e dos valores das três translações 3D segundo os três eixos principais, da transformação geométrica do sistema de coordenadas mundo para o sistema câmara, para a segunda posição e orientação da câmara a considerar.
- ✓ Determinação da transformação geométrica do sistema de coordenadas mundo para o sistema do plano imagem da câmara, para a primeira posição e orientação desta, utilizando para o efeito o módulo *compute_mpp*.
- ✓ Apresentação da matriz da transformação geométrica do sistema de coordenadas mundo para o sistema do plano imagem da câmara, obtida para a primeira posição e orientação.
- ✓ Determinação da transformação geométrica do sistema de coordenadas mundo para o sistema do plano imagem da câmara, para a segunda posição e orientação desta, utilizando para o efeito o módulo *compute_mpp*.
- ✓ Apresentação da matriz da transformação geométrica do sistema de coordenadas mundo para o sistema do plano imagem da câmara, obtida para a segunda posição e orientação.
- ✓ Interrogação ao utilizador das coordenadas na memória *frame* para o ponto, segundo as direcções *x* e *y*, para a primeira posição e orientação da câmara a considerar.
- ✓ Determinação das coordenadas imagem não distorcidas para o ponto, segundo as direcções *x* e *y*, para a primeira posição e orientação da câmara a considerar, utilizando para o efeito o módulo *frame_undistorted*.
- ✓ Interrogação ao utilizador das coordenadas na memória *frame* para o ponto, segundo as direcções *x* e *y*, para a segunda posição e orientação da câmara a considerar.
- ✓ Determinação das coordenadas imagem não distorcidas para o ponto, segundo as direcções *x* e *y*, para a segunda posição e orientação da câmara a considerar.
- ✓ Apresentação das coordenadas imagem não distorcidas para o ponto, segundo as direcções *x* e *y*, nos planos imagem da primeira e da segunda posição e orientação da câmara.
- ✓ Determinação das coordenadas 3D para o respectivo ponto, utilizando para o efeito o módulo *stereo_eq*.
- ✓ Apresentação das coordenadas 3D obtidas.
- ✓ Comunicação da existência de erros em argumentos de entrada ou ao longo do decurso da execução.

Como restrições quanto à utilização deste módulo têm-se:

- ☒ Os valores especificados para a distância entre centros dos elementos sensores vizinhos na direcção *x* e para a distância entre centros dos sensores CCD vizinhos na direcção *y*, devem ser maiores do que zero.

- ⊗ Os valores especificados para a distância focal efectiva e para o factor de incerteza horizontal, devem ser diferentes de zero.

Na Fig. 16, estão representados os grupos que constituem esta implementação, assim como a relação existente entre estes.

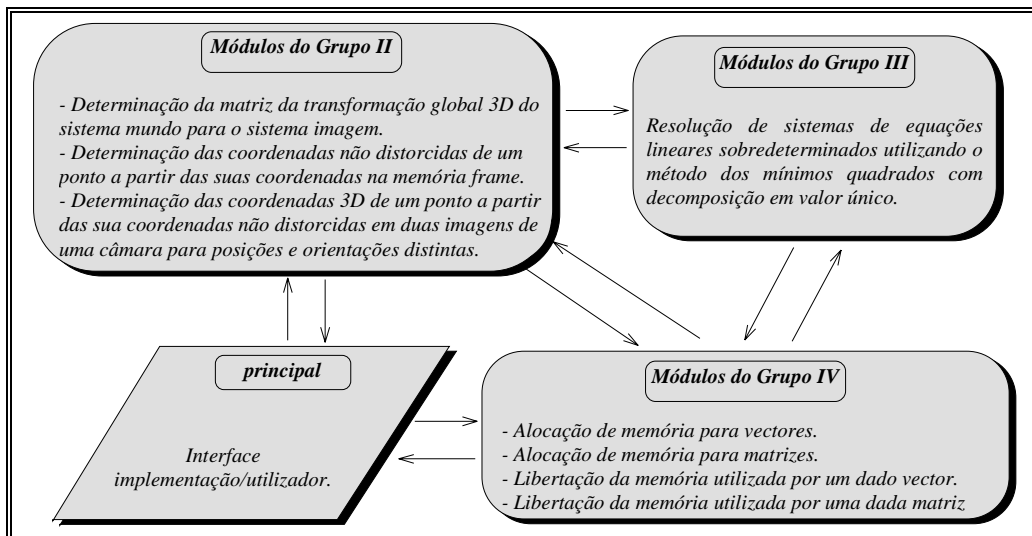


Fig. 16 - Módulos integrantes da implementação *compdeep*.

4 - Alguns resultados experimentais obtidos pelas implementações desenvolvidas

Nesta secção, são apresentados alguns resultados experimentais¹⁴ obtidos pelas várias implementações descritas de forma sumária na secção anterior. Assim serão apresentados:

- um exemplo de resultados experimentais obtido na utilização da implementação *simspace*, para a determinação da área visível por uma dada câmara, sobre um dado plano;
- um exemplo de resultados experimentais obtidos na utilização da implementação *simoca*, para a simulação de uma câmara móvel, para segmentos de recta definidos no espaço tridimensional;
- um exemplo de resultados experimentais obtidos na utilização da implementação *deep*, para a obtenção de informação tridimensional a partir do movimento de uma câmara;
- um exemplo de resultados experimentais obtidos pela implementação *compdeep*, para a extracção de coordenadas 3D de pontos, a partir das suas coordenadas 2D em duas imagens distintas.

4.1 - Utilizando a implementação *simspace* para a determinação da área visível por uma dada câmara sobre um dado plano

Neste ponto é apresentado um exemplo de resultados experimentais obtido pela implementação *simspace*, apresentada de forma sumária no ponto 3.1. Os parâmetros de entrada utilizados estão referidos na Tab. III, sendo representados na Fig. 17 os resultados obtidos.

¹⁴ No capítulo V são apresentados outros resultados obtidos, utilizando sequências de imagens reais, pela implementação *deep*.

<i>xcenterbuf</i>	<i>ycenterbuf</i>	<i>xsizebuf</i>	<i>ysizebuf</i>	<i>rotationx</i>	<i>rotationy</i>
256	256	512	512	-30.0	-30.0
<i>rotationz</i>	<i>translationx</i>	<i>translationy</i>	<i>translationz</i>	<i>f</i>	<i>dx</i>
-30.0	100.0	100.0	150.0	50.0	8,37765957e-3
	<i>dy</i>	<i>sx</i>	<i>k</i>	<i>zwplane</i>	
	8.075601357e-3	0.710935	0.01	2000.0	

Tab. III - Parâmetros de entrada para a implementação *simspace* utilizados no exemplo apresentado .

Rotation matrix:		
r1=0.75	r2=-0.433013	r3=0.5
r4=0.649519	r5=0.625	r6=-0.433013
r7=-0.125	r8=0.649519	r9=0.75.
Maximum world space visible by the camera in the plane with z=2000:		
First corner: x=-621.643, y=1674.02		
Second corner: x=-478.956, y=1932.45		
Third corner: x=-469.703, y=1524.26		
Fourth corner: x=-325.393, y=1763.82.		

Fig. 17 - Resultados obtidos pela implementação *simspace* com os parâmetros da Tab. III.

4.2 - Utilizando a implementação *simoca* para a simulação de uma câmara móvel, para segmentos de recta definidos no espaço tridimensional

Neste ponto é apresentado um exemplo de resultados obtidos pela implementação *simoca*, apresentada de forma sumária no ponto 3.2. O objectivo deste exemplo foi utilizar esta implementação para gerar sinteticamente uma sequência de 12 imagens constituídas por 10 linhas, para ser posteriormente utilizada como teste da implementação *deep*. Assim, foram utilizados os parâmetros de entrada apresentados na Tab. IV e os ficheiros com o conteúdo apresentado na Fig. 18.

<i>xcenterbuf</i>	<i>ycenterbuf</i>	<i>xsizebuf</i>	<i>ysizebuf</i>	<i>verbose</i>	<i>rotationy</i>	<i>f</i>
256	256	512	512	0	30.0	50.0
<i>dx</i>	<i>dy</i>	<i>sx</i>	<i>k</i>	<i>filepoints</i>	<i>filesimulation</i>	
8,37765957e-3	8.075601357e-3	0.710935	0.001	3d.txt	3ds.txt	

Tab. IV - Parâmetros de entrada para a implementação *simoca* utilizados no exemplo apresentado .

Conteúdo do ficheiro 3d.txt	Conteúdo do ficheiro 3ds.txt
0.0 0.0 2000.0 50.0 50.0 2000.0	0.0 0.0 0.0 -55.0 0.0 0.0
50.0 50.0 2000.0 0.0 80.0 2000.0	out1.txt
0.0 80.0 2000.0 0.0 0.0 2000.0	0.0 0.0 0.0 -55.0 -5.0 -10.0
20.0 20.0 2500.0 60.0 60.0 2500.0	out2.txt
60.0 60.0 2500.0 60.0 80.0 2500.0	0.0 0.0 0.0 -55.0 -10.0 -20.0
60.0 80.0 2500.0 20.0 60.0 2500.0	out3.txt
20.0 60.0 2500.0 20.0 20.0 2500.0	0.0 0.0 0.0 -55.0 -15.0 -30.0
10.0 10.0 2000.0 80.0 80.0 2500.0	out4.txt
0.0 40.0 2200.0 40.0 55.0 2400.0	0.0 0.0 0.0 -55.0 -20.0 -40.0
40.0 30.0 2100.0 80.0 80.0 2100.0	out5.txt
	0.0 0.0 0.0 -55.0 -25.0 -50.0
	out6.txt
	0.0 0.0 0.0 -55.0 -58.0 -105.0
	out7.txt
	0.0 0.0 0.0 -55.0 -40.0 -70.0
	out8.txt
	0.0 0.0 0.0 -55.0 -45.0 -80.0
	out9.txt
	0.0 0.0 0.0 -55.0 -50.0 -90.0
	out10.txt
	0.0 0.0 0.0 -55.0 -55.0 -100.0
	out11.txt
	0.0 0.0 0.0 -55.0 -55.0 -100.0
	out12.txt
	...

Fig. 18 - Conteúdo dos ficheiros de entrada para a implementação *simoca* utilizados no exemplo apresentado.

A sequência de imagens¹⁵ obtida, está representada nas *Figs. 19 a 30*. Estas imagens foram obtidas a partir do conteúdo dos ficheiros de saída da implementação *simoca*, utilizando-se para tal a implementação *showlines* apresentada no capítulo anterior. De referir que, na 4ª e na 6ª imagem da sequência, não estão representados todos os segmentos de recta obtidos pela implementação *simoca*. Tal facto deveu-se à anulação propositada das coordenadas dos pontos inicial e final de um dos segmentos nos ficheiros respectivos, e teve como único objectivo o de simular a possibilidade real de um qualquer segmento passar de visível a invisível (ou vice-versa) em qualquer passo da sequência, constituindo assim uma certa perturbação no teste da implementação *deep* com esta sequência.

A título de exemplo, está representado na *Fig. 31* o conteúdo do ficheiro de saída obtido neste exemplo para a primeira posição e orientação da câmara.

4.3 - Utilizando a implementação *deep* para a obtenção de informação tridimensional a partir do movimento de uma câmara

Para teste da implementação *deep*, apresentada de forma sumária no ponto 3.3, foi utilizada a sequência de imagens (mais rigorosamente, os ficheiros de texto equivalentes a estas), gerada sinteticamente pela implementação *simoca*, de 12 imagens constituídas por 10 segmentos de recta, e que está representada no ponto anterior pelas *Figs. 19 a 30*. De notar que, como já foi anteriormente referido, nem todas as linhas existem em algumas das imagens consideradas.

Na *Fig. 32* é apresentada a imagem resultante da soma da primeira com a última imagem da sequência de teste, sendo facilmente observado o deslocamento total que as diversas entidades sofreram na referida sequência.

Os parâmetros de entrada utilizados, neste exemplo para a implementação *deep*, estão representados na *Tab. V*.

<i>title</i>	<i>maxDeep</i>	<i>minDeep</i>	<i>nFrameDeep</i>	<i>xSize</i>	<i>ySize</i>	<i>sx</i>
Deep extracted	2800.0	1800.0	0	512	512	0.710935
<i>focalDist</i>	<i>radialLensDist</i>	<i>dx</i>	<i>dy</i>	<i>xCenterBuf</i>	<i>yCenterBuf</i>	<i>maxDistMedP</i>
50.0	0.001	0.00837766	0.0080756	256	256	100
<i>maxDifLength</i>	<i>maxDifAng</i>	<i>outputFile</i>	<i>verbose</i>	<i>inputFile</i>	<i>outputImage</i>	
20	10	teste1.txt	0	teste.txt	teste1.img	

Tab. V - Parâmetros de entrada para a implementação *deep* utilizados no exemplo apresentado .

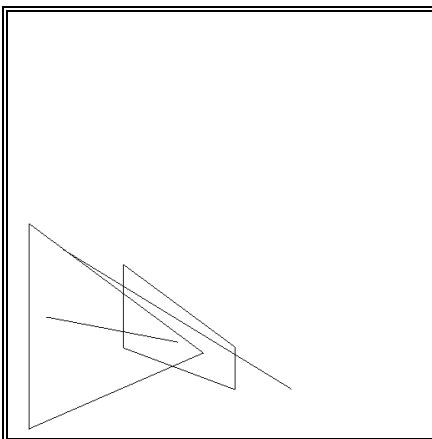


Fig. 19 - 1ª Imagem da sequência obtida.

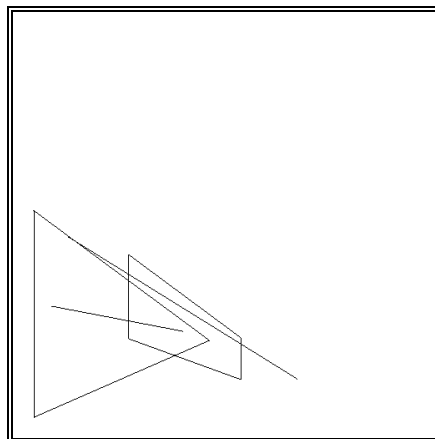


Fig. 20 - 2ª Imagem da sequência obtida.

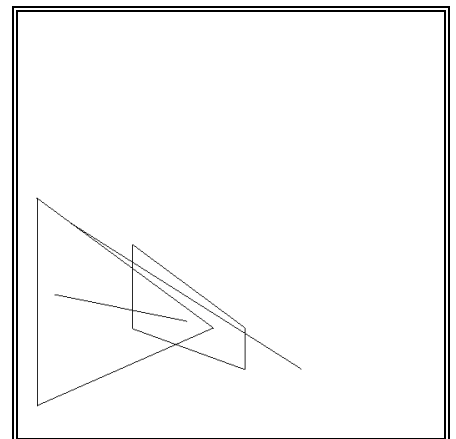


Fig. 21 - 3ª Imagem da sequência obtida.

¹⁵ Todas as imagens representadas nesta secção foram escalonadas de 65% nas suas dimensões.

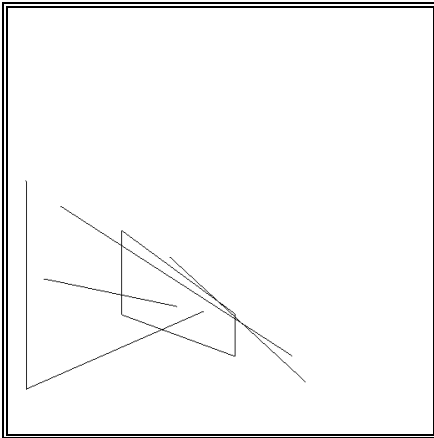


Fig. 22 - 4ª Imagem da sequência obtida.

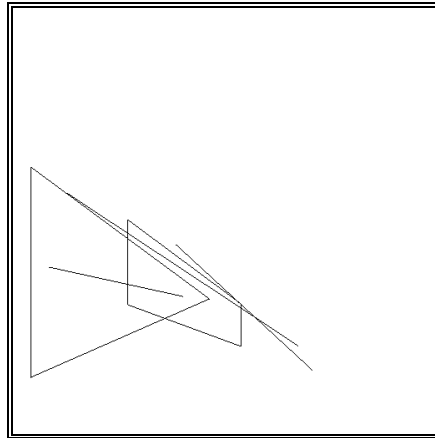


Fig. 23 - 5ª Imagem da sequência obtida.

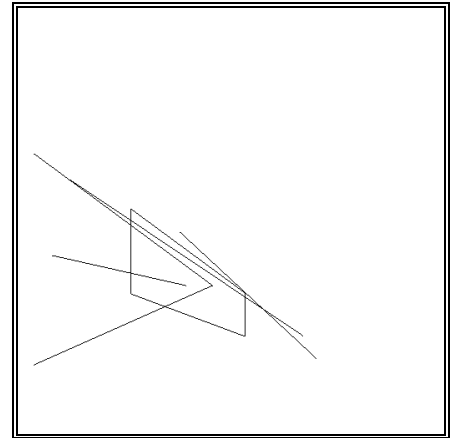


Fig. 24 - 6ª Imagem da sequência obtida.

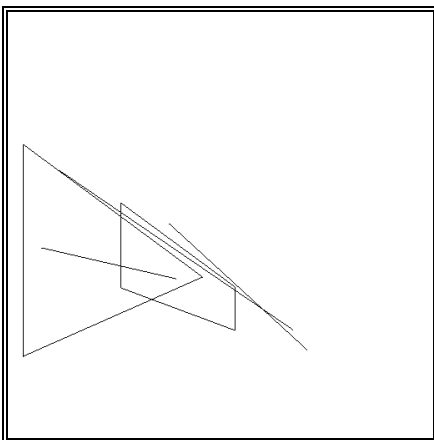


Fig. 25 - 7ª Imagem da sequência obtida.

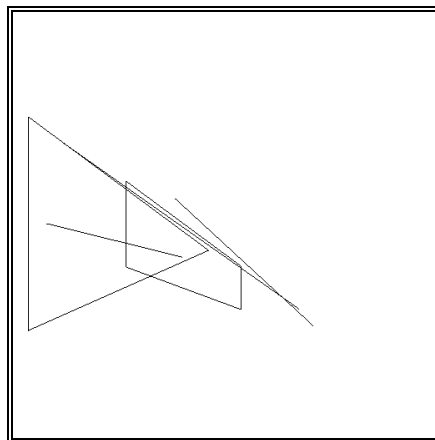


Fig. 26 - 8ª Imagem da sequência obtida.

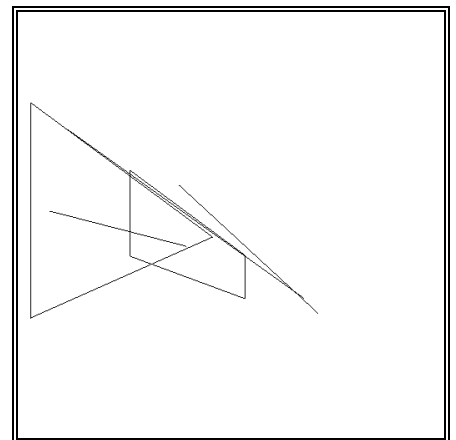


Fig. 27 - 9ª Imagem da sequência obtida.

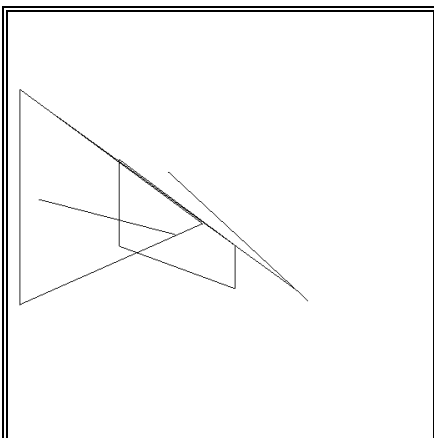


Fig. 28 - 10ª Imagem da sequência obtida.

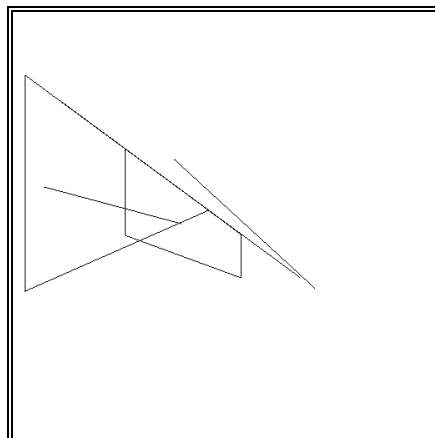


Fig. 29 - 11ª Imagem da sequência obtida.

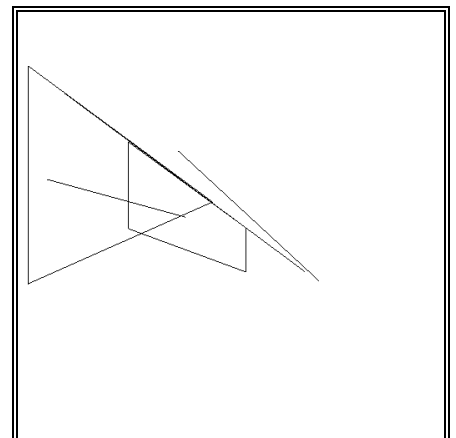


Fig. 30 - 12ª Imagem da sequência obtida.

```

0.000000 0.000000 0.000000 -55.000000 0.000000 0.000000
26 256 235 411
235 411 26 502
26 502 26 256
139 305 273 404
273 404 273 454
273 454 139 404
139 404 139 305
67 287 340 454
47 368 204 398
    
```

Fig. 31- Conteúdo do ficheiro de saída obtido neste exemplo para a primeira posição e orientação da câmara.

Na Fig. 33 está representado o conteúdo do ficheiro de entrada “teste.txt”. O conteúdo de um dos ficheiros de texto, nomeadamente o conteúdo do ficheiro “out1.txt”, já foi mostrado na Fig. 31.

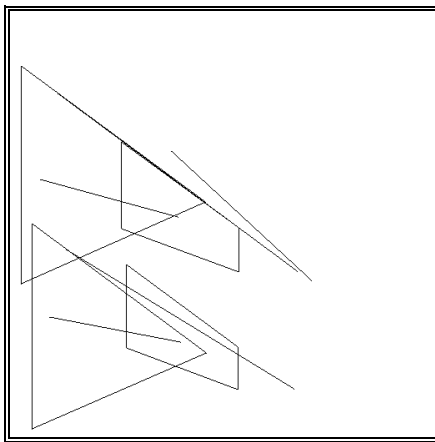


Fig. 32 - Soma da 1ª com a 10ª imagem da sequência de teste.

```

out1.txt
out2.txt
out3.txt
out4.txt
out5.txt
out6.txt
out7.txt
out8.txt
out9.txt
out10.txt
out11.txt
out12.txt
    
```

Fig. 33 - Conteúdo do ficheiro de entrada para o teste apresentado.

As bandas da imagem, resultantes da aplicação da implementação *deep* à sequência de teste, constituídas pelas entidades devidamente emparelhadas no seguimento, estão apresentadas nas Figs. 34 a 44, sendo o nível de cinzento de cada ponto que constitui as mesmas entidades função da sua profundidade (neste caso, a coordenada 3D segundo o eixo z).

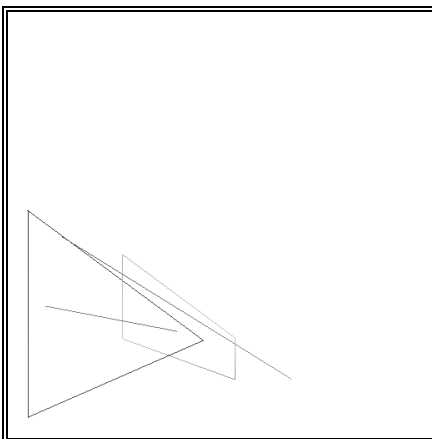


Fig. 34 - 1ª Banda de entidades emparelhadas.

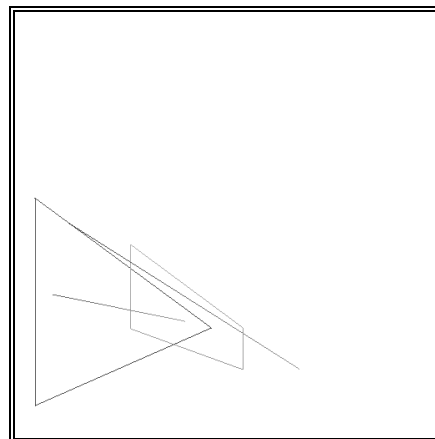


Fig. 35 - 2ª Banda de entidades emparelhadas.

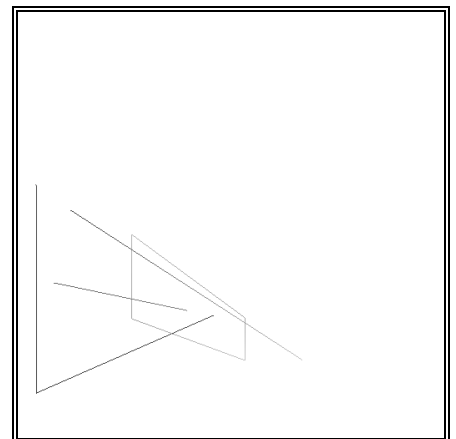


Fig. 36 - 3ª Banda de entidades emparelhadas.

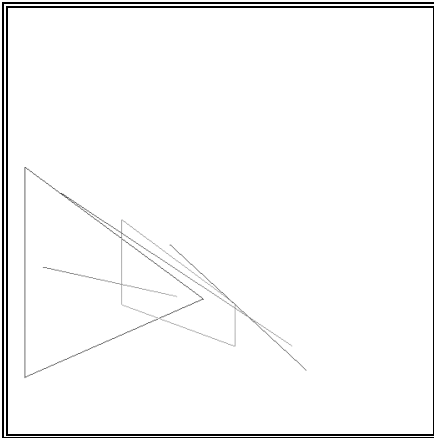


Fig. 37 - 4ª Banda de entidades emparelhadas.

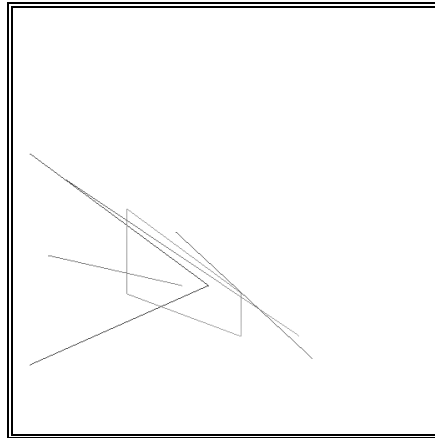


Fig. 38 - 5ª Banda de entidades emparelhadas.

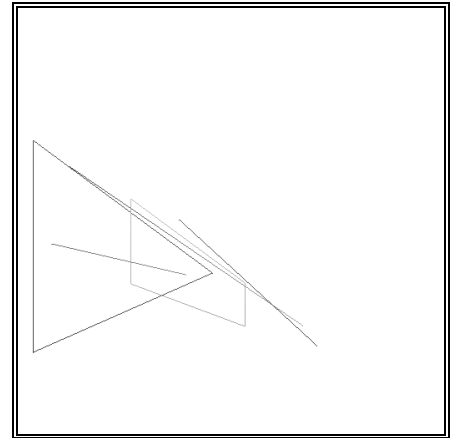


Fig. 39 - 6ª Banda de entidades emparelhadas.

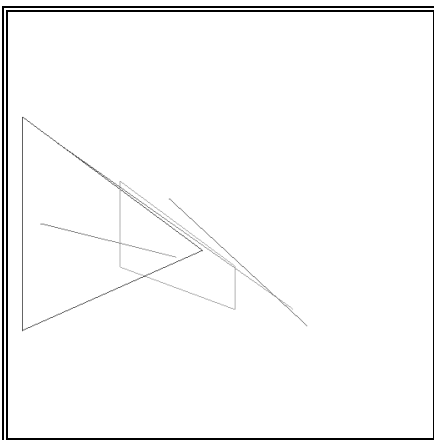


Fig. 40 - 7ª Banda de entidades emparelhadas.

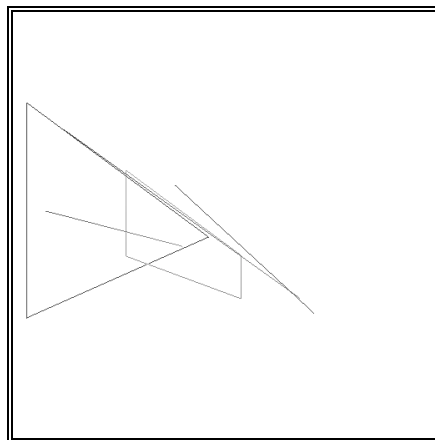


Fig. 41 - 8ª Banda de entidades emparelhadas.

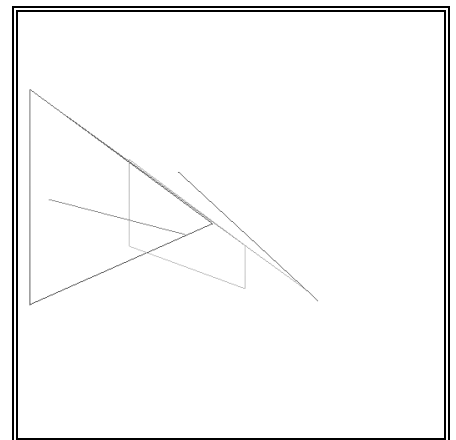


Fig. 42 - 9ª Banda de entidades emparelhadas.

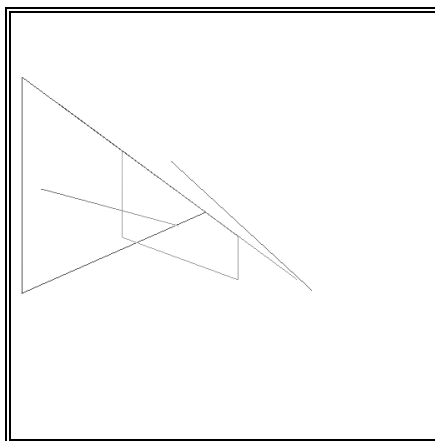


Fig. 43 - 10ª Banda de entidades emparelhadas.

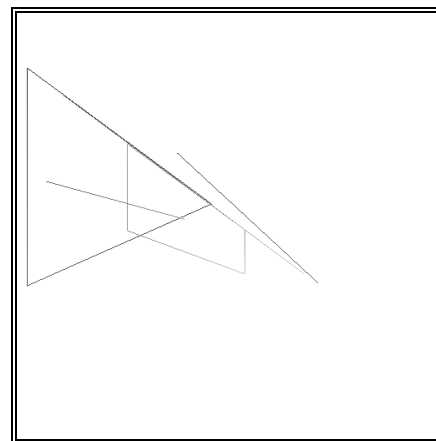


Fig. 44 - 11ª Banda de entidades emparelhadas.

Na Tab. VI estão apresentados os resultados obtidos no seguimento pela implementação *deep* e também o modo como decorreu todo o processo.

1ª Imagem					2ª Imagem						
Ref.	P. Inic.		P. Fin.		Emp.	Ref.	P. Inic.		P. Fin.		Emp.
1	26	256	235	411	I	1	24	240	235	396	KM
2	235	411	26	502	I	2	235	396	25	488	KM
3	26	502	26	256	I	3	25	488	24	240	KM
4	139	305	273	404	I	4	138	293	273	393	KM
5	273	404	273	454	I	5	273	393	273	442	KM
6	273	454	139	404	I	6	273	442	138	393	KM
7	139	404	139	305	I	7	138	393	138	293	KM
8	67	287	340	454	I	8	66	272	340	442	KM
9	47	368	204	398	I	9	46	355	203	385	KM
10	Não visível					10	Não visível				
3ª Imagem					4ª Imagem						
Ref.	P. Inic.		P. Fin.		Emp.	Ref.	P. Inic.		P. Fin.		Emp.
1	23	225	235	381	KM	1	Não visível				
2	235	381	24	474	KM	2	235	366	23	459	KM
3	24	474	23	225	KM	3	23	459	22	209	KM
4	138	281	273	381	KM	4	137	269	273	369	KM
5	273	381	273	430	KM	5	273	369	273	419	KM
6	273	430	138	381	KM	6	273	419	137	369	KM
7	138	381	138	281	KM	7	137	369	137	269	KM
8	65	256	340	430	KM	8	64	240	341	419	KM
9	45	341	203	373	KM	9	44	327	203	360	KM
10	Não visível					10	195	301	357	450	I
5ª Imagem					6ª Imagem						
Ref.	P. Inic.		P. Fin.		Emp.	Ref.	P. Inic.		P. Fin.		Emp.
1	21	193	235	351	KM	1	20	177	234	335	KM
2	235	351	21	445	KM	2	234	335	20	430	KM
3	21	445	21	193	KM	3	Não visível				
4	137	256	273	357	KM	4	136	243	273	344	KM
5	273	357	273	407	KM	5	273	344	273	395	KM
6	273	407	137	357	KM	6	273	395	136	344	KM
7	137	357	137	256	KM	7	136	344	136	243	KM
8	64	224	341	407	KM	8	63	208	342	395	KM
9	43	313	203	348	KM	9	42	299	202	335	KM
10	195	286	358	436	KM	10	195	271	358	422	KM
7ª Imagem					8ª Imagem						
Ref.	P. Inic.		P. Fin.		Emp.	Ref.	P. Inic.		P. Fin.		Emp.
1	19	161	234	320	KM	1	18	128	234	288	KRG
2	234	320	19	415	KM	2	234	288	18	384	KRG
3	19	415	19	161	KM	3	18	384	18	128	KM
4	136	231	273	332	KM	4	135	205	273	307	KM
5	273	332	273	383	KM	5	273	307	273	358	KM
6	273	383	136	332	KM	6	273	358	135	307	KM
7	136	332	136	231	KM	7	135	307	135	205	KM
8	62	192	342	383	KM	8	61	160	342	358	KM
9	41	285	202	322	KM	9	40	256	202	296	KM
10	194	256	359	407	KM	10	194	226	359	378	KM

Tab. VI - Resultados obtidos no seguimento pela implementação *deep* para a sequência de teste utilizada.
(continua)

9 ^a Imagem					10 ^a Imagem						
Ref.	P. Inic.		P. Fin.		Emp.	Ref.	P. Inic.		P. Fin.		Emp.
1	16	111	234	272	KRG	1	15	95	234	256	KM
2	234	272	16	369	KRG	2	234	256	15	353	KM
3	16	369	16	111	KRG	3	15	353	15	95	KM
4	135	192	273	294	KM	4	134	179	273	282	KM
5	273	294	273	345	KRG	5	273	282	273	333	KM
6	273	345	135	294	KRG	6	273	333	134	282	KM
7	135	294	135	192	KRG	7	134	282	134	179	KM
8	60	143	343	345	KM	8	59	127	343	333	KM
9	39	241	202	283	KRG	9	38	227	201	269	KM
10	194	210	360	363	KM	10	193	194	360	348	KM
11 ^a Imagem					12 ^a Imagem						
Ref.	P. Inic.		P. Fin.		Emp.	Ref.	P. Inic.		P. Fin.		Emp.
1	14	78	234	240	KM	1	13	67	234	230	KM
2	234	240	14	337	KM	2	234	230	13	328	KM
3	14	337	14	78	KM	3	13	328	13	67	KM
4	134	166	273	269	KM	4	133	158	274	261	KM
5	273	269	273	320	KM	5	274	261	274	313	KM
6	273	320	134	269	KM	6	274	313	133	261	KM
7	134	269	134	166	KM	7	133	261	133	158	KM
8	58	110	343	320	KM	8	57	100	344	313	KM
9	37	212	201	256	KM	9	36	203	201	248	KM
10	193	179	361	333	KM	10	193	169	361	324	KM

Legenda:
Ref. - Referência da entidade, **I** - Inicialização da entidade, **P. Inic.** - Coordenadas na memória *frame* do ponto inicial no respectivo ficheiro, **P. Fin.** - Coordenadas na memória *frame* do ponto final no respectivo ficheiro, **Emp.** - Tipo de emparelhamento: **KM** - Filtros de Kalman e distâncias de Mahalanobis e **KRG** - Filtros de Kalman e restrições geométricas.

Tab. VI - Continuação.

Nas *Figs. 45 e 46* estão representadas as duas bandas da imagem obtida pela implementação *deep* na mesma sequência de teste mas, desta feita, com o parâmetro *nFrameDeep* igual a 5. Assim, nestas bandas, apenas estão representadas as entidades que tenham sido emparelhadas com êxito após cada intervalo de cinco posições e orientações da câmara. Na *Fig. 47* é possível observar-se o conteúdo do ficheiro de saída obtido neste último caso.

Como é possível deprender pelos resultados experimentais obtidos, esta implementação oferece resultados, em termos médios, de boa qualidade. Dos mesmos resultados, é também de destacar que a utilização de restrições geométricas como medida de emparelhamento, quando este não é previamente obtido com êxito utilizando as distâncias de Mahalanobis, tem como grande vantagem o contínuo êxito do emparelhamento e, conseqüentemente, o respectivo seguimento. Dos mesmos resultados, também é possível deprender que a filosofia utilizada na medição e posterior actualização é correcta, pois, mesmo nos casos em que uma dada entidade não é visível numa imagem e é visível na seguinte, o seguimento é obtido com êxito. Do mesmo modo, também se verifica que, quando os filtros utilizados começam a divergir ou a convergir em demasia (ficando a respectiva elipse de emparelhamento demasiado pequena), a utilização de restrições geométricas como medida de emparelhamento origina que os filtros se autoajustem de forma mais rápida e produzam emparelhamentos com êxito ao longo do processo de seguimento.

Da observação das *Fig. 18 e 47*, é possível concluir que o método utilizado para a obtenção das coordenadas 3D origina, em média, bons resultados mesmo para posições e orientações da câmara muito semelhantes e considerando apenas translações da mesma, como o caso correspondente à *Fig. 47*. No decorrer das experiências realizadas, verificou-se que para *nFrameDeep* igual a 11 os resultados obtidos para as coordenadas 3D eram, em média, ainda de melhor qualidade, *Fig. 48*.

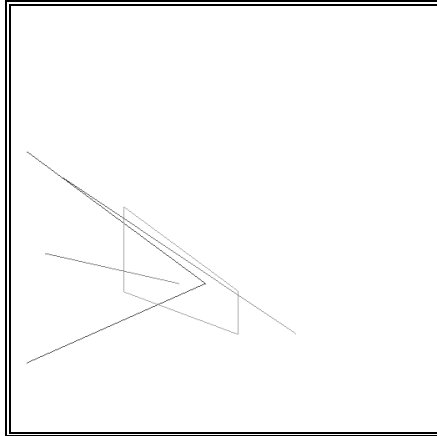


Fig. 45 - 1ª Banda de entidades emparelhadas com o parâmetro $nFrameDeep$ igual a 5.

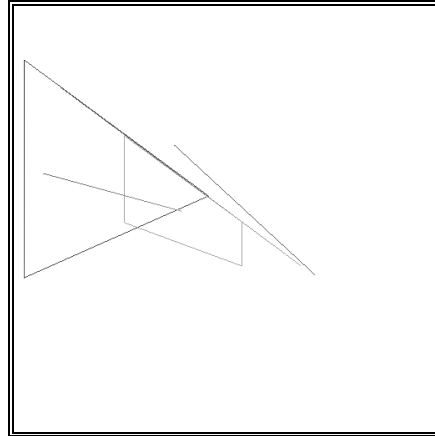


Fig. 46 - 2ª Banda de entidades emparelhadas com o parâmetro $nFrameDeep$ igual a 5.

Results for deep in teste.txt with:	...
Maximum deep: 2800	(42, 299) (-0.052527, 39.9993, 2205.98)
Minimum deep: 1800	(202, 335) (40.0727, 54.8393, 2387.62)
Number of between frames for deep's calculation: 5	Image coordinates (X,Y), World coordinates (X,Y,Z) of the lines's end points in file:
Horizontal factor, sx: 0.710935	out12.txt
Camera's sensors distance for x, dx: 0.00837766	(13, 67) (0.248233, 0.184139, 1990.55)
Camera's sensors distance for y, dy: 0.0080756	(234, 230) (49.9374, 50.0046, 2008.86)
Focal distance, f: 50	(234, 230) (49.9374, 50.0046, 2008.86)
Radial lens distortion, k: 0.001	(13, 328) (-0.140351, 80.1442, 2004.53)
Maximum distance of the middle's point for matching: 100	(13, 328) (-0.219774, 80.1633, 2006)
Maximum difference of length for matching: 20	(13, 67) (0.107051, 0.0050798, 1996.33)
Maximum angular difference for matching: 10.	(133, 158) (19.7125, 19.8252, 2513.94)
Image coordinates (X,Y), World coordinates (X,Y,Z) of the lines's end points in file: out6.txt	(274, 261) (60.064, 59.939, 2506.91)
(20, 177) (-0.0467793, 0.00156601, 2004.43)	(274, 261) (60.064, 59.939, 2506.91)
(234, 335) (49.9919, 49.6241, 1979.09)	(274, 313) (60.0488, 80.0387, 2498.45)
(234, 335) (49.9919, 49.6241, 1979.09)	(274, 313) (60.0488, 80.0387, 2498.45)
(20, 430) (-0.300212, 80.2478, 2008.09)	(133, 261) (19.8344, 59.939, 2505.69)
(136, 243) (19.9789, 19.8398, 2504.89)	(133, 261) (19.8344, 59.939, 2505.69)
(273, 344) (60.0157, 59.8319, 2499.43)	(133, 158) (19.7125, 19.8252, 2513.94)
(273, 344) (60.0157, 59.8319, 2499.43)	(57, 100) (9.69881, 9.75587, 2013.91)
(273, 395) (60.0045, 79.883, 2491.56)	(344, 313) (80.1063, 80.0411, 2497.98)
(273, 395) (60.0045, 79.883, 2491.56)	(36, 203) (-0.0705478, 39.9915, 2204.48)
(136, 344) (20.0486, 59.823, 2497.77)	(201, 248) (39.9031, 55.0314, 2402.49)
(136, 344) (20.0486, 59.823, 2497.77)	(193, 169) (40.0211, 30.0148, 2095.37)
(136, 243) (19.9789, 19.8398, 2504.89)	(361, 324) (79.902, 79.8633, 2094.29)
(63, 208) (10.2925, 9.96081, 1987.52)	...
(342, 395) (80.0185, 79.8811, 2490.88)	

Fig. 47 - Conteúdo do ficheiro de saída obtido pela implementação **deep** para a seqüência utilizada e com o parâmetro $nFrameDeep$ igual a 5.

Results for deep in teste.txt with:	...
Maximum deep: 2800	(13, 328) (-0.219774, 80.1633, 2006)
Minimum deep: 1800	(13, 328) (-0.219774, 80.1633, 2006)
Number of between frames for deep's calculation: 11	(13, 67) (0.107051, 0.0050798, 1996.33)
Horizontal factor, sx: 0.710935	(133, 158) (19.8566, 19.8825, 2510.15)
Camera's sensors distance for x, dx: 0.00837766	(274, 261) (60.1106, 59.9371, 2503.82)
Camera's sensors distance for y, dy: 0.0080756	(274, 261) (60.1106, 59.9371, 2503.82)
Focal distance, f: 50	(274, 313) (60.0968, 80.0132, 2495.61)
Radial lens distortion, k: 0.001	(274, 313) (60.0968, 80.0132, 2495.61)
Maximum distance of the middle's point for matching: 100	(133, 261) (19.9564, 59.934, 2502.39)
Maximum difference of length for matching: 20	(133, 261) (19.9564, 59.934, 2502.39)
Maximum angular difference for matching: 10.	(133, 261) (19.9564, 59.934, 2502.39)
Image coordinates (X,Y), World coordinates (X,Y,Z) of the lines's end points in file: out12.txt	(133, 158) (19.8566, 19.8825, 2510.15)
(13, 67) (0.107051, 0.0050798, 1996.33)	(57, 100) (9.86397, 10.0406, 2002.81)
(234, 230) (50.0206, 50.0571, 1996.38)	(344, 313) (80.0504, 80.0133, 2495.02)
(234, 230) (50.0206, 50.0571, 1996.38)	(36, 203) (-0.084182, 39.986, 2205.1)
...	(201, 248) (40.061, 55.038, 2396.34)

Fig. 48 - Conteúdo do ficheiro de saída obtido pela implementação *deep* para a sequência utilizada e com o parâmetro *nFrameDeep* igual a 11.

4.4 - Utilizando a implementação *compdeep* para a extracção de coordenadas 3D de pontos a partir das suas coordenadas 2D em duas imagens distintas

Neste ponto é apresentado um exemplo de resultados experimentais obtidos pela implementação *compdeep*, apresentada de forma sumária no ponto 3.4. Os parâmetros de entrada para a referida implementação, utilizados neste exemplo, estão representados na Tab. VII, e o diálogo com o utilizador, assim como os resultados obtidos, constam da Fig. 49.

<i>sx</i>	<i>dx</i>	<i>dy</i>	<i>xCenterBuf</i>	<i>yCenterBuf</i>	<i>focalDist</i>	<i>radialLensDist</i>
0.710935	0.00837766	0.0080756	256	256	50.0	0.001

Tab. VII - Parâmetros de entrada para a implementação *compdeep* utilizados no exemplo apresentado.

Como o ponto em questão tinha coordenadas 3D (40, 30, 2100), esta implementação obteve, para este exemplo, bons resultados.

De referir que, no decorrer da análise dos resultados obtidos por esta implementação, puderam-se efectuar as seguintes observações¹⁶:

- tornou-se óbvia a importância da qualidade da calibração da câmara utilizada na obtenção de bons resultados;
- a qualidade dos valores introduzidos para a posição e orientação da câmara relativamente ao referencial global para cada imagem, tem grande influência nos resultados obtidos, em particular nos valores da orientação;
- as localizações do ponto a considerar nas duas imagens, para produzirem bons resultados, devem ser bem distintas;
- as coordenadas imagem dos pontos considerados influenciam a qualidade dos resultados obtidos, embora em menor escala do que os factores anteriores.

¹⁶ Semelhantes observações poderiam ser efectuadas a partir dos resultados da implementação *deep* no que diz respeito à obtenção de coordenadas 3D dos pontos que constituem as diversas entidades consideradas.


```

For camera one:
  Enter the rotation about the x axis (in degree):0
  Enter the rotation about the y axis (in degree):0
  Enter the rotation about the z axis (in degree):0
  Enter the translation along the x axis:-55
  Enter the translation along the y axis:-15
  Enter the translation along the z axis:-30
For camera 2:
  Enter the rotation about the x axis (in degree):0
  Enter the rotation about the y axis (in degree):0
  Enter the rotation about the z axis (in degree):0
  Enter the translation along the y axis:-55
  Enter the translation along the z axis:-100
Perspective transformation's matrix computed for camera one:
mpp1[0][0]=1 mpp1[0][1]=-0 mpp1[0][2]=0 mpp1[0][3]=-55
mpp1[1][0]=0 mpp1[1][1]=1 mpp1[1][2]=0 mpp1[1][3]=-15
mpp1[2][0]=0 mpp1[2][1]=0 mpp1[2][2]=0.02 mpp1[2][3]=-0.6
Perspective transformation's matrix computed for camera two:
mpp2[0][0]=1 mpp2[0][1]=-0 mpp2[0][2]=0 mpp2[0][3]=-55
mpp2[1][0]=0 mpp2[1][1]=1 mpp2[1][2]=0 mpp2[1][3]=-55
mpp2[2][0]=0 mpp2[2][1]=0 mpp2[2][2]=0.02 mpp2[2][3]=-2
For camera one:
  Enter the frame buffer's coordinate in direction x of the desired point (9999 to quit):195
  Enter the frame buffer's coordinate in direction y of the desired point:301
For camera two:
  Enter the frame buffer's coordinate in direction x of the desired point:193
  Enter the frame buffer's coordinate in direction y of the desired point:179
Undistorted's coordinates computed of the desired point in camera one's image plane: -0.36341 0.363498
Undistorted's coordinates computed of the desired point in camera two's image plane: -0.375424 -0.622149
World's coordinates computed of the desired point: X=39.9444, Y=30.0728, Z=2103.32

```

Fig. 49 - Diálogo utilizador/implementação **compdeep** e os respectivos resultados obtidos no exemplo apresentado.

5 - Algumas conclusões finais

Neste capítulo é apresentado um método para o seguimento de segmentos de recta em imagens sucessivas. Essas entidades são parametrizadas pela posição do ponto médio, pelo comprimento e pela direcção. Para o seguimento, são utilizados três filtros de Kalman independentes; nomeadamente, um para a posição do ponto médio, outro para a direcção e um último para o comprimento. Como medida do emparelhamento entre as características verificadas por uma dada candidata e as previstas pelos respectivos filtros, são utilizadas distâncias normalizadas de Mahalanobis ou, quando desta forma o emparelhamento não é obtido com êxito, restrições geométricas. O modelo cinemático associado a cada filtro é de aceleração constante localmente.

Um método para a obtenção de informação tridimensional de segmentos de recta, a partir das suas imagens obtidas por uma mesma câmara para duas posições e orientações distintas, é também apresentado neste capítulo. Com este método torna-se possível obter as coordenadas 3D dos pontos inicial e final de cada segmento de recta devidamente emparelhado em duas imagens, obtidas por uma câmara em posições e orientações distintas. Este método é baseado no princípio da triangulação estereoscópica e utiliza linhas epipolares para determinar a verdadeira localização dos pontos inicial e final de um segmento de recta, visíveis numa dada imagem, sobre o segmento visível na imagem anterior a esta. Quando tal determinação não é possível por este processo, esta é realizada comparando as localizações destes pontos nas duas imagens.

São ainda apresentadas, de forma sumária, quatro implementações desenvolvidas no âmbito deste capítulo. Assim, são apresentadas implementações para: determinar os quatro vértices da área visível por uma dada câmara sobre um dado plano; simular uma câmara móvel, para segmentos de recta

definidos no espaço tridimensional; obtenção de informação tridimensional a partir do movimento de uma dada câmara; obter as coordenadas 3D de um ponto, a partir de duas imagens distintas.

Alguns resultados experimentais obtidos pelas implementações anteriormente referidas são também apresentados neste capítulo, tendo, em termos médios, resultados de boa qualidade.

Como é possível depreender pelos resultados experimentais apresentados, a abordagem global induzida neste capítulo obtém resultados de boa qualidade. Dos mesmos, também é possível depreender que a filosofia utilizada na medição e posterior actualização é correcta, pois mesmo nos casos em que uma dada entidade não é visível numa imagem e é visível na seguinte, o seu seguimento é obtido com êxito. Também se verifica que, quando os filtros utilizados começam a divergir ou a convergir em demasia, a utilização de restrições geométricas origina que os filtros se autoajustem de forma mais rápida, e mantenham emparelhamentos com êxito ao longo do processo de seguimento. A solução encontrada para resolver o problema dos filtros poderem convergir em demasia, foi a introdução de ruído na modelação do sistema; assim, as respectivas elipses de emparelhamento não podem assumir dimensões demasiado reduzidas e o emparelhamento é facilitado continuamente. Também se verificou que a utilização de restrições geométricas, como medida de emparelhamento, tem como consequência tornar mais fácil e flexível a fase de inicialização das entidades consideradas.

A utilização de um factor de confiança para as entidades presentes no modelo, tornou-se de grande utilidade, nomeadamente para manter o modelo de entidades sempre actualizado, evitando assim que este cresça para valores muito elevados e desnecessários.

Capítulo V

ALGUNS RESULTADOS EXPERIMENTAIS OBTIDOS EM SITUAÇÕES REAIS

Após a apresentação da abordagem global utilizada nesta dissertação para a obtenção de estrutura tridimensional a partir do movimento de uma câmara, neste capítulo são apresentados dois exemplos de tal obtenção. No primeiro exemplo considerado, a câmara tem um movimento de translação pura e, no segundo, tem um movimento de rotação pura.

São assim, apresentados resultados experimentais obtidos para a calibração da câmara, para a determinação das entidades consideradas em cada imagem das sequências, e para a obtenção de informação tridimensional a partir do movimento da câmara.

Introdução

Neste capítulo são apresentados alguns resultados obtidos pela abordagem global utilizada nesta dissertação para a obtenção de informação tridimensional em duas situações reais. Na primeira situação pretende-se obter informação tridimensional de um cubo a partir de uma sequência de 16 imagens obtidas por uma câmara, devidamente calibrada, que apresenta um movimento de translação pura. Na segunda situação, pretende-se novamente obter informação tridimensional do mesmo cubo a partir de uma sequência de 8 imagens obtidas por uma câmara, também devidamente calibrada, que apresenta um movimento de rotação pura.

A câmara utilizada foi uma *SONY CCD-IRIS SSC-M370CE* com uma área sensora de dimensões 6.3×4.7 mm, número de elementos de imagem 752×582 , distância entre centros dos elementos sensores vizinhos segundo a direcção x igual a 8.377659574 μm , distância entre centros dos sensores CCD vizinhos segundo a direcção y igual a 8.075601375 μm , com uma lente *FUJINON TV LENS HF16A-2 1:1.4/16*; a placa de aquisição de imagem foi uma *MATROX IMAGE-LC* com *frame buffer* de 765×574 *pixel*.

São apresentados neste capítulo os resultados experimentais obtidos pelas seguintes implementações:

- na determinação das coordenadas na memória *frame* dos pontos de calibração utilizados: *calpoint*;
- na calibração da câmara utilizada: *calcamera*;
- no seguimento de linhas e aproximação poligonal utilizando faixas dinâmicas: *dstrip*;
- na fusão de segmentos de recta próximos e de direcções similares: *linelink*;
- na obtenção de informação tridimensional a partir do movimento da câmara: *deep*.

As referidas implementação estão apresentadas, de forma sumária, nos capítulos anteriores desta publicação e, de forma mais detalhada, em [Tavares, 1995]; e foram executadas numa estação de trabalho *SunSparc* em ambiente *UNIX*.

Esta apresentação encontra-se dividida em três secções; na primeira, são apresentados os resultados experimentais obtidos na calibração da câmara utilizada; na segunda, os obtidos na determinação de informação tridimensional e na terceira, e última, algumas conclusões finais.

Deve referir-se que o objectivo deste capítulo é apenas o de apresentar dois exemplos de utilização da abordagem global utilizada nesta dissertação; assim, não foi realizado nenhum tipo de optimização na procura dos melhores resultados experimentais possíveis para cada situação considerada.

1 - Resultados experimentais obtidos para a calibração da câmara

As posições relativas da câmara e do plano de calibração, e a definição dos referencias mundo (x_w, y_w, z_w) e câmara (x_c, y_c, z_c) , para a primeira calibração realizada, estão representados na *Fig. 1*.

Na *Fig. 2* está representada a imagem¹ obtida pela câmara para as posições relativas consideradas nesta primeira calibração. A referida imagem foi posteriormente sujeita ao detector de orlas de intensidade de Deriche para a classificação dos seus *pixels* em termos de amplitude e de direcção. Após esta classificação, foram determinadas as coordenadas na memória *frame* dos pontos de calibração considerados (os vértices dos quadrados representados) utilizando-se a implementação *calpoint* com os parâmetros de entrada *erthresang=5* e *thresamp=25*, e os restantes assumindo os respectivos valores por defeito. Na *Fig. 3*, está representada a terceira banda da imagem resultante da implementação *calpoint* para esta primeira calibração; nesta banda as localizações dos pontos de calibração considerados estão referenciadas por meio de cruces.

¹ Todas as imagens apresentadas neste capítulo, excepto zonas ampliadas, sofreram um escalonamento nas suas dimensões de 65%.

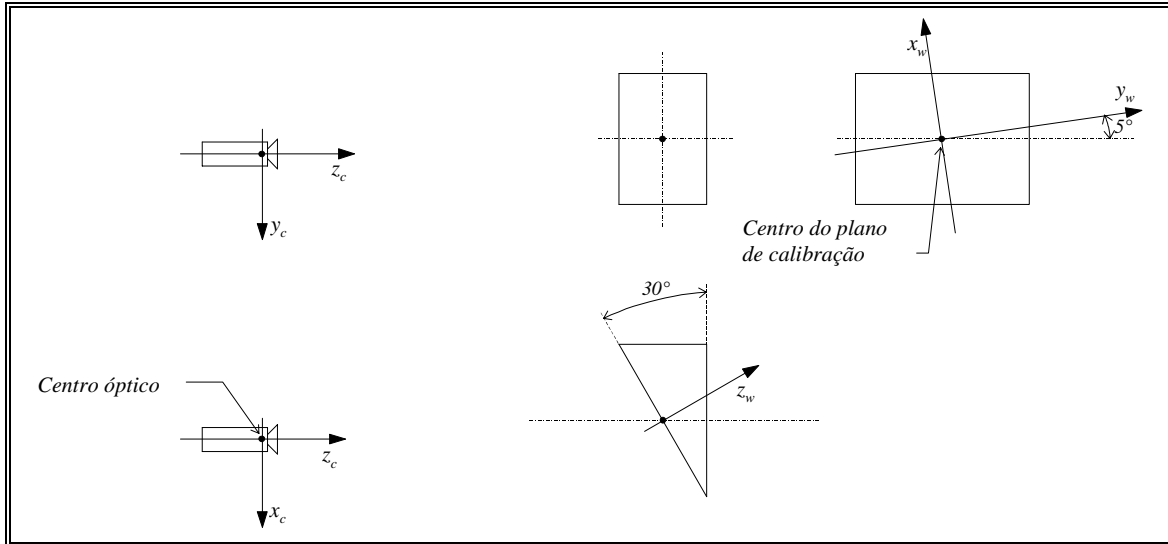


Fig. 1 - Posições relativas e referencias, utilizados na primeira calibração efectuada.

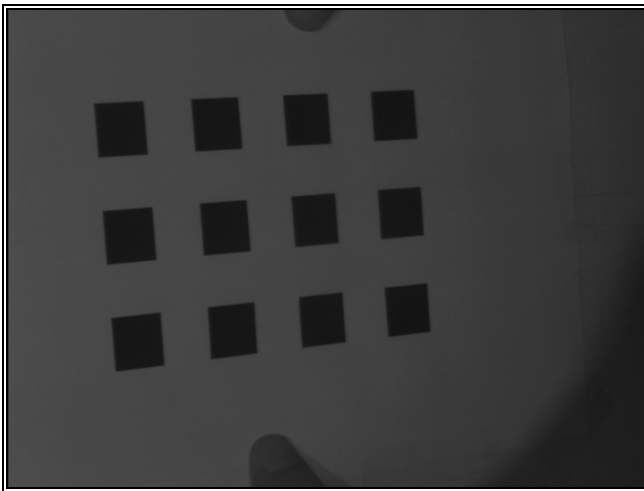


Fig. 2 - Imagem obtida pela câmara para as posições relativas utilizadas na primeira calibração efectuada.

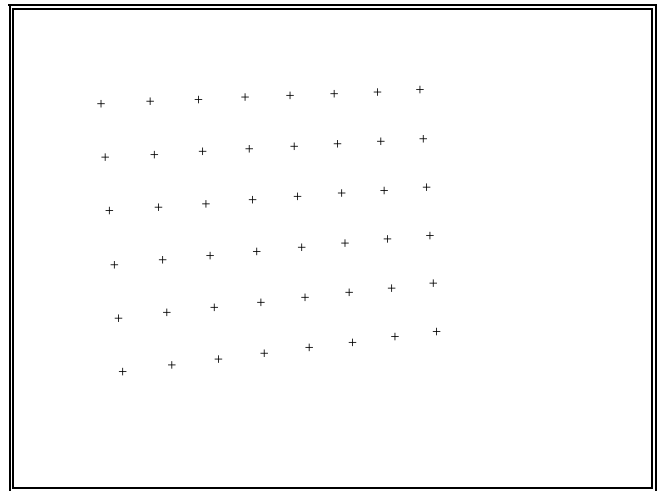


Fig. 3 - 3ª Banda da imagem resultante da implementação calpoint para a primeira calibração efectuada.

Para verificar a qualidade obtida pela implementação **calpoint**, na determinação das localizações dos pontos de calibração considerados nesta primeira calibração, está representada na Fig. 4 a soma da imagem representada na Fig. 2 com a banda representada na Fig. 3. Com o mesmo objectivo, está representada na Fig. 5 uma zona ampliada da imagem representada na Fig. 4 e escolhida de forma aleatória.

As coordenadas 3D mundo e as 2D na memória *frame* dos pontos de calibração considerados nesta primeira calibração, foram devidamente agrupadas num ficheiro de texto. Desta forma, por intermédio da implementação **calcamera** com os parâmetros de entrada apresentados na Tab. I, é neste momento possível proceder à calibração da câmara utilizada.

Na Fig. 6, estão representados os resultados obtidos pela implementação **calcamera** para esta primeira calibração.

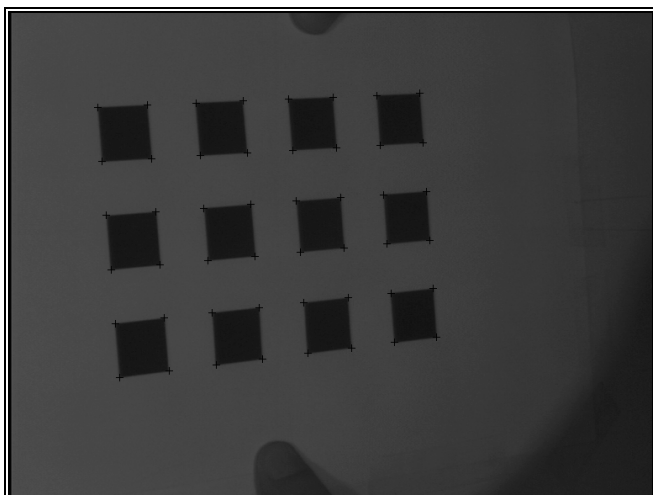


Fig. 4 - Soma da imagem representada na Fig. 2 com a banda representada na Fig. 3.

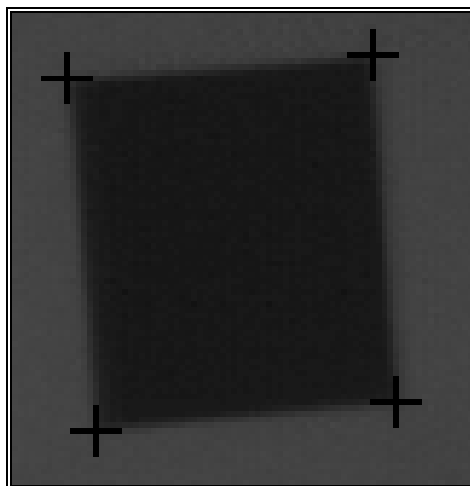


Fig. 5 - Zona aleatória da imagem representada na Fig. 4 ampliada 900%.

<i>outfile</i>	<i>nlstop</i>	<i>sx</i> *	<i>dx</i>	<i>dy</i>
output.txt	1e-07	0.983007	0.00837766	0.0080756
<i>Cx</i>	<i>Cy</i>	<i>findcenter</i>	<i>infile</i>	
382	287	0	incalib1.txt	

*O factor de incerteza horizontal não foi previamente calibrado, pois não houve condições práticas para a utilização da implementação *findsx*. Assim, utilizou-se para este factor a aproximação dada pelo cociente entre o número de elementos sensores na direcção *x* e o número de elementos imagem numa linha discretizada pela placa de aquisição.

Tab. I - Parâmetros de entrada para a implementação *calcamera* utilizados na primeira calibração efectuada.

Results for <i>calcamera</i> with: Stop for nonlinear equations=1e-07, Find center=0, <i>sx</i> =0.983007, <i>dx</i> =0.00837766, <i>dy</i> =0.0080756, <i>Cx</i> =382, <i>Cy</i> =287, Input file: incalib1.txt with 48 calibration points. Rotation matrix R: <table style="margin-left: 40px;"> <tr> <td>-0.0763951</td> <td>+0.887468</td> <td>-0.454494</td> </tr> <tr> <td>-0.996712</td> <td>-0.0803089</td> <td>+0.0107205</td> </tr> <tr> <td>-0.0269858</td> <td>+0.453819</td> <td>+0.890685</td> </tr> </table> Translation vector T: (-21.9711, -9.44748, +640.131) Focal length f: +15.7665 Radial lens distortion k1: -0.000766017	-0.0763951	+0.887468	-0.454494	-0.996712	-0.0803089	+0.0107205	-0.0269858	+0.453819	+0.890685
-0.0763951	+0.887468	-0.454494							
-0.996712	-0.0803089	+0.0107205							
-0.0269858	+0.453819	+0.890685							

Fig. 6 - Resultados obtidos pela implementação *calcamera* para a primeira calibração efectuada.

Algumas observações podem ser realizadas acerca dos resultados apresentados na Fig. 6:

- Em virtude de as coordenadas 3D atribuídas a cada ponto de calibração utilizado não serem suficientemente rigorosas e exactas, assim como o desenho dos quadrados utilizados não ser rigoroso nem os pontos utilizados serem verdadeiramente coplanares, pode-se considerar que os resultados obtidos apresentam uma qualidade aceitável. Tal qualidade pode ser comprovada pelos ângulos da rotação 3D do sistema mundo para o câmara: segundo o eixo *x*, obteve-se 0.707 quando o ângulo real era de cerca de 0.0 grau; segundo o eixo *y*, obteve-se 27.032 quando o real era próximo de 30.0 grau; e, segundo o eixo *z*, obteve-se 94.919 quando o real era de cerca de 95 grau. Os valores obtidos para a translação do sistema mundo para o da câmara,

também parecem bastante aceitáveis, assim como os obtidos para a distância focal efectiva e para a distorção radial da lente.

- Não foi possível utilizar a referida implementação para a determinação das coordenadas correctas do centro da imagem na memória *frame* de forma a corresponder a resultados aceitáveis. Pensa-se que um dos factores que teve influência nesta incapacidade, se prende com o baixo rigor das coordenadas 3D atribuídas aos pontos de calibração utilizados; por outro lado, a baixa distorção radial da lente também teve influência decisiva nesta incapacidade.

Após esta calibração, a câmara foi aproximada do plano de calibração (cerca de 150 mm segundo o eixo *z* do referencial câmara), mantendo-se todas as restantes variáveis do *setup* prático constantes. Para esta nova posição relativa da câmara e do plano de calibração, efectuou-se uma segunda calibração.

Na *Fig. 7* está representada a imagem obtida pela câmara para esta segunda calibração. A referida imagem foi posteriormente sujeita ao detector de orlas de intensidade de Deriche. Após a utilização deste detector, foram determinadas as coordenadas na memória *frame* dos pontos de calibração considerados, utilizando-se para tal a implementação *calpoint* com os parâmetros de entrada *erthresang*=5 e *thresamp*=25, e os restantes assumindo os respectivos valores por defeito. Na *Fig. 8*, está representada a terceira banda da imagem resultante da implementação *calpoint* para esta segunda calibração.

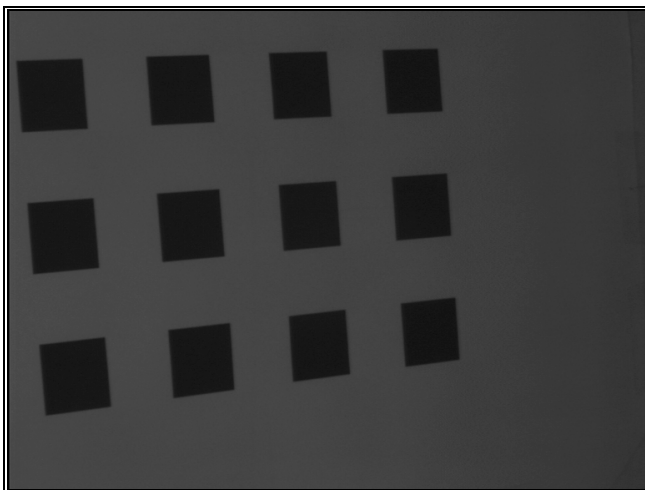


Fig. 7 - Imagem obtida pela câmara para as posições relativas utilizadas na segunda calibração efectuada.

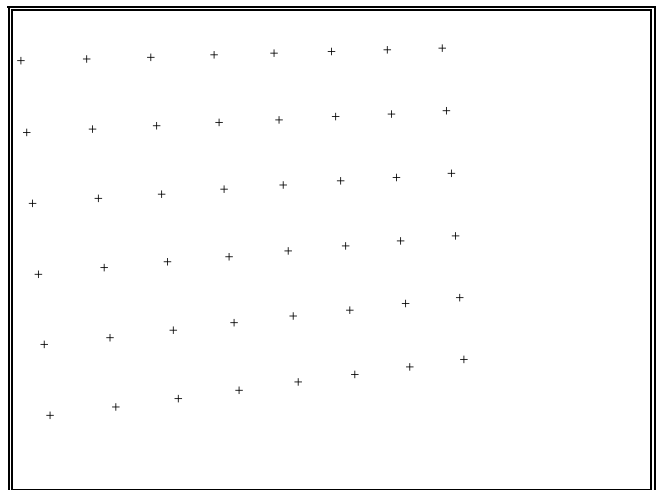


Fig. 8 - 3ª Banda da imagem resultante da implementação calpoint para a segunda calibração efectuada.

Para verificar a qualidade obtida pela implementação *calpoint*, na determinação das localizações dos pontos de calibração considerados nesta segunda calibração, está representada na *Fig. 9* a soma da imagem representada na *Fig. 7* com a banda representada na *Fig. 8*. Com o mesmo objectivo, está representada na *Fig. 10* uma zona ampliada da imagem representada na *Fig. 9*, escolhida de forma aleatória.

As coordenadas 3D mundo e as 2D na memória *frame* dos pontos de calibração foram devidamente agrupadas num ficheiro de texto. Procedeu-se de seguida à calibração da câmara utilizada, por intermédio da implementação *calcamera*, com os parâmetros de entrada apresentados na *Tab. II*.

<i>outfile</i>	<i>nstop</i>	<i>sx</i> *	<i>dx</i>	<i>dy</i>
output.txt	1e-07	0.983007	0.00837766	0.0080756
<i>Cx</i>	<i>Cy</i>	<i>findcenter</i>	<i>infile</i>	
382	287	0	incalib2.txt	

*Ver comentário da *Tab. I*.

Tab. II - Parâmetros de entrada para a implementação calcamera utilizados na segunda calibração efectuada.

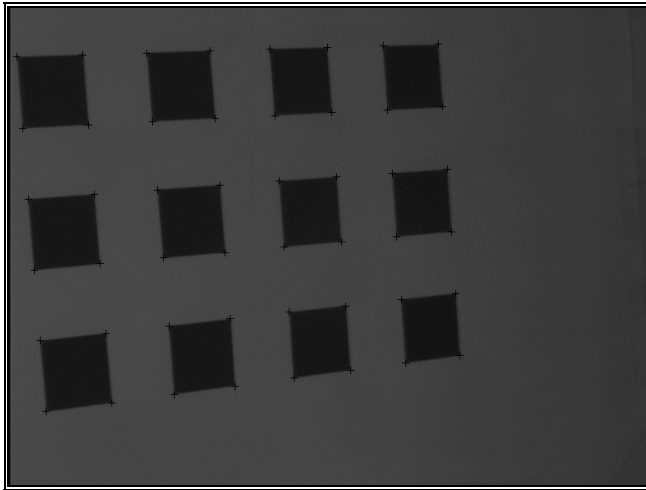


Fig. 9 - Soma da imagem representada na Fig. 7 com a banda representada na Fig. 8.

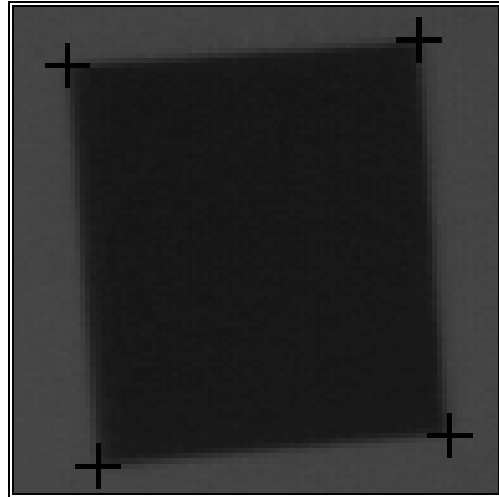


Fig. 10 - Zona aleatória da imagem representada na Fig. 9 ampliada 800%.

Na Fig. 11, estão representados os resultados obtidos pela implementação *calcamera* para esta segunda calibração.

Results for <i>calcamera</i> with:									
Stop for nonlinear equations=1e-07, Find center=0, sx=0.983007, dx=0.00837766, dy=0.0080756, Cx=382, Cy=287, Input file: incalib2.txt with 48 calibration points.									
Rotation matrix R:									
<table> <tr> <td>-0.0766848</td> <td>+0.891369</td> <td>-0.446744</td> </tr> <tr> <td>-0.996703</td> <td>-0.0804481</td> <td>+0.0105721</td> </tr> <tr> <td>-0.0265161</td> <td>+0.446082</td> <td>+0.894599</td> </tr> </table>	-0.0766848	+0.891369	-0.446744	-0.996703	-0.0804481	+0.0105721	-0.0265161	+0.446082	+0.894599
-0.0766848	+0.891369	-0.446744							
-0.996703	-0.0804481	+0.0105721							
-0.0265161	+0.446082	+0.894599							
Translation vector T: (-22.8665, -8.82566, +484.805)									
Focal length f: +15.5906									
Radial lens distortion k1: -0.000255061									

Fig. 11 - Resultados obtidos pela implementação *calcamera* para a segunda calibração efectuada.

A qualidade dos resultados apresentados na Fig. 11 é semelhante à dos apresentados na Fig. 6 (isto é, antes da respectiva translação). De notar que os valores das duas translações segundo o eixo z diferem entre si 155.3 mm, valor próximo da translação realizada entre as duas situações. Do mesmo modo, pode-se verificar que houve pequenas alterações nos valores dos restantes parâmetros. Assim, as observações realizadas para os resultados obtidos para a primeira calibração efectuada são do mesmo modo válidas para esta calibração.

Após esta segunda calibração, foi ainda efectuada uma terceira na qual a câmara utilizada estava colocada verticalmente sobre o plano de calibração. As posições relativas da câmara e do plano de calibração, e a definição dos referenciais mundo e câmara, para esta terceira calibração, estão representadas na Fig. 12.

Na Fig. 13 está representada a imagem obtida pela câmara para as posições relativas consideradas nesta terceira calibração. A referida imagem foi posteriormente sujeita ao detector de orlas de intensidade de Deriche. Após a utilização deste detector, foram determinadas as coordenadas na memória *frame* dos pontos de calibração considerados, utilizando-se para tal a implementação *calpoint* com o parâmetro de entrada *erthresang*=6, e os restantes assumindo os respectivos valores por defeito. Na Fig. 14, está representada a terceira banda da imagem resultante da implementação *calpoint* para esta terceira calibração.

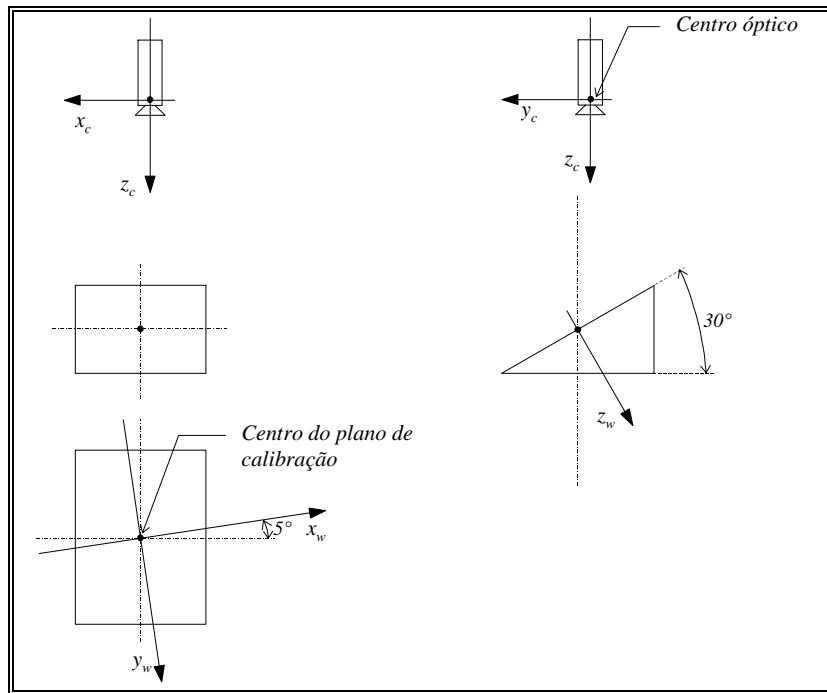


Fig. 12 - Posições relativas e referências utilizados na terceira calibração efectuada.

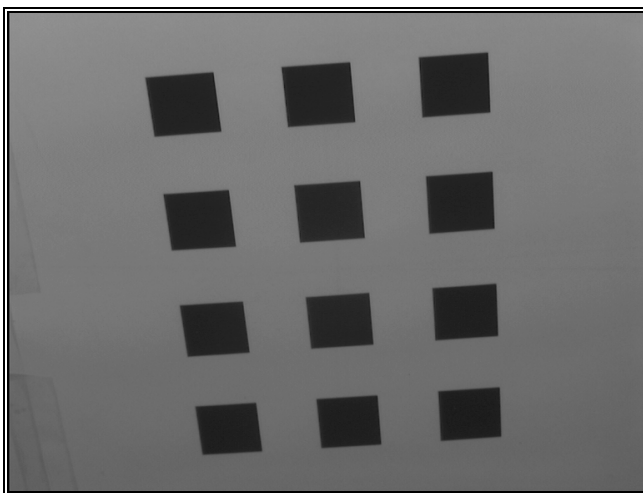


Fig. 13 - Imagem obtida pela câmara para as posições relativas utilizadas na terceira calibração efectuada.

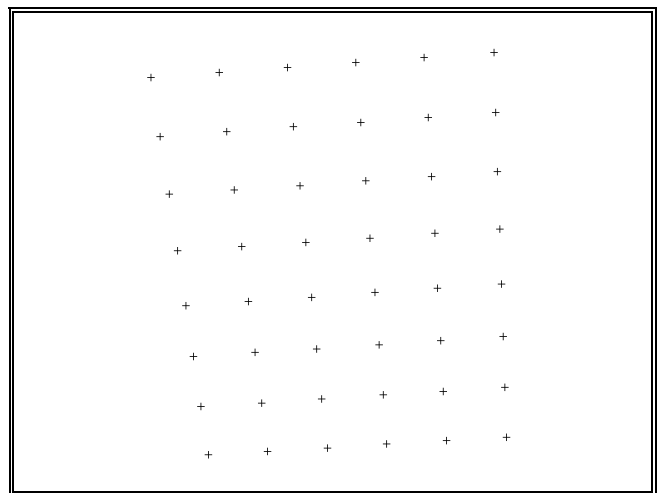


Fig. 14 - 3ª Banda da imagem resultante da implementação **calpoint** para terceira calibração efectuada.

Para verificar a qualidade obtida pela implementação **calpoint**, na determinação das localizações dos pontos de calibração, está representada na Fig. 15 a soma da imagem representada na Fig. 13 com a banda representada na Fig. 14. Com o mesmo objectivo, está representada na Fig. 16 uma zona ampliada da imagem representada na Fig. 15, escolhida de forma aleatória.

As coordenadas 3D mundo e as 2D na memória *frame* dos pontos de calibração foram devidamente agrupadas num ficheiro de texto. Procedeu-se de seguida à calibração da câmara, por intermédio da implementação **calcamera** com os parâmetros de entrada apresentados na Tab. III.

Na Fig. 17 estão representados os resultados obtidos pela implementação **calcamera** para esta terceira calibração.

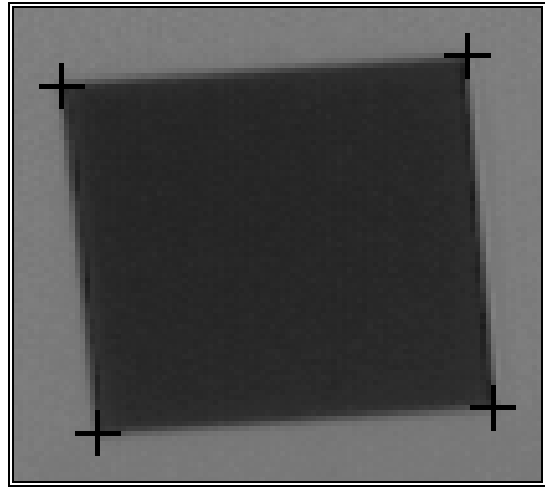
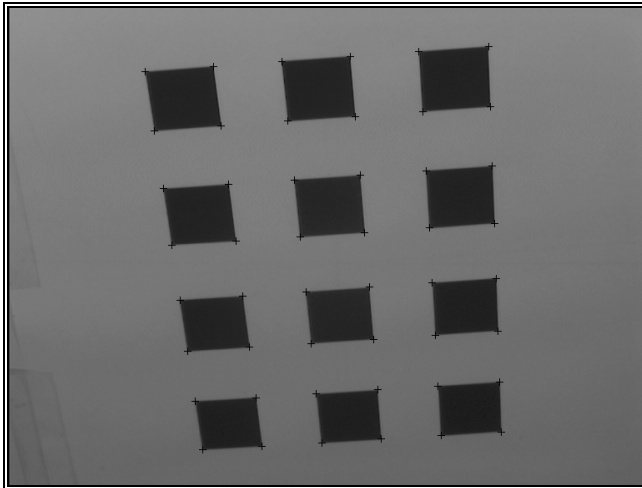


Fig. 15 - Soma da imagem representada na Fig. 13 com a banda representada na Fig. 14.

Fig. 16 - Zona aleatória da imagem representada na Fig. 15 ampliada 800%.

<i>outfile</i>	<i>nstop</i>	<i>sx</i> *	<i>dx</i>	<i>dy</i>
output.txt	1e-07	0.983007	0.00837766	0.0080756
<i>Cx</i>	<i>Cy</i>	<i>findcenter</i>	<i>infile</i>	
382	287	0	incalib3.txt	

*Ver comentário da Tab. I.

Tab. III - Parâmetros de entrada para a implementação *calcamera* utilizados na terceira calibração efectuada.

Results for <i>calcamera</i> with: Stop for nonlinear equations=1e-07, Find center=0, <i>sx</i> =0.983007, <i>dx</i> =0.00837766, <i>dy</i> =0.0080756, <i>Cx</i> =382, <i>Cy</i> =287, Input file: incalib3.txt with 48 calibration points. Rotation matrix R: <table style="margin-left: 40px;"> <tr> <td>+0.996727</td> <td>+0.0808120</td> <td>+0.00238398</td> </tr> <tr> <td>-0.0665790</td> <td>+0.837192</td> <td>-0.542841</td> </tr> <tr> <td>-0.0458639</td> <td>+0.540906</td> <td>+0.839832</td> </tr> </table> Translation vector T: (+2.47871, +4.99329, +565.868) Focal length f: +17.9330 Radial lens distortion k1: +0.000606815	+0.996727	+0.0808120	+0.00238398	-0.0665790	+0.837192	-0.542841	-0.0458639	+0.540906	+0.839832
+0.996727	+0.0808120	+0.00238398							
-0.0665790	+0.837192	-0.542841							
-0.0458639	+0.540906	+0.839832							

Fig. 17 - Resultados obtidos pela implementação *calcamera* para a terceira calibração efectuada.

Algumas observações podem ser realizadas acerca dos resultados apresentados na Fig. 17:

- Novamente em virtude das condições existentes para o *setup* prático, pode-se considerar que os resultados obtidos apresentam uma qualidade aceitável. Tal qualidade pode ser comprovada pelos ângulos da rotação 3D do sistema mundo para o câmara: segundo o eixo *x*, obteve-se -32.877 quando o ângulo real era de cerca de -30.0 grau; segundo o eixo *y*, obteve-se -0.136 quando o real era de cerca de 0.0 grau; e segundo o eixo *z*, obteve-se 4.635 quando o real era próximo de 5.0 grau. Os valores obtidos para a translação do sistema mundo para o da câmara, também parecem bastante aceitáveis, assim como os obtidos para a distância focal efectiva e para a distorção radial da lente.
- Como nas duas calibrações anteriores, não foi possível utilizar a referida implementação para a determinação das coordenadas correctas do centro da imagem na memória *frame* de forma a

corresponder a resultados aceitáveis. Pensa-se que tal se deve aos mesmos factores antes apontados.

2 - Resultados experimentais obtidos para a determinação de informação tridimensional

Nesta secção são apresentadas duas situações reais em que se pretende a obtenção de informação tridimensional. Na primeira situação, uma câmara com um movimento de translação pura obteve uma sequência de 16 imagens de um dado cubo. Na segunda situação, uma câmara com um movimento de rotação pura obteve uma sequência de 8 imagens do mesmo cubo.

Para cada situação, a câmara encontra-se previamente calibrada e pretende-se determinar informação tridimensional a partir do movimento da mesma. Sendo assim, é necessário determinar as diversas entidades que constituem as várias imagens, de seguida fundir as entidades de cada imagem que sejam próximas e de direcções similares e, finalmente, determinar a informação tridimensional a partir do movimento da câmara e das diversas entidades no plano 2D imagem.

2.1 - Situação real de uma câmara com movimento de translação pura

Nesta situação a câmara utilizada, devidamente calibrada (correspondente à primeira calibração efectuada na secção anterior), move-se com um movimento de translação pura segundo incrementos de 10 mm. Obtém-se desta forma uma sequência composta por 16 imagens de um dado cubo. Na Fig. 18 estão representados os referenciais mundo e câmara (este para a primeira posição considerada), utilizados para esta situação real.

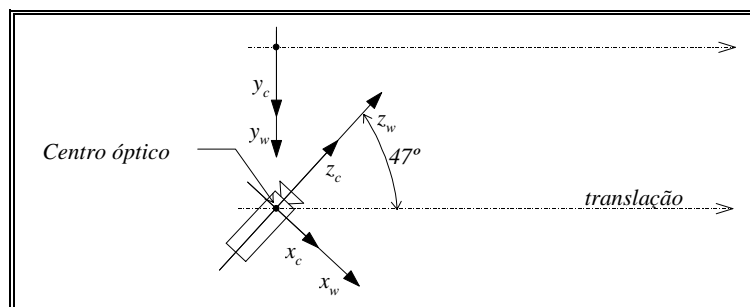


Fig. 18 - Referenciais mundo e câmara adoptados numa situação real de uma câmara com movimento de translação pura.

As imagens obtidas pela câmara para cada posição par² desta, estão representadas nas Figs. 19 a 26.

As entidades a considerar foram determinadas utilizando-se o detector de orlas de intensidade de Deriche para classificar os diversos *pixels* de cada imagem em termos de amplitude e de direcção³, e a implementação *dstrip* para executar o seguimento de linhas e conseqüente aproximação poligonal utilizando faixas dinâmicas. Os parâmetros de entrada utilizados para a implementação *dstrip* foram *tsure*=20, *tprobable*=3, *erdirection*=10, *dadm*=2.0, e os restantes assumindo os respectivos valores por defeito. Nas Figs. 27 a 34, estão representadas as imagens obtidas para as imagens par da sequência.

As diversas entidades obtidas foram sujeitas ao processo *linelink*, com os diversos parâmetros de entrada o mais adequados possível a cada imagem, para fusão daquelas que sejam próximas e de direcções similares. Nas Figs. 35 a 42, estão representadas as imagens obtidas pelo referido processo.

² Optou-se nesta secção por apenas representar as imagens par da respectiva sequência de forma a não alongar este capítulo em demasia.

³ As linhas de intensidade obtidas pela utilização deste detector não apresentavam espessura unitária.

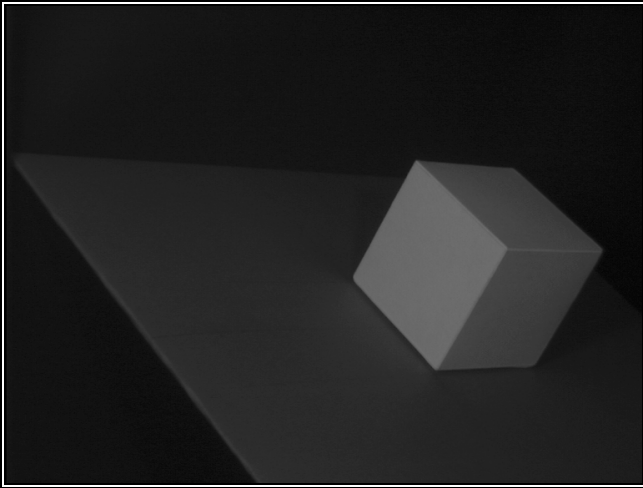


Fig. 19 - 2ª Imagem da sequência.

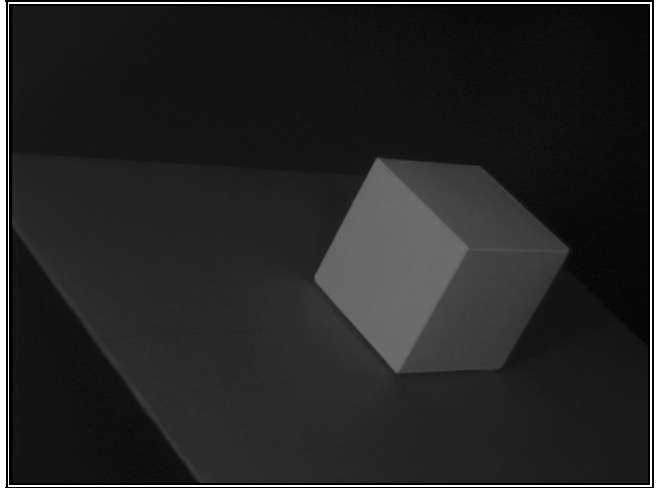


Fig. 20 - 4ª Imagem da sequência.

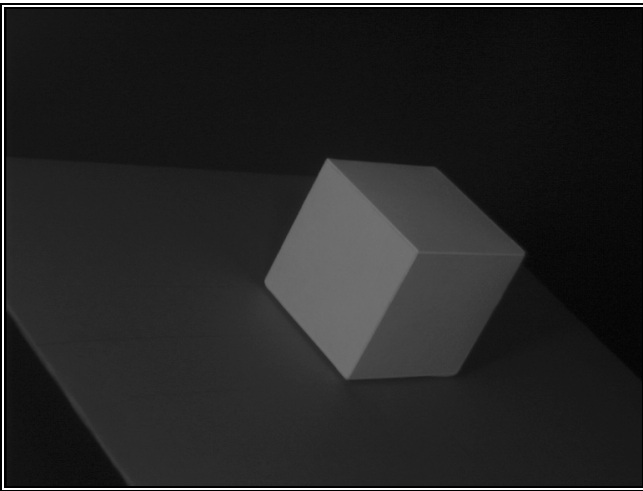


Fig. 21 - 6ª Imagem da sequência.

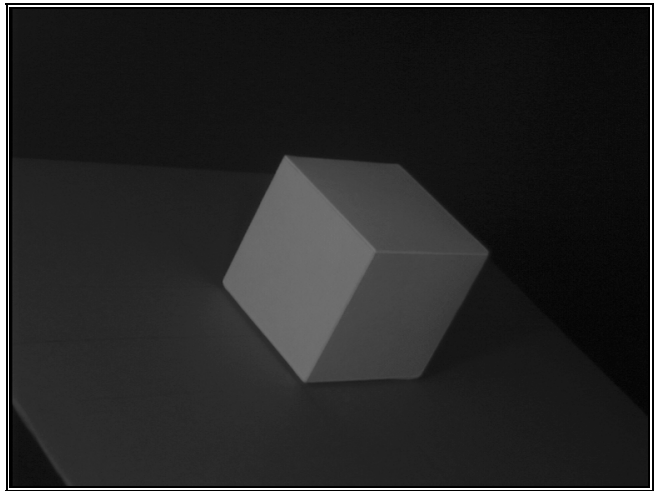


Fig. 22 - 8ª Imagem da sequência.

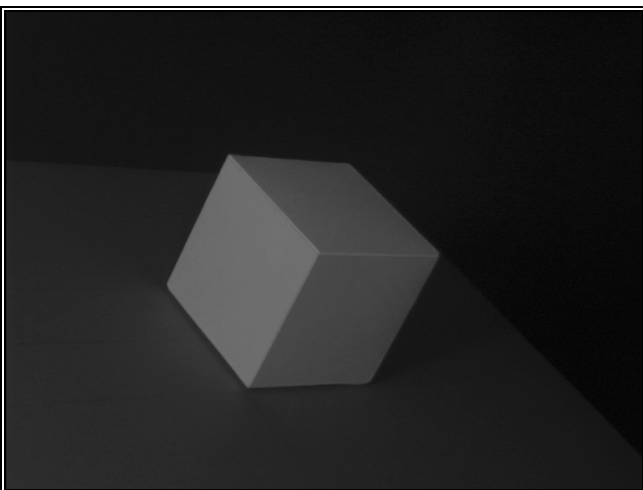


Fig. 23 - 10ª Imagem da sequência.

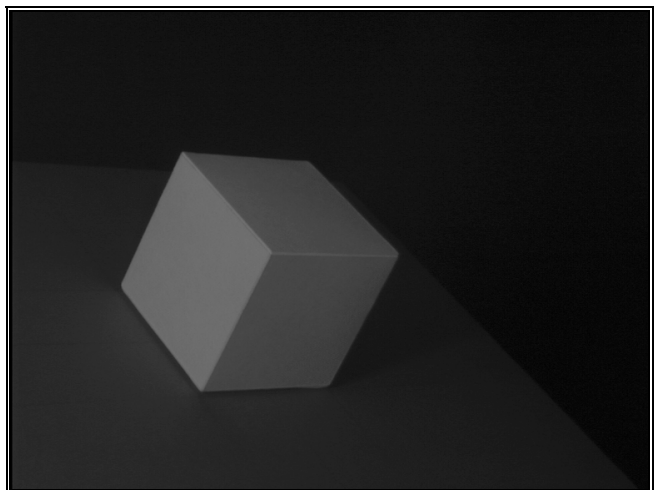


Fig. 24 - 12ª Imagem da sequência.

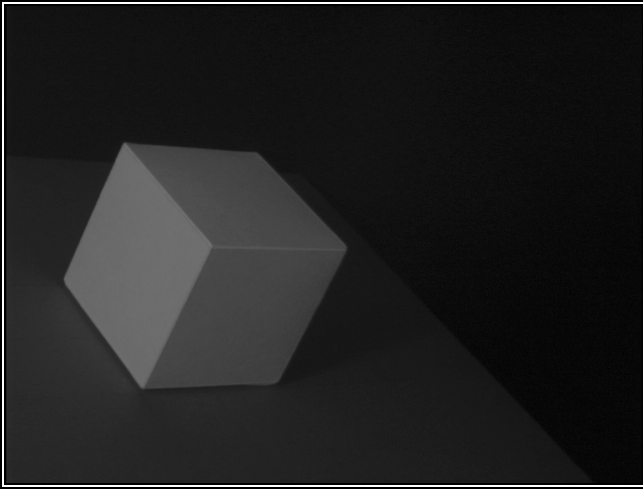


Fig. 25 - 14ª Imagem da sequência.

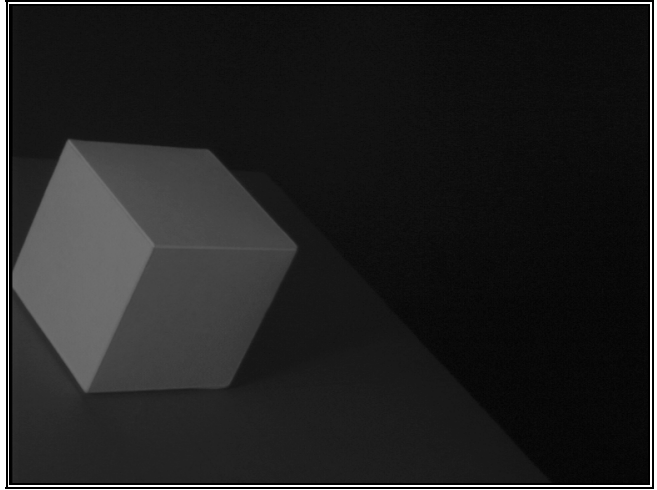


Fig. 26 - 16ª Imagem da sequência.

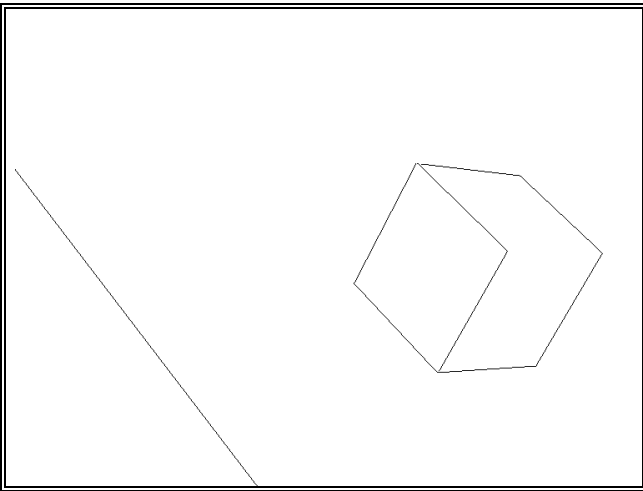


Fig. 27 - Imagem obtida pela implementação **dstrip** para a 2ª imagem da sequência.

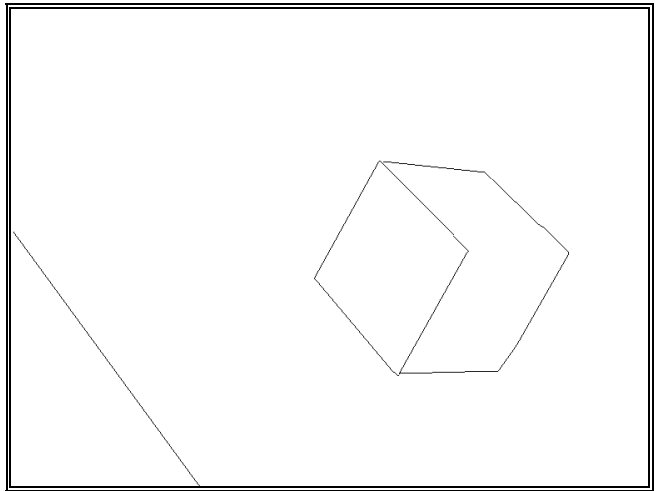


Fig. 28 - Imagem obtida pela implementação **dstrip** para a 4ª imagem da sequência.

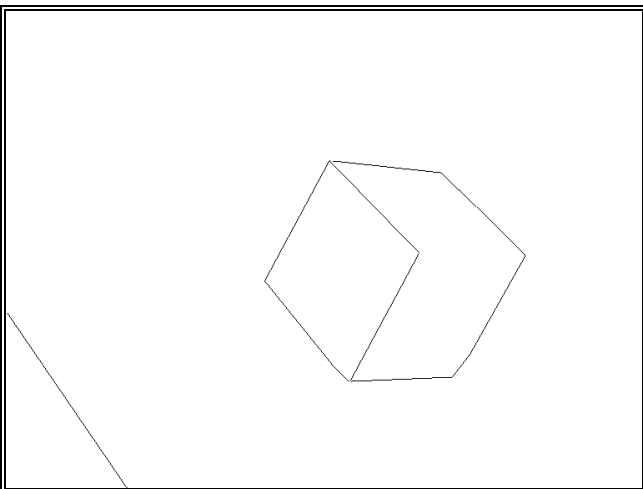


Fig. 29 - Imagem obtida pela implementação **dstrip** para a 6ª imagem da sequência.

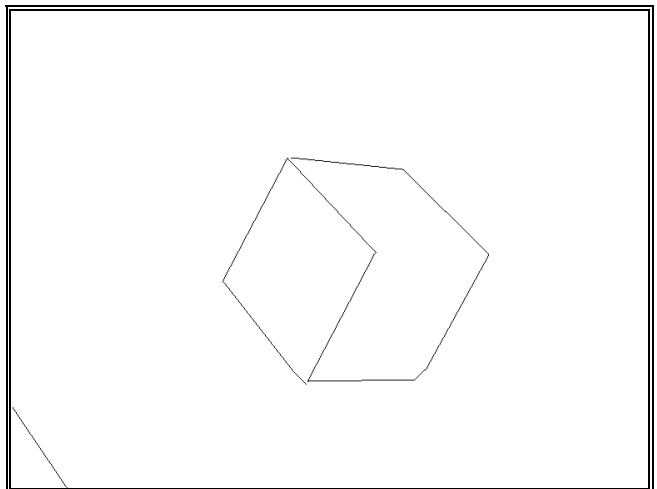


Fig. 30 - Imagem obtida pela implementação **dstrip** para a 8ª imagem da sequência.

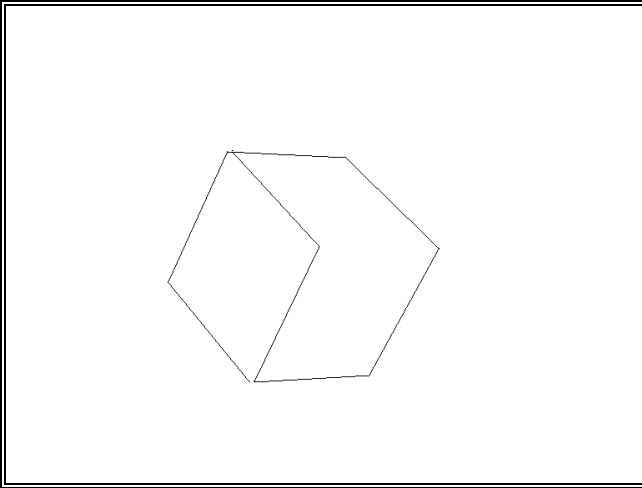


Fig. 31 - Imagem obtida pela implementação **dstrip** para a 10ª imagem da sequência.

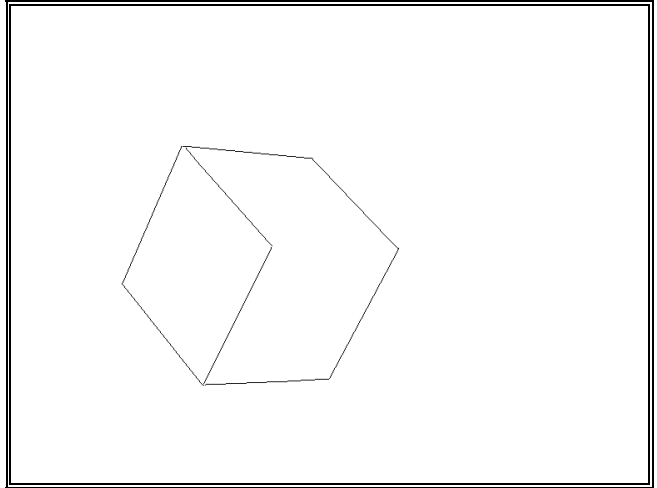


Fig. 32 - Imagem obtida pela implementação **dstrip** para a 12ª imagem da sequência.

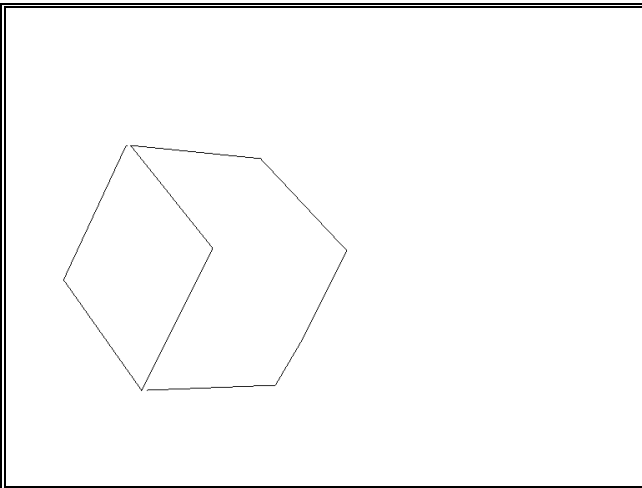


Fig. 33 - Imagem obtida pela implementação **dstrip** para a 14ª imagem da sequência.

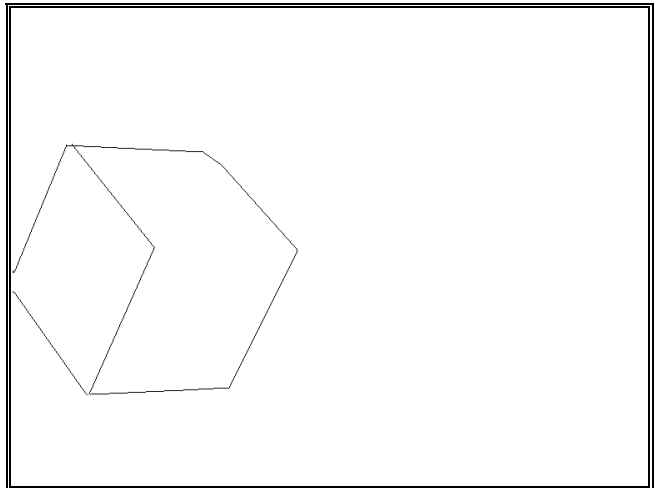


Fig. 34 - Imagem obtida pela implementação **dstrip** para a 16ª imagem da sequência.

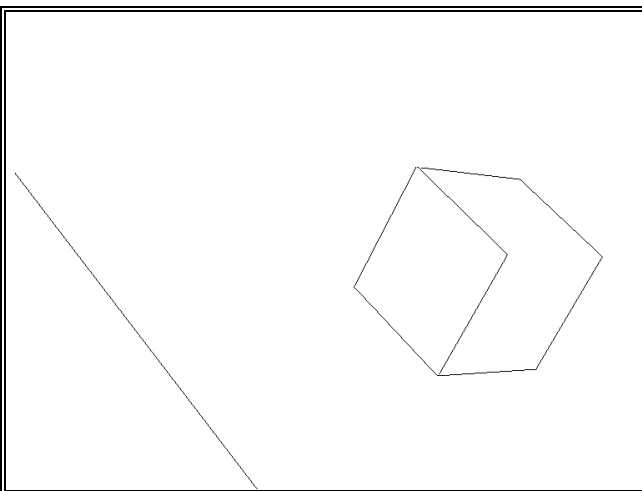


Fig. 35 - Imagem obtida pela implementação **linelink** para as entidades determinadas na 2ª imagem da sequência.

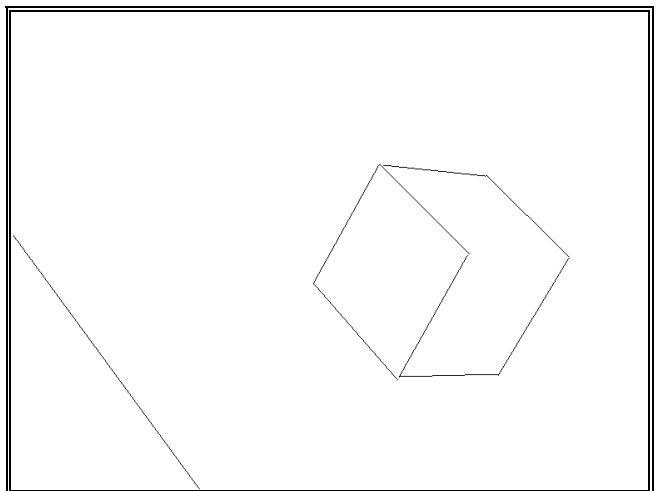


Fig. 36 - Imagem obtida pela implementação **linelink** para as entidades determinadas na 4ª imagem da sequência.

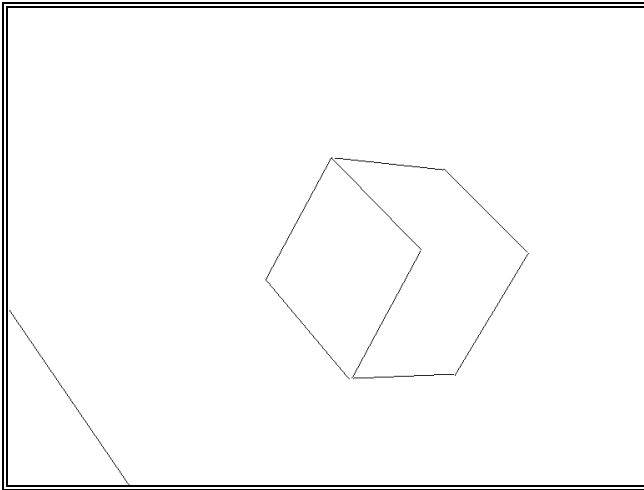


Fig. 37 - Imagem obtida pela implementação **linelink** para as entidades determinadas na 6ª imagem da sequência.

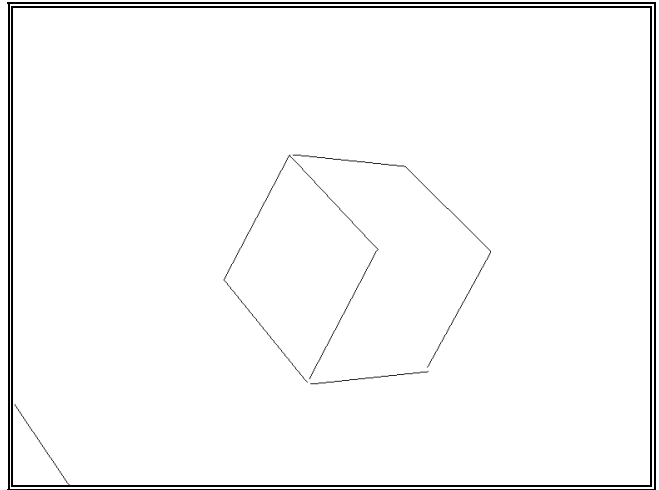


Fig. 38 - Imagem obtida pela implementação **linelink** para as entidades determinadas na 8ª imagem da sequência.

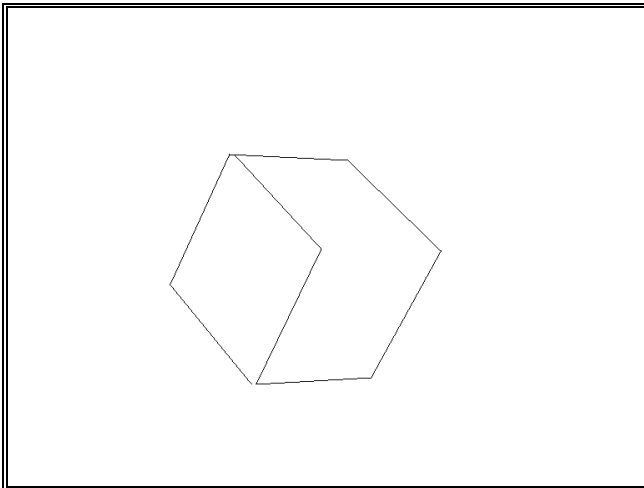


Fig. 39 - Imagem obtida pela implementação **linelink** para as entidades determinadas na 10ª imagem da sequência.

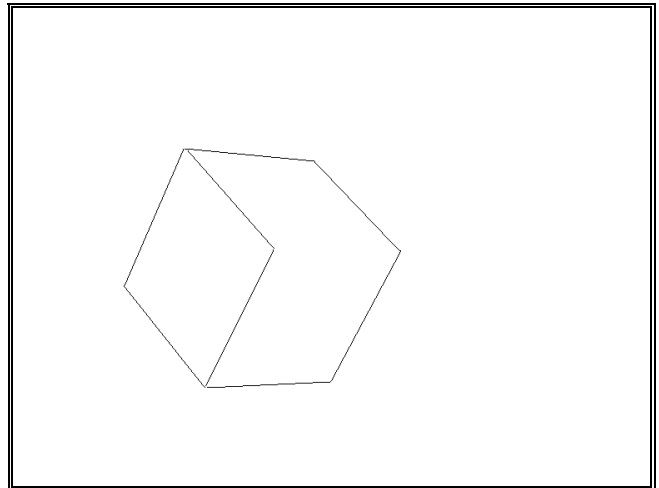


Fig. 40 - Imagem obtida pela implementação **linelink** para as entidades determinadas na 12ª imagem da sequência.

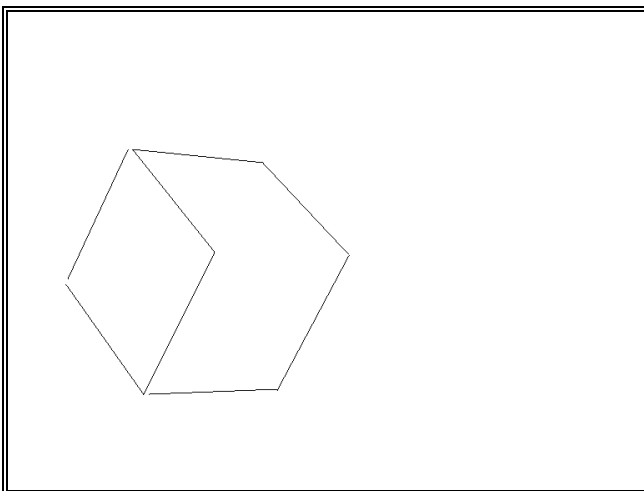


Fig. 41 - Imagem obtida pela implementação **linelink** para as entidades determinadas na 14ª imagem da sequência.

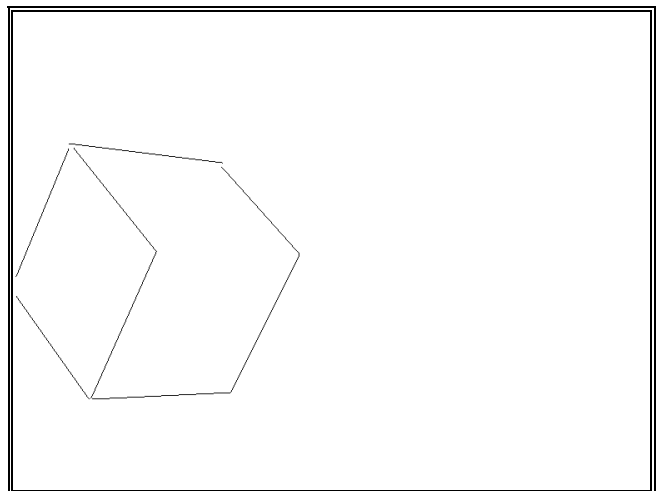


Fig. 42 - Imagem obtida pela implementação **linelink** para as entidades determinadas na 16ª imagem da sequência.

Das Figs. 27 a 42 pode-se concluir que a utilização das implementações *dstrip* e *linelink* forma uma boa solução global para o problema da determinação das entidades com interesse, sendo a sua utilização bastante simples e flexível.

A sequência das entidades foi posteriormente sujeita ao processo *deep* para a obtenção de informação tridimensional. Os parâmetros de entrada utilizados estão apresentados na Tab. IV.

Na Fig. 43 está representada uma imagem em que é possível observar-se o movimento no plano 2D imagem das diversas entidades ao longo da sequência.

Na Fig. 44 estão etiquetadas as diversas entidades consideradas na primeira imagem da sequência.

<i>title</i>	<i>maxDeep</i>	<i>minDeep</i>	<i>nFrameDeep</i>	<i>xSize</i>	<i>ySize</i>	<i>sx</i>
Deep extracted	800.0	350.0	0	765	574	0.983007
<i>focalDist</i>	<i>radialLensDist</i>	<i>dx</i>	<i>dy</i>	<i>xCenterBuf</i>	<i>yCenterBuf</i>	<i>maxDistMedP</i>
15.7665	-0.000766017	0.00837766	0.0080756	382	287	60.0
<i>maxDifLength</i>	<i>maxDifAng</i>	<i>outputFile</i>	<i>verbose</i>	<i>inputFile</i>	<i>outputImage</i>	
60.0	40.0	testet.txt	0	trans.txt	testet.img	

Tab. IV - Parâmetros de entrada para o processo *deep*, utilizados para a sequência.

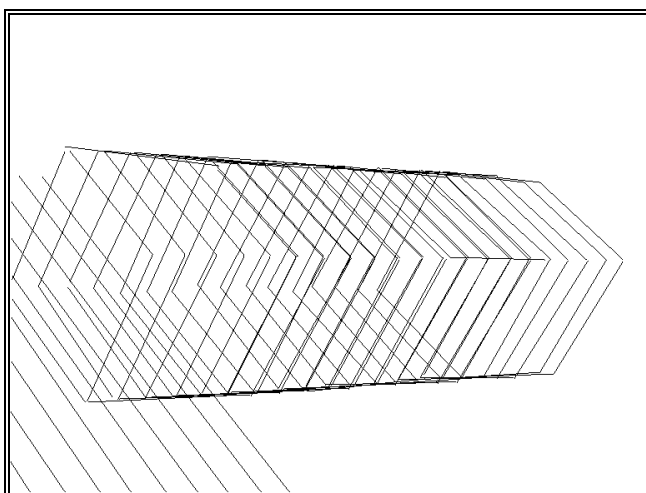


Fig. 43 - Movimento das diversas entidades no plano 2D imagem.

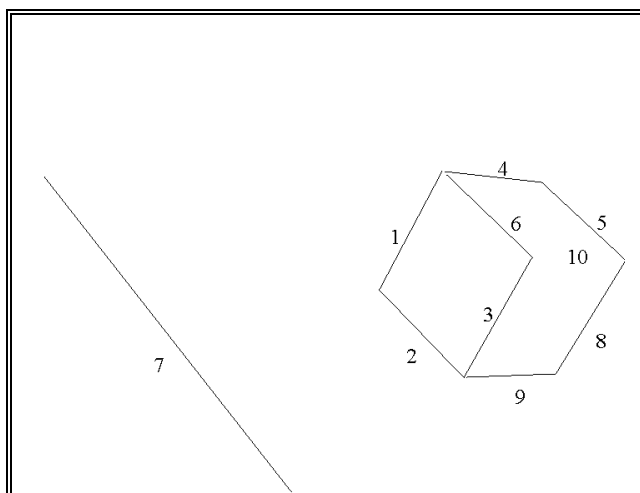


Fig. 44 - Entidades consideradas na 1ª imagem da sequência e sua designação. A entidade 10 não é detectada na referida imagem.

Nas Figs. 45 a 52 estão representadas as bandas ímpar (correspondentes às imagens par da sequência representadas nas Figs. 35 a 42: a primeira imagem da mesma sequência é utilizada na inicialização do processo de seguimento) da imagem resultante do processo *deep*. Nestas bandas é possível observar as diversas entidades seguidas ao longo da sequência, com os *pixels* respectivos com um nível de cinzento equivalente à sua profundidade (neste caso, a coordenada segundo o eixo *z* do referencial mundo).

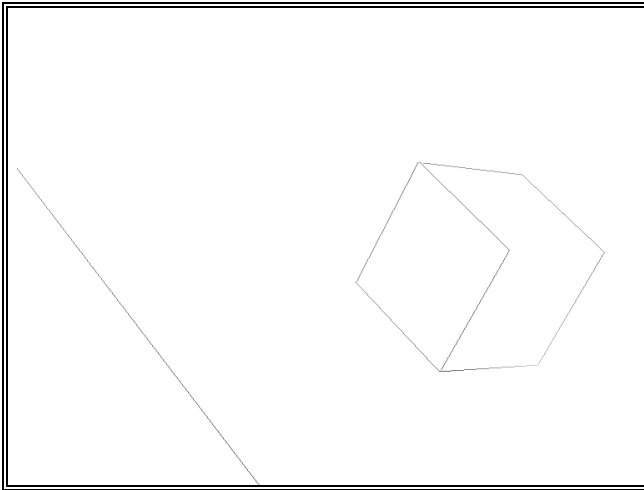


Fig. 45 - 1ª banda da imagem obtida pela implementação *deep* para a sequência.

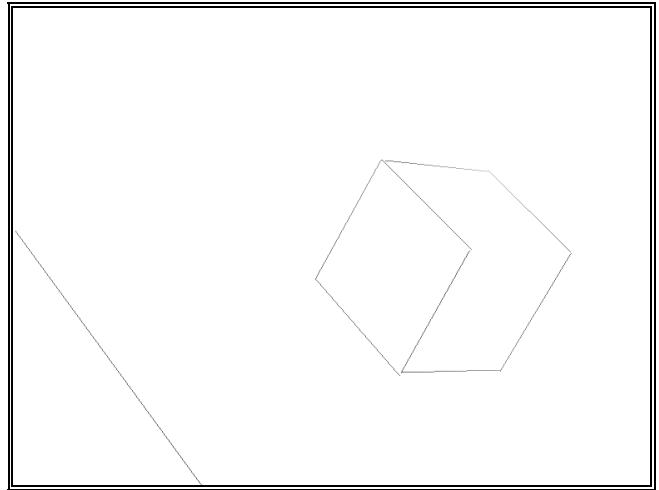


Fig. 46 - 3ª banda da imagem obtida pela implementação *deep* para a sequência.

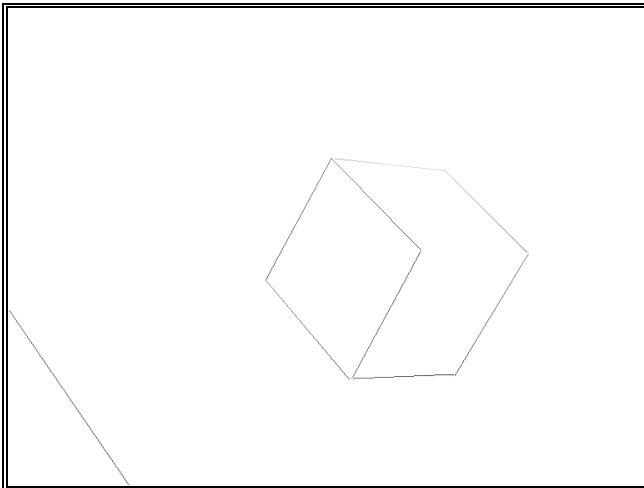


Fig. 47 - 5ª banda da imagem obtida pela implementação *deep* para a sequência.

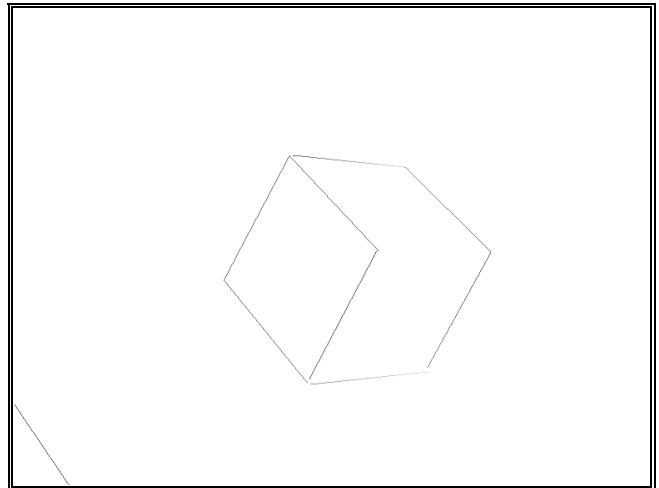


Fig. 48 - 7ª banda da imagem obtida pela implementação *deep* para a sequência.

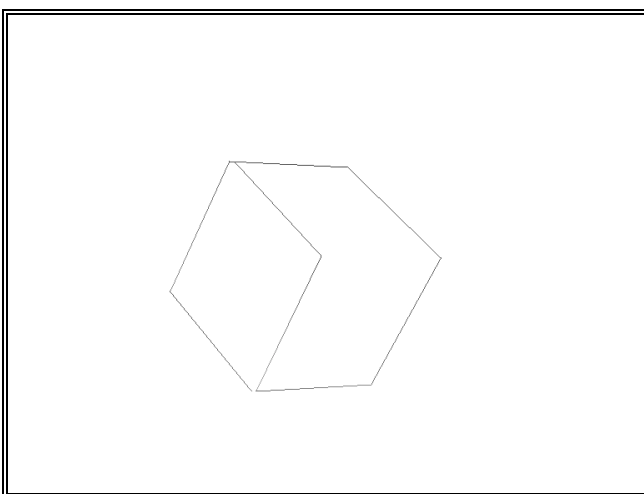


Fig. 49 - 9ª banda da imagem obtida pela implementação *deep* para a sequência.

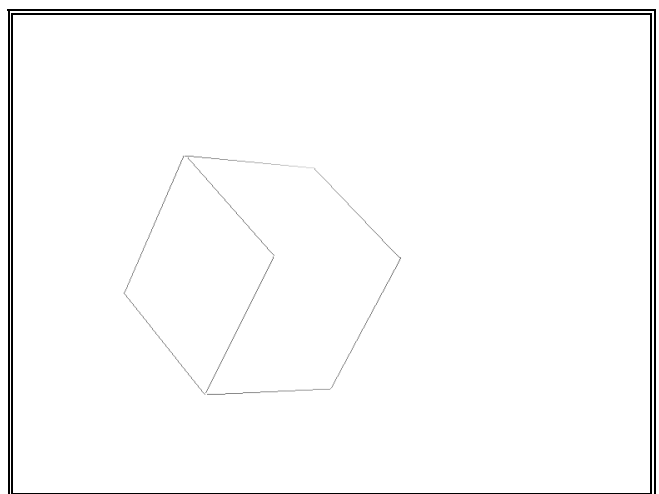


Fig. 50 - 11ª banda da imagem obtida pela implementação *deep* para a sequência.

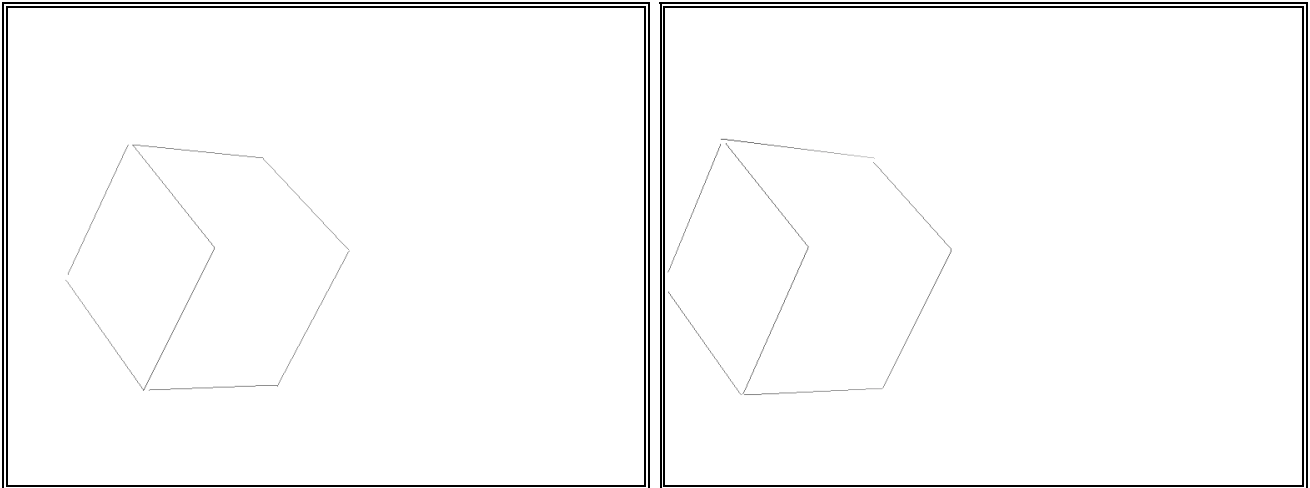


Fig. 51 - 13ª banda da imagem obtida pela implementação *deep* para a sequência. Fig. 52 - 15ª banda da imagem obtida pela implementação *deep* para a sequência.

Na Tab. V estão apresentados os resultados obtidos pelo processo *deep* no seguimento das várias entidades ao longo da sequência e também o modo como decorreu todo este processo.

Como se pode depreender de uma análise cuidadosa dos resultados experimentais apresentados, o processo *deep* obteve resultados de boa qualidade para o seguimento das várias entidades ao longo da sequência. Desses resultados, verifica-se também que a utilização de restrições geométricas como medida do emparelhamento, quando este não é previamente obtido utilizando as distâncias de Mahalanobis, tem como grande vantagem o contínuo êxito do emparelhamento e, conseqüentemente, o respectivo seguimento.

Quanto aos resultados relativos à obtenção de informação tridimensional, pode-se considerar que, para as condições de *setup* prático existentes⁴, são aceitáveis; tal pode ser confirmando pela análise da Fig. 53, que apresenta o conteúdo do ficheiro de texto obtido. São visíveis as coordenadas 2D na memória *frame* e 3D dos pontos inicial e final de cada entidade que tenha um factor de confiança não inferior a 3, que tenha sido emparelhada com êxito na última posição e orientação da câmara e que tenha sido inicializada na primeira posição e orientação da mesma.

Dos resultados pode-se concluir que a metodologia utilizada funciona em pleno.

2.2 - Situação real de uma câmara com movimento de rotação pura

Nesta situação a câmara utilizada, devidamente calibrada (correspondente à terceira calibração efectuada na secção anterior), move-se com um movimento de rotação pura em torno do eixo z do referencial mundo segundo incrementos de 3 grau. Obtém-se desta forma uma sequência composta por 8 imagens de um dado cubo. Na Fig. 54 estão representados os referenciais mundo e câmara (este para a primeira posição considerada), utilizados nesta situação real.

⁴ Por exemplo, não havia o necessário rigor nos valores atribuídos às translações e às rotações da transformação geométrica do sistema mundo para o câmara; também não se conheciam as verdadeiras coordenadas 3D dos pontos inicial e final de cada entidade relativamente ao sistema mundo adoptado, nem o verdadeiro factor de incerteza horizontal do conjunto câmara e placa de aquisição utilizado.

1ª Imagem					2ª Imagem						
Ref.	P. Inic.		P. Fin.		Emp.	Ref.	P. Inic.		P. Fin.		Emp.
1	516	188	441	330	I	1	492	187	418	330	KM
2	441	330	543	436	I	2	418	330	518	437	KM
3	543	436	624	293	I	3	601	293	520	435	KM
4	520	189	636	202	I	4	499	188	617	202	KM
5	636	202	735	294	I	5	617	202	715	294	KM
6	522	193	625	292	I	6	494	187	602	292	KM
7	40	195	336	573	I	7	12	194	302	573	KRG
8	735	297	652	432	I	8	637	429	716	295	KM
9	652	432	546	436	I	9	518	437	637	429	KM
10	Não visível					10	Não visível				
3ª Imagem					4ª Imagem						
Ref.	P. Inic.		P. Fin.		Emp.	Ref.	P. Inic.		P. Fin.		Emp.
1	391	325	464	186	KM	1	364	325	442	184	KM
2	390	326	490	440	KM	2	363	326	464	442	KM
3	493	436	575	293	KM	3	466	439	548	293	KM
4	472	188	585	194	KM	4	447	185	569	198	KM
5	692	294	585	195	KM	5	570	197	669	294	KRG
6	472	189	575	292	KM	6	442	184	550	291	KM
7	3	225	262	573	KRG	7	3	269	226	573	KRG
8	606	437	694	296	KM	8	585	437	670	296	KM
9	604	436	493	436	KRG	9	584	436	466	439	KM
10	Não visível					10	Não visível				
5ª Imagem					6ª Imagem						
Ref.	P. Inic.		P. Fin.		Emp.	Ref.	P. Inic.		P. Fin.		Emp.
1	413	186	340	332	KM	1	311	326	389	181	KM
2	340	332	442	444	KM	2	310	327	410	446	KM
3	442	442	524	294	KM	3	495	293	413	445	KM
4	535	192	413	186	KM	4	393	182	522	196	KM
5	536	191	649	294	KRG	5	523	195	623	294	KRG
6	419	187	529	292	KM	6	389	181	496	291	KM
7	4	314	188	573	KM	7	3	364	146	573	KM
8	569	435	648	295	KM	8	537	442	625	296	KM
9	442	444	569	435	KM	9	535	441	415	446	KRG
10	529	292	642	295	I	10	Não visível				
7ª Imagem					8ª Imagem						
Ref.	P. Inic.		P. Fin.		Emp.	Ref.	P. Inic.		P. Fin.		Emp.
1	284	326	361	180	KM	1	254	326	332	178	KM
2	379	445	284	326	KM	2	253	327	353	450	KM
3	467	295	386	445	KM	3	356	446	437	292	KM
4	487	189	365	183	KM	4	336	178	470	192	KRG
5	488	188	600	293	KRG	5	470	192	573	293	KRG
6	365	183	468	292	KM	6	332	178	438	291	KM
7	3	417	109	573	KM	7	2	477	67	573	KM
8	514	443	600	296	KM	8	573	294	497	432	KRG
9	513	442	379	445	KRG	9	357	452	498	437	KM
10	Não visível					10	Não visível				

Tab. V - Resultados obtidos pelo processo *deep* no seguimento, e modo como decorreu todo este processo. (Continua)

9ª Imagem					10ª Imagem						
Ref.	P. Inic.		P. Fin.		Emp.	Ref.	P. Inic.		P. Fin.		Emp.
1	226	325	299	175	KM	1	266	176	195	332	KM
2	322	449	226	325	KM	2	195	332	293	452	KM
3	409	291	322	449	KM	3	376	290	298	452	KM
4	301	175	446	190	KM	4	408	183	266	176	KRG
5	446	190	548	292	KM	5	520	292	408	183	KRG
6	305	177	409	290	KM	6	273	177	376	289	KM
7	2	537	25	573	KM	7	Não visível				
8	463	445	549	294	KRG	8	437	444	520	293	KM
9	462	444	329	450	KRG	9	298	452	437	444	KM
10	Não visível					10	Não visível				
11ª Imagem					12ª Imagem						
Ref.	P. Inic.		P. Fin.		Emp.	Ref.	P. Inic.		P. Fin.		Emp.
1	237	172	165	332	KM	1	204	170	133	334	KM
2	165	332	262	454	KM	2	133	334	230	456	KM
3	262	454	345	290	KM	3	312	291	231	454	KM
4	238	171	381	182	KM	4	206	170	362	185	KM
5	382	181	494	291	KM	5	362	185	464	292	KM
6	242	174	346	290	KM	6	210	172	313	289	KM
7	Não visível					7	Não visível				
8	407	450	494	294	KM	8	465	292	381	449	KM
9	406	449	266	454	KM	9	233	456	381	449	KM
10	Não visível					10	Não visível				
13ª Imagem					14ª Imagem						
Ref.	P. Inic.		P. Fin.		Emp.	Ref.	P. Inic.		P. Fin.		Emp.
1	102	327	177	168	KM	1	73	321	145	166	KM
2	101	328	197	459	KM	2	163	460	70	328	KM
3	281	290	200	458	KM	3	248	290	163	460	KM
4	182	168	335	184	KM	4	150	166	307	182	KM
5	335	184	438	292	KM	5	307	182	409	291	KM
6	281	289	182	168	KM	6	248	289	150	166	KM
7	Não visível					7	Não visível				
8	438	293	353	452	KM	8	324	455	410	293	KM
9	353	452	202	458	KM	9	323	454	170	460	KM
10	Não visível					10	Não visível				
15ª Imagem					16ª Imagem						
Ref.	P. Inic.		P. Fin.		Emp.	Ref.	P. Inic.		P. Fin.		Emp.
1	35	327	107	164	KM	1	66	165	3	318	KRG
2	123	459	35	327	KM	2	4	342	91	465	KM
3	210	289	130	462	KM	3	171	289	93	464	KM
4	113	164	274	180	KM	4	67	159	251	182	KRG
5	274	180	376	291	KM	5	343	291	250	187	KM
6	210	288	113	164	KM	6	73	164	172	288	KM
7	Não visível					7	Não visível				
8	290	458	378	293	KM	8	343	292	261	457	KM
9	288	457	132	462	KM	9	95	465	261	457	KM
10	Não visível					10	Não visível				

Legenda:
Ref. - Referência da entidade, **I** - Inicialização da entidade, **P. Inic.** - Coordenadas na memória *frame* do ponto inicial no respectivo ficheiro de entrada, **P. Fin.** - Coordenadas na memória *frame* do ponto final no respectivo ficheiro de entrada, **Emp.** - Tipo de emparelhamento: **KM** - Filtros de Kalman e distâncias de Mahalanobis e **KRG** - Filtros de Kalman e restrições geométricas.

Tab. V - Continuação.

Results for deep in transteste.txt with:

Maximum deep: 800
 Minimum deep: 350
 Number of between frames for deep's calculation: 15
 Horizontal factor, sx: 0.983007
 Camera's sensors distance for x, dx: 0.00837766
 Camera's sensors distance for y, dy: 0.0080756
 Focal distance, f: 15.7665
 Radial lens distortion, k: -0.000766017
 Maximum distance of the middle's point for matching: 60
 Maximum difference of length for matching: 60
 Maximum angular difference for matching: 40.

Image coordinates (X,Y), World coordinates (X,Y,Z) of the lines's end points in file: in16.txt

(66, 165)	(37.750641, -27.017143, 540.823120)
(3, 318)	(19.996635, 7.427284, 558.878723)
(4, 342)	(18.028938, 13.317991, 570.159241)
(91, 465)	(43.729435, 39.794781, 538.963440)
(171, 289)	(64.913902, 0.624464, 509.635406)
(93, 464)	(45.527874, 38.760113, 529.965942)
(67, 159)	(38.655693, -27.341322, 536.414978)
(251, 182)	(76.922653, -25.224541, 581.843140)
(343, 291)	(100.481087, 1.159377, 555.041077)
(250, 187)	(77.021088, -24.036171, 576.975098)
(73, 164)	(38.264606, -27.658533, 547.457581)
(172, 288)	(64.260132, 0.424397, 517.539978)
(343, 292)	(100.705849, 1.357344, 544.006836)
(261, 457)	(80.451096, 40.537926, 566.164795)
(95, 465)	(45.520058, 40.077587, 533.309814)
(261, 457)	(80.161713, 41.769165, 571.525085)

Fig. 53 - Conteúdo do ficheiro obtido pela implementação **deep** para a sequência com o parâmetro $nFrameDeep=15$ (não igual a 0 para o conteúdo apresentado não ser demasiado extenso).

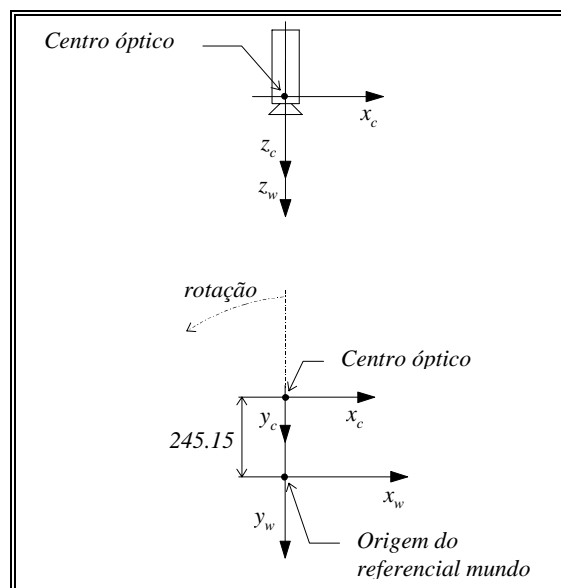


Fig. 54 - Referenciais mundo e câmara adoptados neste caso real de rotação pura da câmara.

As imagens obtidas pela câmara, para cada posição par desta, estão representadas nas *Figs. 55 a 58*.

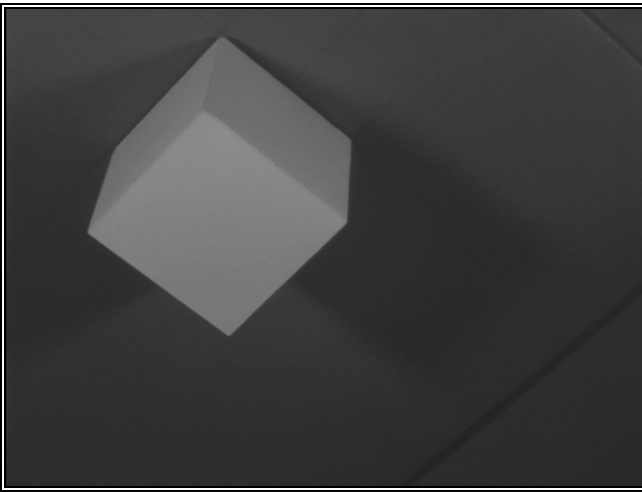


Fig. 55 - 2ª Imagem da sequência.

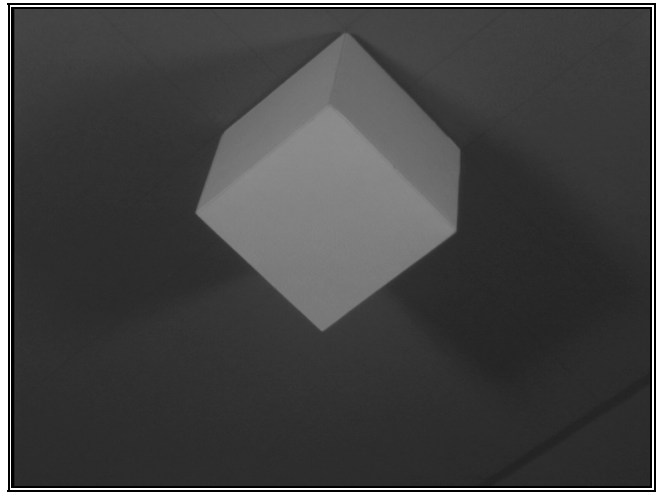


Fig. 56 - 4ª Imagem da sequência.

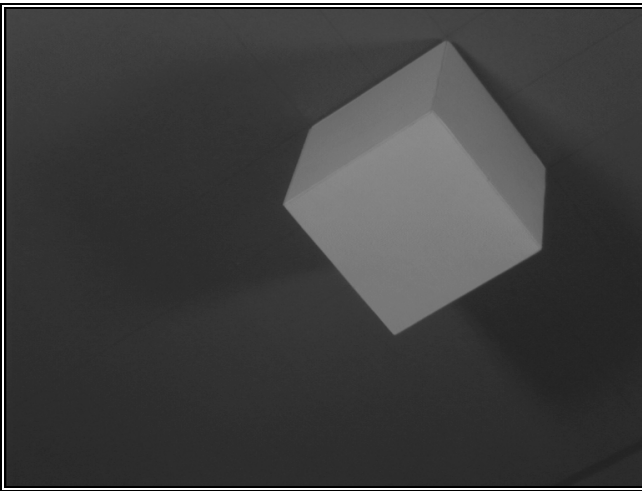


Fig. 57 - 6ª Imagem da sequência.

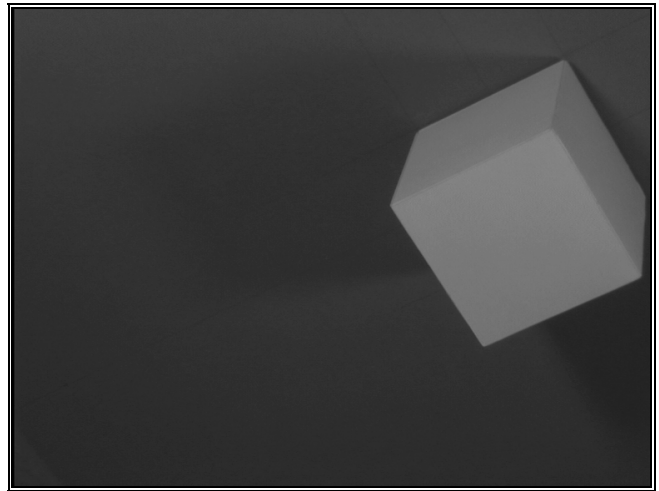


Fig. 58 - 8ª Imagem da sequência.

As entidades a considerar foram determinadas utilizando-se o detector de orlas de intensidade de Deriche⁵ e o processo *dstrip*. Os parâmetros de entrada utilizados foram $tsure=20$, $tprobable=3$, $erdirection=10$, $dadm=2.0$, e os restantes assumindo os respectivos valores por defeito. Nas *Figs. 59 a 62* estão representadas as imagens par obtidas.

As diversas entidades foram sujeitas ao processo *linelink*, com os diversos parâmetros de entrada o mais adequados possível a cada imagem, para fusão daquelas que sejam próximas e de direcções similares. Nas *Figs. 63 a 66* estão representados os resultados obtidos para cada imagem par da sequência.

Das *Figs. 59 a 66* pode-se novamente concluir que a utilização de *dstrip* e *linelink* forma uma boa solução global para o problema da determinação das entidades com interesse nas várias imagens que constituem a sequência em estudo.

A sequência de entidades foi posteriormente sujeita ao processo *deep*, para a obtenção de informação tridimensional. Os parâmetros de entrada utilizados estão apresentados na *Tab. VI*.

⁵ As linhas de intensidade obtidas pela utilização deste detector não apresentavam espessura unitária.

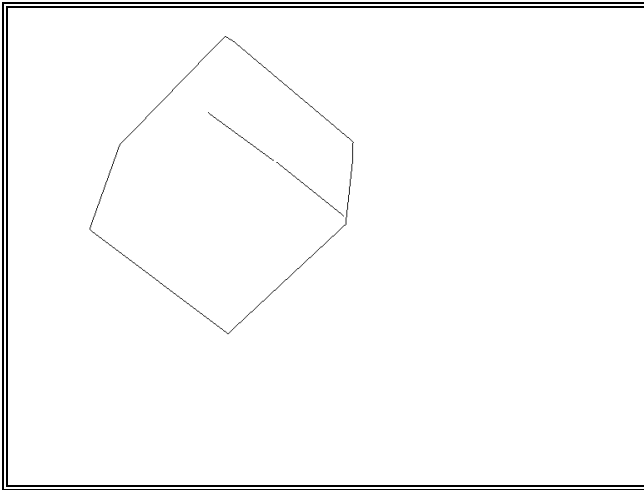


Fig. 59 - Imagem obtida pela implementação **dstrip** para a 2ª imagem da sequência.

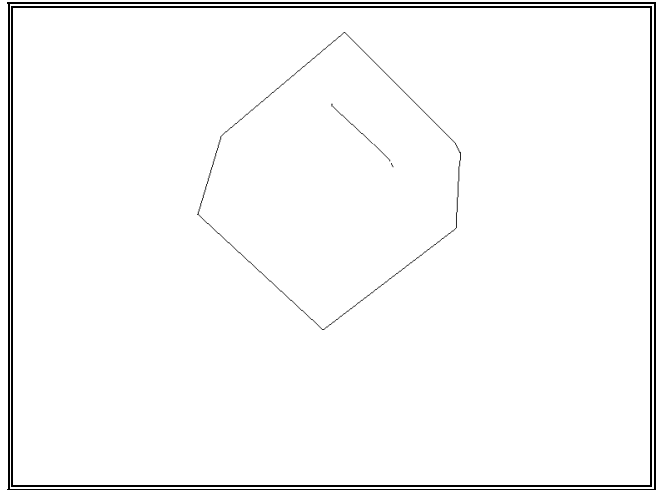


Fig. 60 - Imagem obtida pela implementação **dstrip** para a 4ª imagem da sequência.

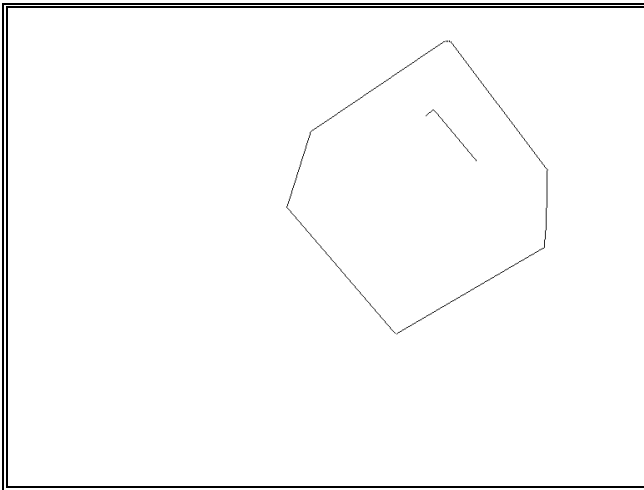


Fig. 61 - Imagem obtida pela implementação **dstrip** para a 6ª imagem da sequência.

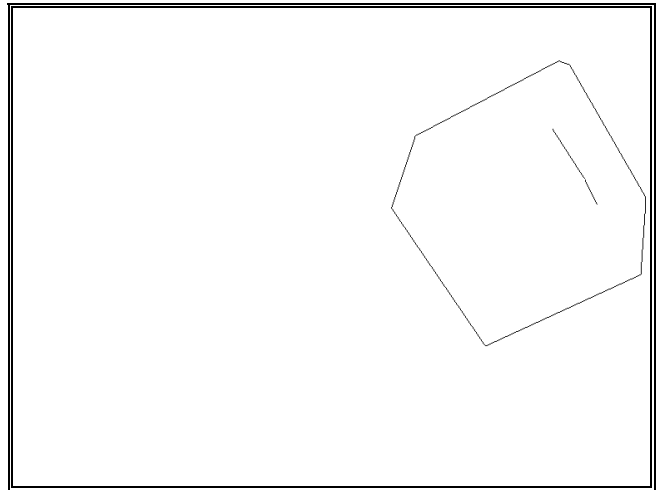


Fig. 62 - Imagem obtida pela implementação **dstrip** para a 8ª imagem da sequência.

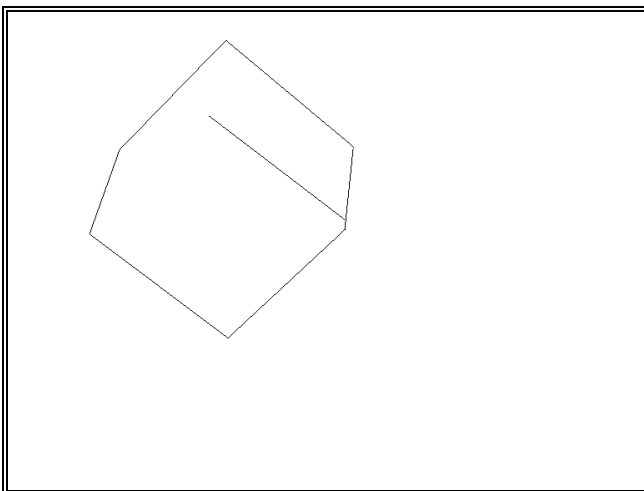


Fig. 63 - Imagem obtida pela implementação **linelink** para as entidades determinadas na 2ª imagem da sequência.

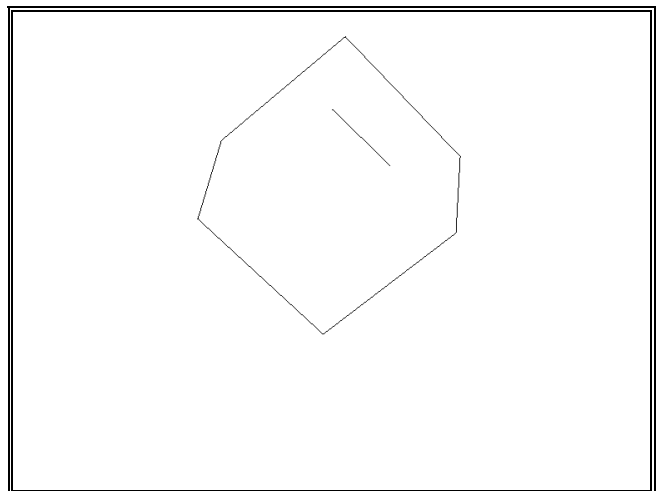


Fig. 64 - Imagem obtida pela implementação **linelink** para as entidades determinadas na 4ª imagem da sequência.

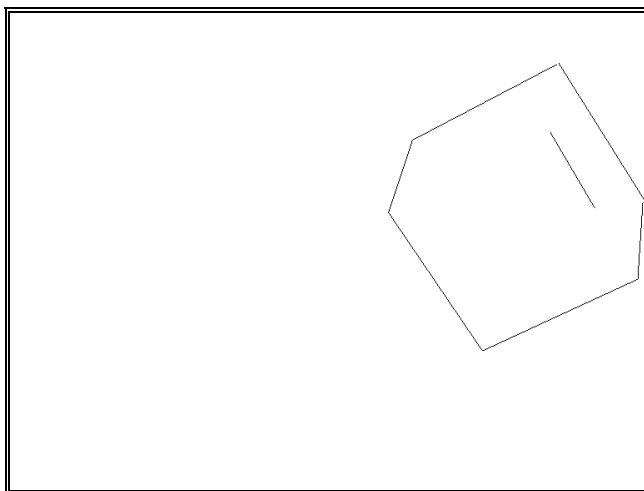
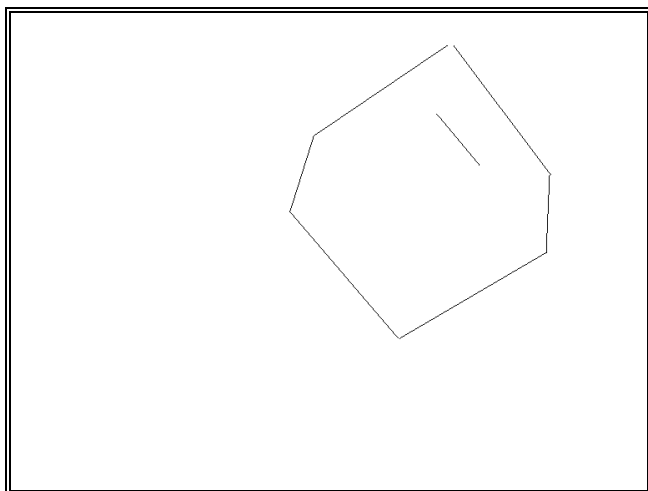


Fig. 65 - Imagem obtida pela implementação **linelink** para as entidades determinadas na 6ª imagem da sequência.

Fig. 66 - Imagem obtida pela implementação **linelink** para as entidades determinadas na 8ª imagem da sequência.

<i>title</i>	<i>maxDeep</i>	<i>minDeep</i>	<i>nFrameDeep</i>	<i>xSize</i>	<i>ySize</i>	<i>sx</i>
Deep extracted	650.0	350.0	0	765	574	0.983007
<i>focalDist</i>	<i>radialLensDist</i>	<i>dx</i>	<i>dy</i>	<i>xCenterBuf</i>	<i>yCenterBuf</i>	<i>maxDistMedP</i>
17.9330	0.000606815	0.00837766	0.0080756	382	287	60.0
<i>maxDifLength</i>	<i>maxDifAng</i>	<i>outputFile</i>	<i>verbose</i>	<i>inputFile</i>	<i>outputImage</i>	
60.0	40.0	tester.txt	0	rotteste.txt	tester.img	

Tab. VI - Parâmetros de entrada para o processo **deep**, utilizados para a sequência de entidades.

Na Fig. 67 está representada uma imagem em que é possível observar-se o movimento no plano 2D imagem das diversas entidades ao longo da sequência.

Na Fig. 68 estão representadas as diversas entidades consideradas na primeira imagem da sequência e a designação que foi atribuída a cada uma na análise dos resultados obtidos.

Nas Figs. 69 a 72 estão representadas as bandas ímpar (correspondentes às imagens par da sequência representadas nas Figs. 63 a 66: a primeira imagem da mesma sequência é utilizada na inicialização do processo de seguimento) da imagem resultante do processo **deep**. Nestas bandas é possível observar as diversas entidades seguidas ao longo da sequência, com os *pixels* respectivos com um nível de cinzento equivalente à sua profundidade (neste caso a coordenada segundo o eixo *z* do referencial mundo).

Na Tab. VII estão apresentados os resultados obtidos no seguimento das várias entidades ao longo da sequência, e também o modo como decorreu todo este processo.

Como é possível deprender pelos resultados experimentais apresentados, a implementação **deep** obteve resultados de boa qualidade para o seguimento das várias entidades ao longo da sequência. Dos mesmos resultados, é também verificável que a utilização de restrições geométricas como medida do emparelhamento, quando este não é previamente obtido com as distâncias de Mahalanobis, tem como grande vantagem o contínuo êxito do emparelhamento e, conseqüentemente, o respectivo seguimento; por outro lado, torna-se a fase de inicialização mais simples e flexível.

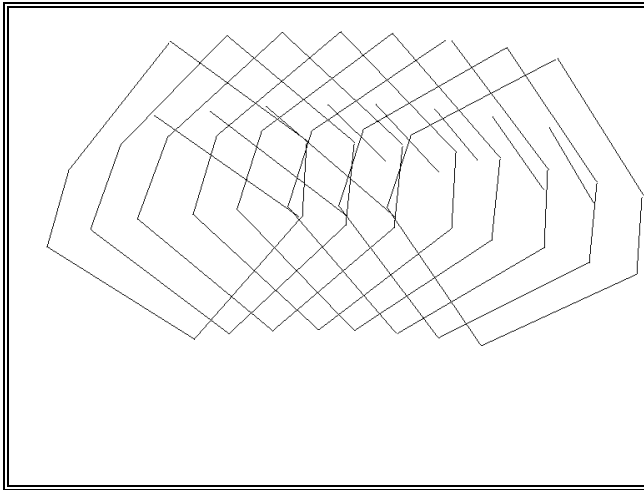


Fig. 67 - Movimento das diversas entidades no plano 2D imagem para a sequência.

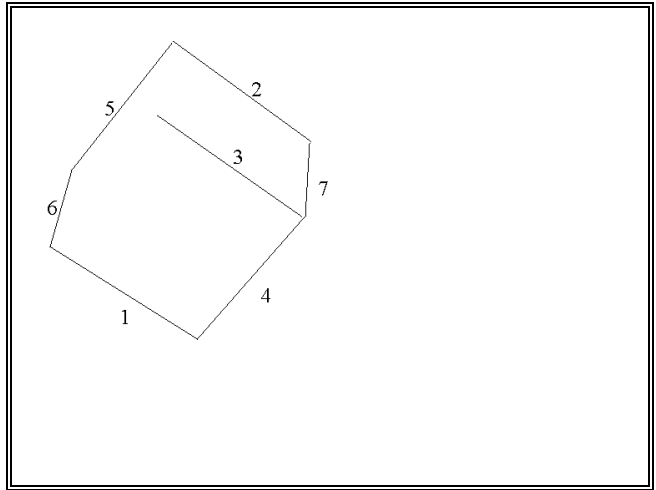


Fig. 68 - Entidades consideradas na 1ª imagem da sequência e sua designação.

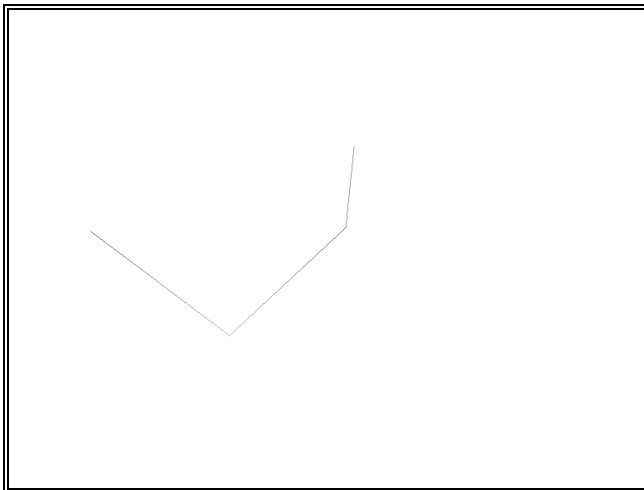


Fig. 69 - 1ª banda da imagem obtida pela implementação *deep* para a sequência.

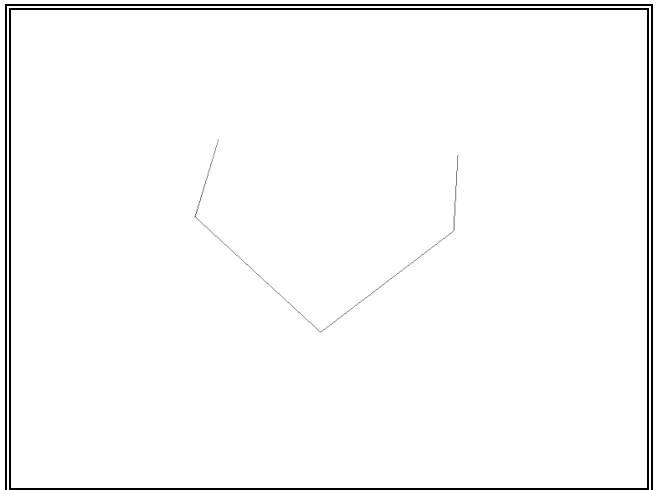


Fig. 70 - 3ª banda da imagem obtida pela implementação *deep* para a sequência.

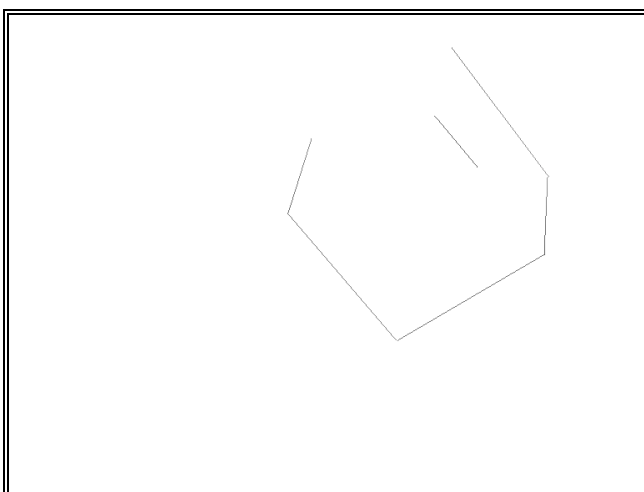


Fig. 71 - 5ª banda da imagem obtida pela implementação *deep* para a sequência.

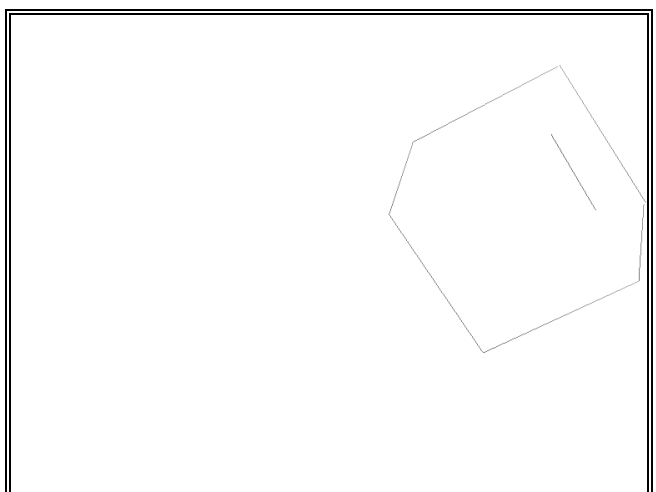


Fig. 72 - 7ª banda da imagem obtida pela implementação *deep* para a sequência.

1ª Imagem						2ª Imagem					
Ref.	P. Inic.		P. Fin.		Emp.	Ref.	P. Inic.		P. Fin.		Emp.
1	46	288	223	399	I	1	99	267	265	392	KRG
2	194	42	359	162	I	2	262	35	415	163	F
3	175	131	348	252	I	3	242	126	404	250	F
4	223	399	353	251	I	4	405	262	265	392	KRG
5	192	44	72	196	I	5	135	166	261	36	F
6	72	196	46	288	I	6	99	267	135	166	F
7	353	251	358	164	I	7	405	262	415	165	KRG
3ª Imagem						4ª Imagem					
Ref.	P. Inic.		P. Fin.		Emp.	Ref.	P. Inic.		P. Fin.		Emp.
1	155	255	317	389	KM	1	221	249	371	387	KM
2	328	31	472	165	F	2	399	30	537	174	F
3	309	120	464	253	F	3	383	118	452	185	F
4	463	265	318	388	KM	4	372	387	532	265	KM
5	192	155	328	31	F	5	398	31	249	156	F
6	155	255	192	155	KRG	6	249	156	221	249	KM
7	463	265	473	168	KM	7	532	265	537	175	KM
5ª Imagem						6ª Imagem					
Ref.	P. Inic.		P. Fin.		Emp.	Ref.	P. Inic.		P. Fin.		Emp.
1	274	242	415	388	KM	1	335	239	465	391	KM
2	461	32	590	181	F	2	532	41	648	195	KM
3	441	117	516	198	KRG	3	562	184	511	123	KRG
4	580	280	416	388	KRG	4	643	289	467	392	KM
5	460	33	303	150	KRG	5	524	41	363	150	KM
6	303	150	274	242	KRG	6	363	150	335	239	KM
7	580	280	590	183	KRG	7	643	289	647	196	KM
7ª Imagem						8ª Imagem					
Ref.	P. Inic.		P. Fin.		Emp.	Ref.	P. Inic.		P. Fin.		Emp.
1	396	238	516	397	KM	1	454	240	567	406	KM
2	598	49	706	210	KM	2	659	62	762	226	KM
3	581	132	642	219	KRG	3	649	145	702	235	KRG
4	696	307	517	397	KM	4	567	406	754	320	KM
5	597	50	425	148	KM	5	656	64	482	155	KM
6	425	148	396	238	KM	6	482	155	454	240	KM
7	697	307	706	212	KM	7	754	320	760	229	KM

Legenda:
Ref. - Referência da entidade, **I** - Inicialização da entidade, **P. Inic.** - Coordenadas na memória *frame* do ponto inicial no respectivo ficheiro de entrada, **P. Fin.** - Coordenadas na memória *frame* do ponto final no respectivo ficheiro de entrada, **Emp.** - Tipo de emparelhamento: **KM** - Filtros de Kalman e distâncias de Mahalanobis, **KRG** - Filtros de Kalman e restrições geométricas e **F** - Não obtido com êxito.

Tab. VII - Resultados obtidos pela implementação *deep* no seguimento, e modo como decorreu todo este processo.

Quanto aos resultados obtidos pelo processo *deep* no que respeita à obtenção de informação tridimensional, pode-se considerar que, para as condições de *setup* prático existentes, são aceitáveis; tal pode ser confirmando pela análise da Fig. 73 que apresenta o conteúdo do ficheiro de texto obtido. São visíveis as coordenadas 2D na memória *frame* e 3D dos pontos inicial e final de cada entidade que tenha um factor de confiança não inferior a 3, que tenha sido emparelhada com êxito na quarta posição e orientação da câmara e que tenha sido inicializada na primeira posição e orientação da mesma; e também as mesmas coordenadas para cada entidade que tenha um factor de confiança não inferior a 3, que tenha sido emparelhada com êxito na última posição e orientação da câmara e que tenha sido inicializada pelo menos na quarta posição e orientação da mesma.

Dos mesmos resultados também se pode concluir que a metodologia utilizada funciona em pleno.

```

Results for deep in rotteste.txt with:
    Maximum deep: 650
    Minimum deep: 350
    Number of between frames for deep's calculation: 3
    Horizontal factor, sx: 0.983007
    Camera's sensors distance for x, dx: 0.00837766
    Camera's sensors distance for y, dy: 0.0080756
    Focal distance, f: 17.933
    Radial lens distortion, k: 0.000606815
    Maximum distance of the middle's point for matching: 60
    Maximum difference of length for matching: 60
    Maximum angular difference for matching: 40.

Image coordinates (X,Y), World coordinates (X,Y,Z) of the lines's end points in file: rotin4.txt
(221, 249)      (-75.332237, -244.738297, 487.976593)
(371, 387)      (-37.287018, -218.721375, 516.945740)
(372, 387)      (-37.087898, -219.397110, 502.822571)
(532, 265)      (-7.118646, -251.790115, 469.298615)
(532, 265)      (-6.440199, -251.999924, 479.493317)
(537, 175)      (-5.755922, -273.961365, 521.450256)

Image coordinates (X,Y), World coordinates (X,Y,Z) of the lines's end points in file: rotin8.txt
(454, 240)      (-76.182137, -244.712387, 501.143707)
(567, 406)      (-37.162891, -218.791061, 513.523193)
(567, 406)      (-37.186592, -218.795776, 513.283081)
(754, 320)      (-4.639064, -252.773331, 502.030212)
(754, 320)      (-5.179496, -252.618073, 498.769806)
(760, 229)      (-4.534945, -276.115845, 542.385071)
(482, 155)      (-76.283974, -267.460876, 535.795227)
(454, 240)      (-76.585129, -244.165298, 483.840790)
(659, 62)       (-43.913315, -303.577240, 530.225708)
(762, 226)      (-6.945745, -275.477448, 525.366943)
(656, 64)       (-44.447933, -302.829926, 530.021973)
(482, 155)      (-76.314041, -267.360596, 534.402954)
(649, 145)      (-45.807850, -276.463654, 457.544708)
(702, 235)      (-26.127522, -264.582458, 477.180359)
    
```

Fig. 73 - Conteúdo do ficheiro obtido pela implementação *deep* para a sequência, com o parâmetro *nFrameDeep=3* (não igual a 0 para o conteúdo apresentado não ser demasiado extenso).

3 - Algumas conclusões finais

Neste capítulo são apresentados alguns resultados experimentais obtidos pela abordagem global utilizada nesta dissertação, para a obtenção de informação tridimensional a partir de dois tipos de movimento de uma dada câmara.

Antes de se proceder à obtenção de informação tridimensional, é necessário calibrar a câmara a utilizar na obtenção das respectivas sequências de imagens. Assim, neste capítulo são apresentados os resultados de calibração obtidos pela implementação *calcamera*, para as duas situações reais que são consideradas. Obviamente que, nestas calibrações, é também indispensável a determinação das coordenadas dos pontos de calibração considerados na memória *frame*; assim, são também apresentados os resultados obtidos para a referida determinação utilizando-se, para tal, a implementação *calpoint*.

Os resultados obtidos na determinação das coordenadas na memória *frame* podem ser classificados como de boa qualidade. Quanto aos obtidos para as calibrações, podem ser classificados, atendendo às condições do *setup* prático existentes, como também de boa qualidade; no entanto, devido a estas condições, e principalmente devido à baixa distorção radial da lente, não foi possível determinar-se as coordenadas correctas do centro da imagem na memória *frame*.

Após as calibrações da câmara, fez-se então a aquisição das respectivas sequências. Uma sequência foi obtida a partir de um movimento de translação pura da câmara; obteve-se uma sequência de 16 imagens de um dado cubo. A outra, foi obtida a partir de um movimento de rotação pura da mesma câmara; obteve-se uma sequência de 8 imagens de um cubo.

As entidades a considerar - segmentos de recta - em cada imagem que constitui cada sequência, foram determinadas utilizando-se para tal: o detector de orlas de intensidade de Deriche, para classificar cada *pixel* de cada imagem em termos de amplitude e de direcção; um seguimento de linhas baseado na direcção e em dois níveis de amplitude e consequente aproximação poligonal utilizando faixas dinâmicas (cada imagem ficou assim constituída por um dado conjunto de segmentos de recta); uma fusão dos segmentos de recta próximos e de direcções similares.

Os resultados obtidos nesta determinação podem ser classificados como bastante satisfatórios, e o processo decorre de forma bastante simples e flexível.

Após a determinação das entidades a considerar em cada imagem, realizou-se o seguimento das mesmas ao longo da sequência. Este seguimento foi realizado utilizando-se o processo *deep*. Sempre que uma entidade é emparelhada com êxito em duas imagens distintas da mesma, pode obter-se as coordenadas 3D dos seus pontos extremos. Uma aproximação da profundidade dos restantes pontos de cada entidade foi obtida através de interpolação linear da profundidades dos seus pontos inicial e final.

No que respeita ao seguimento os resultados obtidos podem ser classificados como bastante satisfatórios para as duas sequências, sendo de referir que o processo não foi optimizado para nenhuma das sequências, sendo assim possível obter-se resultados ainda de melhor qualidade; bastaria para tal, afinar o processo de seguimento para cada uma das sequências a considerar.

Dos mesmos resultados conclui-se também que a utilização de restrições geométricas como medida do emparelhamento, quando este não é previamente obtido com as distâncias de Mahalanobis, tem como grande vantagem o contínuo êxito do emparelhamento e, consequentemente, o respectivo seguimento e, por outro lado, torna mais fácil e flexível a fase de inicialização das entidades consideradas. Também se depreende que a filosofia utilizada na medição e posterior actualização é correcta, pois mesmo nos casos em que uma dada entidade é não visível numa imagem e visível na seguinte, o seu seguimento é obtido com êxito.

Quanto aos resultados obtidos pela implementação *deep* no que respeita à obtenção de informação tridimensional, pode-se considerar que, para as condições existentes de *setup*, são aceitáveis.

Capítulo VI

CONCLUSÕES FINAIS E PERSPECTIVAS DE DESENVOLVIMENTO FUTURO

1 - Algumas conclusões finais

Nesta dissertação é apresentada a abordagem global concebida e implementada, para a obtenção de estrutura tridimensional a partir de movimento de uma câmara. Nesta abordagem é pressuposto que o movimento da câmara utilizada é perfeitamente conhecido, que o ambiente considerado é do tipo industrial, e que as entidades a considerar em tal ambiente são segmentos de recta.

A metodologia desenvolvida é basicamente constituída por três fases: na primeira, é determinada a transformação que ocorre na câmara de um ponto 3D no espaço para um ponto 2D no plano imagem - calibração da câmara; na segunda, são determinadas as entidades consideradas - segmentos de recta - em cada imagem da sequência; e, na terceira e última fase, as entidades são seguidas ao longo da sequência e são obtidas as coordenadas 3D dos pontos das entidades.

A calibração é constituída por três etapas: na primeira, é determinado o factor de incerteza horizontal; na segunda, são determinadas as coordenadas correctas do centro da imagem na memória *frame*; e na terceira, são determinados os restantes parâmetros intrínsecos e extrínsecos. Durante o desenvolvimento de ferramentas para esta fase tornou-se de grande utilidade o simulador realizado para câmaras de geometria interna conhecida. Com este simulador foi possível testar o método de calibração concebido (para todos os parâmetros, a menos do factor de incerteza horizontal) sem a necessidade de montar todo o *setup* prático, constituindo assim, uma solução mais flexível e onde todas as variáveis envolvidas eram facilmente controladas.

Para a determinação das entidades é utilizado o filtro de Deriche, para detecção de orlas de intensidade. Sobre os resultados obtidos é executado o seguimento e subsequente aproximação poligonal, utilizando faixas dinâmicas, das várias linhas detectadas. Finalmente, sobre os resultados obtidos pelo processo de seguimento e aproximação, é realizada a fusão dos segmentos de recta que sejam presumivelmente pertencentes a uma mesma entidade. Desta forma, cada imagem original fica reduzida a um conjunto de segmentos de recta, de características fortes, representando as entidades com interesse presentes.

A fase de seguimento das entidades ao longo de uma sequência e consequente obtenção de informação tridimensional é solucionada do seguinte modo: entidades parametrizadas pelo ponto médio, o comprimento e a direcção de cada segmento de recta; introdução de ruído na modelação do sistema; utilização de três filtros de Kalman independentes, cada um com um modelo cinemático associado de aceleração constante localmente; utilização de distâncias normalizadas de Mahalanobis ou, quando desta forma o emparelhamento não é obtido com êxito, restrições geométricas como medida do grau de emparelhamento; utilização do princípio da triangulação estereoscópica, após a determinação da verdadeira localização dos pontos extremos do segmento de recta, visíveis numa dada imagem, sobre o segmento visível na imagem anterior a esta, para a obtenção de informação tridimensional; utilização de um modelo de entidades continuamente actualizado. A filosofia utilizada na medição e posterior actualização revelou-se correcta, pois mesmo nos casos em que uma dada entidade não é visível numa imagem e é visível na seguinte, o seu seguimento é obtido com êxito. Também se verificou que, quando os filtros utilizados começam a divergir ou a convergir em demasia, a utilização de restrições geométricas origina que os filtros se autoajustem de forma mais rápida, e mantenham emparelhamentos com êxito ao longo do processo de seguimento. A introdução de ruído na modelação do sistema resolveu de forma satisfatória o problema de os filtros poderem convergir em demasia. Também se verificou que a utilização de restrições geométricas teve como consequência tornar mais fácil e flexível a fase de inicialização das entidades consideradas. A utilização do factor de confiança para as entidades presentes no modelo revelou-se de grande utilidade, nomeadamente para o manter sempre actualizado, evitando assim que este crescesse para valores muito elevados e desnecessários.

A metodologia desenvolvida foi aplicada em duas situações reais. Para cada uma das situações foram efectuadas: a calibração da câmara, a menos do factor horizontal de incerteza, pois não foi possível desligar o filtro passa-baixo da câmara disponível; a determinação das entidades em cada imagem da sequência; o seguimento das entidades ao longo da sequência e a obtenção de informação tridimensional. Atendendo às condições práticas existentes, os resultados obtidos em cada fase podem ser considerados de boa qualidade. No entanto, não foi possível determinar-se as coordenadas

correctas do centro da imagem na memória *frame*, utilizando-se o método concebido para tal efeito. Pensa-se que duas das causas para tal impossibilidade foram a baixa distorção radial da lente utilizada e a grande distância focal considerada.

A abordagem desenvolvida, durante o trabalho desta dissertação, ficou comprovada como aplicável de forma satisfatória às situações nas quais se pretende obter informação tridimensional de uma cena quando o movimento da câmara é conhecido e as entidades a considerar são segmentos de recta, tal como em cenas tipicamente industriais. Conseguiu-se assim estudar e desenvolver com êxito um conjunto de ferramentas para execução de tal tarefa.

2 - Perspectivas de desenvolvimento futuro

Apesar da abordagem global adoptada e implementada ao longo desta dissertação produzir resultados experimentais bastante satisfatórios e promissores, a mesma pode ser enriquecida e melhorada, nomeadamente, nos seguintes pontos:

- *Calibração da câmara* - O método implementado para a calibração de uma câmara, exceptuando o factor de incerteza horizontal, não pôde ser usado, nas experiências realizadas, para a obtenção da verdadeira localização do centro da imagem na memória *frame*, devido à baixa distorção radial da lente utilizada. O método adoptado para a calibração do factor de incerteza horizontal revelou-se de utilização difícil para as, provavelmente fracas, condições práticas normalmente existentes. Desta forma, pensa-se que um método de calibração que provavelmente ultrapassará tais inconvenientes seja o proposto em [Batista, 1993] e que foi também apresentado no segundo capítulo desta publicação. No entanto, pensa-se que talvez seja mais conveniente determinar os ângulos da rotação 3D envolvida na transformação geométrica do sistema mundo para o da câmara, não como nesse método é realizada, mas antes como é efectuada no método proposto em [Tsai, 1987] para pontos de calibração coplanares, também apresentado no mesmo capítulo. Se apenas se pretender melhorar a determinação da localização correcta do centro da imagem na memória *frame*, então pensa-se que uma boa solução passará por adoptar um dos métodos baseados na variação da distância focal que foram propostos em [Lenz, 1988] e que também estão apresentados no mesmo capítulo.
- *Seguimento das entidades e obtenção de informação 3D* - A solução adoptada e implementada ao longo desta dissertação para tal seguimento e obtenção de informação 3D, apesar de obter resultados bastante satisfatórios, pode ainda ser enriquecida. Uma solução que pode ser adoptada para tal melhoria, poderá passar, em parte, pela solução proposta em [Crowley, 1992]. Assim, a solução adoptada pode ser enriquecida se for incluindo um modelo 3D de entidades continuamente actualizado. As entidades 3D são também segmentos de recta devidamente parametrizados, sendo as respectivas características de parametrização devidamente refinadas pela utilização de filtros de Kalman que considerem toda a historia dos mesmos ao longo da respectiva sequência. As referidas entidades poderiam ser parametrizadas pela posição do ponto médio, neste caso com três coordenadas, por um dos seus vectores devidamente normalizado, de forma a definir as direcções, e pelo comprimento. Sempre que se obtêm as coordenadas 3D dos pontos inicial e final de uma entidade, a partir do seu seguimento 2D, que ainda não esteja incluída no respectivo modelo 3D, ela é inserida neste, devidamente inicializada, e utilizando-se a mesma identificação do modelo 2D. Sempre que uma entidade da qual se obtêm as coordenadas 3D dos pontos inicial e final, a partir do seu seguimento 2D, já exista no modelo 3D, a mesma é devidamente actualizada, utilizando para tal os respectivos filtros de Kalman. Com esta solução torna-se possível permitir o emparelhamento de várias entidades 2D com uma única entidade 3D de forma bastante simples; tal pode ser obtido fazendo uma projecção de cada entidade do modelo 3D no plano imagem da câmara para a respectiva posição e orientação que está a ser considerada e assim facilitar o referido emparelhamento. O modelo de entidades 3D é actualizado de forma diferente do modelo 2D; assim, sempre que uma entidade é inserida, assume um factor de

confiança de valor médio; sempre que é visível, aumenta este factor em uma unidade até ao valor máximo admissível; no entanto, se não for visível na imagem seguinte, tal factor só será diminuído em uma unidade até ao valor mínimo admissível, após o qual será removida do modelo 3D, se o mesmo factor ainda não for igual ao valor máximo admissível. Deste modo, cada entidade que foi visível pelo menos durante um certo número de imagens sucessivas, nunca será removida do modelo pois é considerado que tal entidade existe na respectiva cena, tornando-se invisível unicamente devido às novas posições e orientações que estão a ser consideradas para a câmara. Obtém-se, desta forma, um modelo de toda a cena que pode ser utilizado na obtenção da estrutura da mesma. Na Fig. 1, está apresentada de forma esquemática a solução que se pensa que poderá enriquecer a abordagem seguida nesta dissertação.

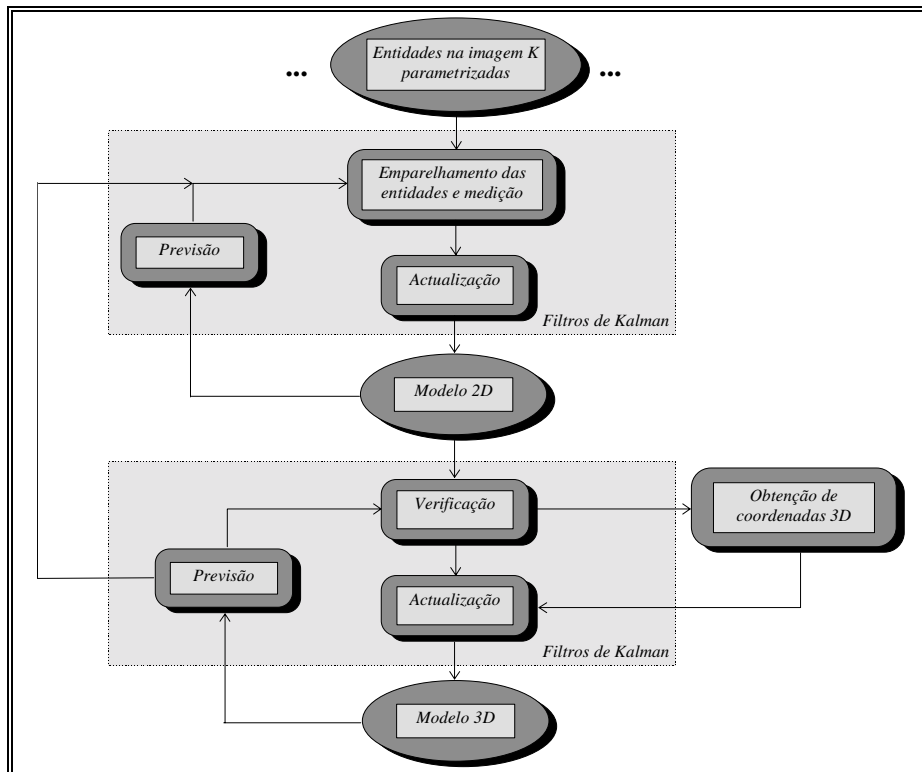


Fig. 1 - Uma abordagem possível para enriquecer o seguimento das várias entidades ao longo da sequência e conseqüente obtenção de informação tridimensional acerca da cena.

- *Criação de um sistema autónomo para obtenção de informação tridimensional a partir do movimento de uma câmara devidamente calibrada* - Uma solução para a obtenção de tal sistema passará por integrar todas as etapas envolvidas (determinação das entidades, seguimento, e obtenção de informação tridimensional). As implementações dessas etapas poderão ser coordenadas por um novo módulo que faça a gestão de toda a transição de informação entre tais etapas e da forma como as mesmas serão executadas.
- *Aumento da velocidade de processamento* - Uma solução para aumentar a velocidade de processamento, passará inevitavelmente por transferir certas etapas da metodologia usada de *software* para *hardware* dedicado, e por implementar versões paralelizadas dos processos, adaptados a sistemas com múltiplas unidades de processamento.

Bibliografia

[Batista, 1993] - *Jorge Batista, Jorge Dias, Helder Araújo, A. Traça de Almeida*
Monoplanar Camera Calibration for Off-the-Shelf TV Cameras and Lenses.
Iterative Multi-step Approach

5TH PORTUGUESE CONFERENCE ON PATTERN RECOGNITION 1993 - 53/62

[Canny, 1986] - *J. F. Canny*

A Computational Approach for Edge Detection

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE PAMI-8(6) (NOVEMBER, 1986) - 679/698

[Chapra, 1988] - *Steven C. Chapra, Raymond P. Canale*

Numerical Methods for Engineers

MCGRAW-HILL - 1988

[Checa, 1993] - *B. Gaii-Checa, R. Deriche, Th. Viéille, O. Faugeras*

Suivi de segments dans une séquence d'images monoculaire

Rapport de recherche N° 2113 - Décembre 1993

INRIA - INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATION

[Chen, 1989] - *J. S. Chen, G. Medioni*

Detection, Localization and Estimation of Edges

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE PAMI-11(2) (FEBRUARY, 1989) - 191/198

[Correia, 1995] - *Miguel F. P. Velhote Correia*

Análise de Movimento em Sequências de Imagens

Dissertação submetida ao Departamento de Engenharia Electrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto para satisfação parcial dos requisitos do Mestrado em Engenharia Electrotécnica e de Computadores - Fevereiro de 1995

[Crowley, 1992] - *James L. Crowley, Pactrick Stelmaszyk, Thomas Skordas, Pierre Puget*

Measurement and Integration of 3-D Structures by Tracking Edge Lines

INTERNATIONAL JOURNAL OF COMPUTER VISION 8:1, 29-52 (1992)

[Deriche, 1987] - *Rachid Deriche*

Using Canny's Criteria to derive a Recursively Implemented Optimal Edge Detector

INTERNATIONAL JOURNAL OF COMPUTER VISION 1(2) (1987) - 167/187

[Deriche, 1990] - *Rachid Deriche, Olivier Faugeras*

Tracking line segments

IMAGE AND VISION COMPUTING - VOL. 8 NO 4 (NOVEMBER 1990)

[Foley, 1991] - *Foley, vanDam, Feiner, Hughes*

Computer Graphics Principles and Practice

ADDISON WESLEY 12110 - 1991 SECOND EDITION

[Hall, 1993] - *Ernest L. Hall*

Fundamental principles of robot vision

SPIE VOL. 2056 INTELLIGENT ROBOTS AND COMPUTER VISION XII (1993) - 321/333

[Haralick, 1984] - *R. M. Haralick*

Digital Step Edges from Zero-crossings of Second Directional Derivatives

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE PAMI-6(1) (JANUARY, 1984) - 58/68

[Huang, 1988] - *J. S. Huang, D. H. Tseng*

Statistical Theory of Edge Detection

CVGIP COMPUTER VISION, GRAPHICS AND IMAGE PROCESSING 43 (1988) - 337/346

[Hussien, 1993] - *B. Hussien, B. Sridhar*

A Robust Line Extraction and Matching Algorithm

SPIE VOL. 2055 INTELLIGENT ROBOTS AND COMPUTER VISION XII (1993) - 369/380

[Lenz, 1988] - *Reimar K. Lenz, Roger Y. Tsai*

Techniques for Calibration of the Scale Factor and Image Center for High Accuracy 3-D Machine Vision Metrology

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, VOL. 10, NO. 5, SEPTEMBER 1988 - 713/720

[Leung, 1990] - *Maylor K. Leung, Yee-Hong Yang*

Dynamic Strip Algorithm in Curve Fitting

COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING 51 146 (1990) - 146/165

[Lim, 1990] - *Jae S. Lim*

Two-Dimensional Signal and Image Processing

PTR PRENTICE HALL - 1990

[Lønnestad, 1992] - *Tor Lønnestad, Otto Milvang*

XITE User's Manual

October 21, 1992

[Marr, 1980] - *D. Marr, E. Hildreth*

Theory of Edge Detection

PROCEEDINGS OF THE ROYAL SOCIETY OF LONDON B-107 (1980) - 187/217

[Marr, 1982] - *David Marr*

Vision

W. H. FREEMAN AND COMPANY - 1982

[Maybeck, 1979] - *Peter S. Maybeck*

Stochastic Models, Estimation, and Control

Volume I

Mathematics In Science and Engineering - Volume 141

ACADEMIC PRESS -1979

[Mendonça, 1990] - *Ana Maria Mendonça, Jorge Alves da Silva*

Comunicação Interna

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO - 1990

[Nalwa, 1986] - *V. S. Nalwa, T. O. Binford*

On detecting edges

IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE PAMI-8(5) (SEPTEMBER 1986) - 699/714

[Press, 1992] - *William H. Press, Saul A. Teukolsky, William T. Vetterling, Brian P. Flannery*
Numerical Recipes - The Art of Scientific Computing
CAMBRIDGE UNIVERSITY PRESS - 1992 SECOND EDITION

[Sarkar, 1991] - *S. Sarkar, K. L. Boyer*
On Optimal Infinite Impulse Response Edge Detection Filters
IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE PAMI-13(11)
(NOVEMBER, 1991) - 1154/1171

[Schalkoff, 1989] - *Robert J. Schalkoff*
Digital Image Processing and Computer Vision
JOHN WILEY & SONS, INC. - 1989

[Shen, 1992] - *J. Shen, S. Castan*
An Optimal Linear Operator for Step Edge Detection
CVGIP: GRAPHICAL MODELS AND IMAGE PROCESSING 54(2) (1992) - 112/133

[Silva, 1994] - *Jorge Alves da Silva*
Aquisição e Processamento de Informação Tridimensional
Dissertação de Doutoramento apresentada à Faculdade de Engenharia da Universidade do Porto,
Departamento de Engenharia Electrotécnica e de Computadores - 1994

[Tavares, 1995] - *João Manuel R. S. Tavares*
Algumas Ferramentas para Visão Tridimensional por Computador
Publicação inserida na dissertação submetida ao Departamento de Engenharia Electrotécnica e de
Computadores em 1995, para satisfação parcial dos requisitos do Mestrado em Engenharia
Electrotécnica e de Computadores.
Orientador: *A. Jorge Padilha*
FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO - 1995

[Torre, 1986] - *V. Torre and T. Poggio*
On Edge Detection
IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE PAMI-8(2) (MARCH,
1986) - 147/163

[Tsai, 1987] - *Roger Y. Tsai*
**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, AUGUST 1987 - 323/344

[Wu, 1988] - *J. J. Wu, R. E. Rink, T. M. Caelli and V. G. Gourishankar*
**Recovery of 3-D Location and Motion of a Rigid Object Through Camera Image (An Extended
Kalman Filter Approach)**
INTERNATIONAL JOURNAL OF COMPUTER VISION 3:3 (1988) - 773-394

ANEXO

TRANSFORMAÇÕES GEOMÉTRICAS EM 2D E EM 3D

Neste anexo são introduzidas as transformações geométricas básicas em 2D e 3D utilizadas em vários domínios como, por exemplo, em computação gráfica e em visão por computador. O anexo encontra-se dividido em quatro secções: na primeira, são apresentadas as transformações geométricas em 2D; na segunda, em 3D; na terceira, é referida de forma muito sucinta uma implementação, desenvolvida no âmbito desta dissertação, capaz de aplicar as diversas transformações a pontos em 2D e em 3D, assim como alguns exemplos de resultados obtidos; na quarta e última secção, são apresentadas algumas conclusões.

A utilização de coordenadas homogéneas é devidamente justificada, assim como a utilização de matrizes para a representação das diversas transformações geométricas.

1 - Transformações geométricas em 2D

Pontos no plano (x, y) podem ser transladados para novas posições, através da soma de quantidades de translação às coordenadas prévias dos pontos. Para cada ponto $P(x, y)$ a ser movido para uma nova posição $P'(x', y')$: t_x unidades paralelamente ao eixo x e t_y unidades paralelamente ao eixo y , pode-se escrever:

$$x' = x + t_x, \quad y' = y + t_y. \quad (\text{Eqs. 1a e 1b})$$

Ao definir-se os vectores coluna:

$$P = \begin{bmatrix} x \\ y \end{bmatrix}, \quad P' = \begin{bmatrix} x' \\ y' \end{bmatrix}, \quad T = \begin{bmatrix} t_x \\ t_y \end{bmatrix},$$

então as Eqs. 1a e 1b, podem ser representadas mais compactamente como $P' = P + T$.

Pode-se transladar um objecto através da aplicação das Eqs. 1a e 1b a cada um dos seus pontos. Contudo, como cada segmento de recta de um objecto é formado por um elevado número de pontos, este processo consumiria demasiado tempo. Felizmente, em certas aplicações, nomeadamente em aplicações de computação gráfica, pode-se transladar todos os pontos de uma recta através da translação dos seus pontos extremos, e desenhando a recta entre estes dois pontos; este procedimento é também utilizado para o caso de rotação, escalonamento, etc. Na Fig. 1, é apresentado o resultado da aplicação da translação $(3, -4)$ a um quadrado.

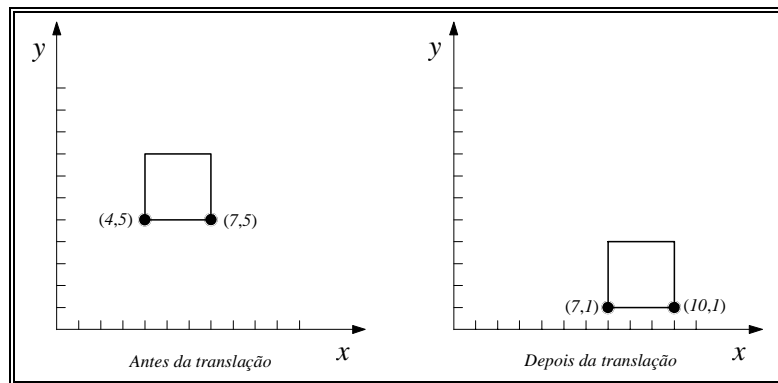


Fig. 1 - Translação de um quadrado.

Pontos podem ser escalonados ao longo do eixo x , por s_x , e ao longo do eixo y , por s_y , em novos pontos pelas multiplicações:

$$x' = s_x * x, \quad y' = s_y * y. \quad (\text{Eqs. 2a e 2b})$$

A representação matricial desta transformação geométrica é:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix} \text{ ou } P' = S * P, \text{ onde } S \text{ é a matriz escalonamento.}$$

Na Fig. 2, um quadrado é escalonado $1/2$ segundo o eixo x e $1/4$ segundo o eixo y . Note-se que o escalonamento é relativamente à origem. Técnicas para escalonamento em torno de um ponto arbitrário serão discutidas no ponto 1.2. No caso da Fig. 2, a figura geométrica reduz-se e aproxima-se da origem, pois, os factores de escalonamento utilizados foram menores do que 1; contudo, se os mesmos factores tivessem sido maiores do que 1, a figura geométrica seria ampliada e afastar-se-ia da origem. As proporções da figura geométrica foram também alteradas: um escalonamento diferencial, no qual $s_x \neq s_y$, foi utilizado. Com um escalonamento uniforme, no qual $s_x = s_y$, as proporções seriam mantidas.

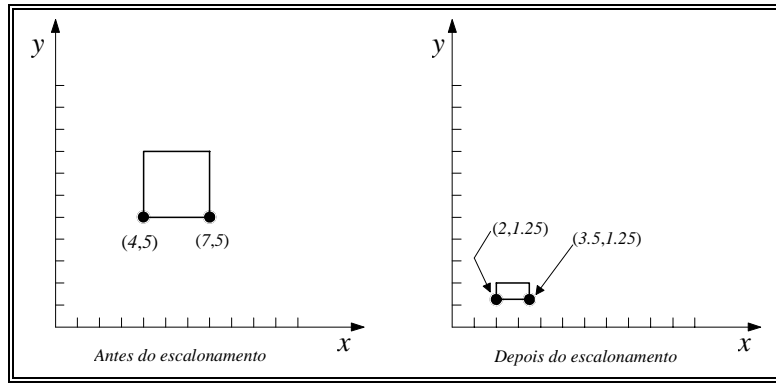


Fig. 2 - Escalonamento de um quadrado. Como o escalonamento não é uniforme o quadrado transformou-se num rectângulo.

Pontos podem ser rodados segundo um ângulo θ em torno da origem. Uma rotação é matematicamente definida por:

$$x' = x \cdot \cos \theta - y \cdot \sin \theta, \quad y' = x \cdot \sin \theta + y \cdot \cos \theta. \quad (\text{Eqs. 3a e 3b})$$

A representação matricial para a transformação geométrica rotação é:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix} \text{ ou } P' = R * P, \text{ onde } R \text{ é a matriz rotação.}$$

A Fig. 3 apresenta a rotação de um quadrado segundo um ângulo de 45° . Como no caso de escalonamento, a rotação é em torno da origem. O caso da rotação em torno de um ponto arbitrário será discutido no ponto 1.2.

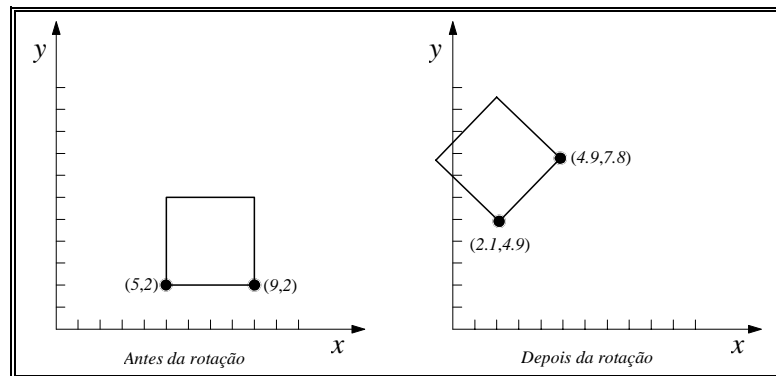


Fig. 3 - Rotação de um quadrado.

Ângulos de rotação positivos são medidos segundo o sentido contrário aos ponteiros do relógio, isto é, do eixo x para o eixo y . Para ângulos negativos (segundo o sentido dos ponteiros do relógio), as equivalências $\cos(-\theta) = \cos \theta$ e $\sin(-\theta) = -\sin \theta$ podem ser utilizadas para modificar as Eqs. 3a e 3b. As Eqs. 3a e 3b são derivadas facilmente da Fig. 4, na qual uma rotação segundo θ transforma $P(x, y)$ em $P'(x', y')$. Como a rotação é em torno da origem, as distâncias da origem a P e a P' , designadas por r na figura, são iguais. Utilizando simples trigonometria verifica-se que:

$$x = r \cdot \cos \phi, \quad y = r \cdot \sin \phi, \quad (\text{Eqs. 4a e 4b})$$

$$x' = r \cdot \cos(\theta + \phi) = r \cdot \cos \phi \cdot \cos \theta - r \cdot \sin \phi \cdot \sin \theta, \quad (\text{Eq. 5})$$

$$y' = r \cdot \sin(\theta + \phi) = r \cdot \cos \phi \cdot \sin \theta + r \cdot \sin \phi \cdot \cos \theta. \quad (\text{Eq. 6})$$

Substituindo as Eqs. 4a e 4b nas Eqs. 5 e 6, obtém-se as Eqs. 3a e 3b.

Uma outra transformação que um ponto $P(x, y)$ pode sofrer é distorção. Neste caso, a nova posição $P'(x', y')$ é matematicamente definida por:

$$x' = d_1 * x + d_2 * y, \quad y' = d_3 * x + d_4 * y. \quad (\text{Eqs. 7a e 7b})$$

A representação matricial da transformação geométrica distorção é:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} d_1 & d_2 \\ d_3 & d_4 \end{bmatrix} * \begin{bmatrix} x \\ y \end{bmatrix} \text{ ou } P' = D * P, \text{ onde } D \text{ representa a matriz distorção.}$$

Esta transformação é, tal como as transformações rotação e escalonamento, relativamente à origem. A Fig. 5, apresenta a aplicação da transformação distorção com $d_1 = 0.5$, $d_2 = 1$, $d_3 = 0.8$ e $d_4 = 0.9$, a um quadrado.

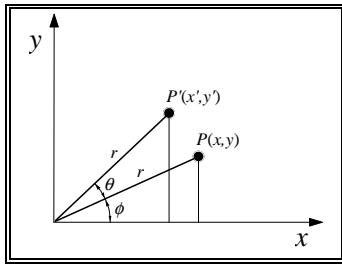


Fig. 4 - Derivação das equações de rotação.

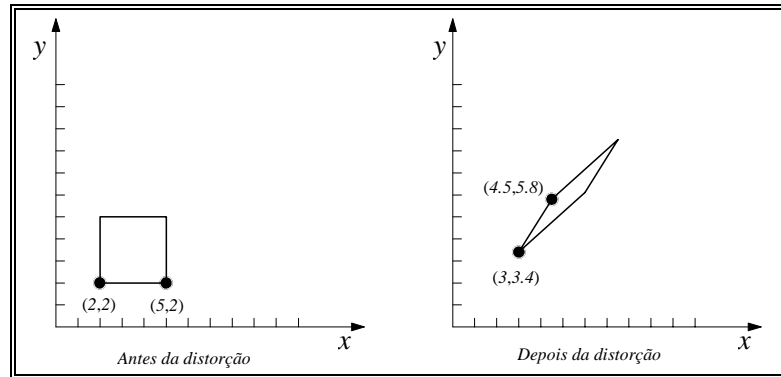


Fig. 5 - Distorção de um quadrado.

Saliente-se que a transformação distorção com $d_2 = d_3 = 0$ é equivalente à transformação escalonamento; com $d_2 = d_4 = 1$, $d_3 = 0$ e $d_1 \neq 0$, é equivalente à transformação distorção segundo o eixo x dado por d_2 , Fig. 6; com $d_1 = d_4 = 1$, $d_2 = 0$ e $d_3 \neq 0$, é equivalente à transformação distorção segundo o eixo y dado por d_3 , Fig. 7.

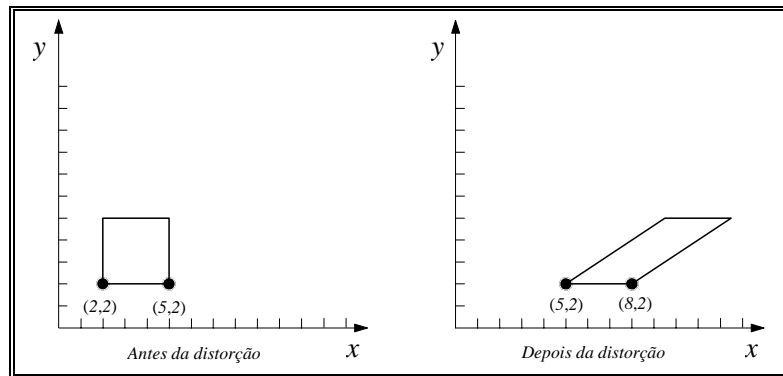


Fig. 6 - Distorção do quadrado segundo o eixo x .

1.1 - Representação matricial das transformações geométricas em 2D e coordenadas homogêneas

As representações matriciais para as transformações geométricas translação, escalonamento, rotação e distorção, são respectivamente:

$$P' = P + T, \quad P' = S * P, \quad P' = R * P, \quad P' = D * P.$$

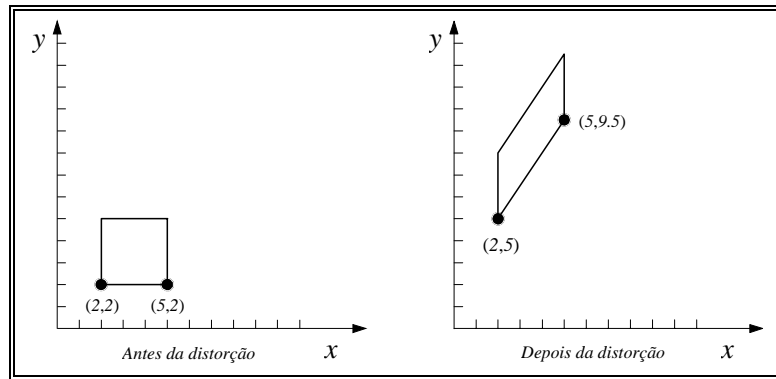


Fig. 7 - Distorção do quadrado segundo o eixo y .

Infelizmente, a translação envolve a operação adição, enquanto as restantes transformações envolvem a operação multiplicação. Seria vantajoso que todas as transformações envolvessem a mesma operação, de tal modo que pudessem ser facilmente combinadas. Isto consegue-se se os pontos passarem a ser representados em coordenadas homogêneas. Em coordenadas homogêneas, uma terceira coordenada é acrescentada ao ponto. Assim, cada ponto em vez de ser representado por um par de números (x, y) é representado por um triplo (x, y, W) . Ao mesmo tempo, diz-se que dois conjuntos de coordenadas homogêneas (x, y, W) e (x', y', W') representam o mesmo ponto, se e só se, um é múltiplo do outro. Deste modo, $(2, 3, 6)$ e $(4, 6, 12)$ representam o mesmo ponto; isto é, cada ponto em coordenadas homogêneas tem várias representações distintas. No entanto, pelo menos uma coordenada homogênea do triplo de representação tem necessariamente de ser diferente de zero: o triplo $(0, 0, 0)$ não é permitido. Se a coordenada W for diferente de zero, pode-se dividir o triplo por W ; assim os triplos (x, y, W) e $(x/W, y/W, 1)$ representam o mesmo ponto. Quando W é diferente de zero, esta divisão é normalmente efectuada e os números x/W e y/W são designados por coordenadas cartesianas do ponto homogêneo. Os pontos com $W=0$ são designados por pontos no infinito.

Tipicamente, triplos de coordenadas representam pontos no espaço 3D, mas neste caso são utilizados para representar pontos no espaço 2D. A justificação é a seguinte: considerando todos os triplos com a forma (tx, ty, tW) com $t \neq 0$ obtém-se uma recta no espaço 3D; deste modo, cada ponto homogêneo representa uma recta no espaço 3D; quando se homogeneiza um ponto (dividindo as suas coordenadas por W), obtém-se um ponto com a forma $(x, y, 1)$; assim, os pontos homogeneizados formam no espaço (x, y, W) um plano definido pela equação $W=1$. A Fig. 8, representa esta relação. Pontos no infinito não estão representados neste plano.

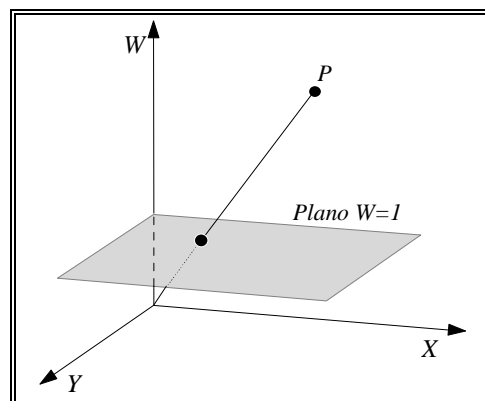


Fig. 8 - O espaço de coordenadas homogêneas (X, Y, W) com o plano $W=1$ e a projecção do ponto $P(X, Y, W)$ neste plano.

Devido aos pontos serem agora vectores com três linhas de elementos, as matrizes de transformação, que multiplicam um vector ponto para produzir outro vector ponto, têm de ser de

dimensão 3×3 . Na representação matricial 3×3 em coordenadas homogêneas, as Eqs. 1a e 1b de translação são:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \quad (\text{Eq. 8})$$

A equação anterior, pode ser representada de forma mais compacta como:

$$P' = T(t_x, t_y) * P, \text{ onde } T(t_x, t_y) = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}.$$

A translação é uma transformação aditiva, isto é, se um ponto P translada para P' segundo $T(t_{x_1}, t_{y_1})$, e depois para P'' segundo $T(t_{x_2}, t_{y_2})$, a sua translação pode ser representada como:

$$P' = T(t_{x_1}, t_{y_1}) * P, P'' = T(t_{x_2}, t_{y_2}) * P'. \quad (\text{Eqs. 9 e 10})$$

Substituindo a Eq. 9 na Eq. 10, obtém-se:

$$P'' = T(t_{x_2}, t_{y_2}) * [T(t_{x_1}, t_{y_1}) * P] = [T(t_{x_2}, t_{y_2}) * T(t_{x_1}, t_{y_1})] * P.$$

O produto de matrizes $T(t_{x_2}, t_{y_2}) * T(t_{x_1}, t_{y_1})$ - isto é, a matriz resultante da composição das transformações $T(t_{x_1}, t_{y_1})$ e $T(t_{x_2}, t_{y_2})$ - é:

$$\begin{bmatrix} 1 & 0 & t_{x_2} \\ 0 & 1 & t_{y_2} \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & t_{x_1} \\ 0 & 1 & t_{y_1} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_{x_1} + t_{x_2} \\ 0 & 1 & t_{y_1} + t_{y_2} \\ 0 & 0 & 1 \end{bmatrix},$$

sendo a confirmação que sucessivas translações são de facto aditivas.

De igual modo, as equações de escalonamento, Eqs. 2a e 2b, são representadas na forma matricial como:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \quad (\text{Eq. 11})$$

Definindo-se:

$$S(s_x, s_y) = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}, \text{ obtém-se } P' = S(s_x, s_y) * P.$$

Como no caso de sucessivas translações serem aditivas, sucessivos escalonamentos são multiplicativos. Isto é, dado:

$$P' = S(s_{x_1}, s_{y_1}) * P, P'' = S(s_{x_2}, s_{y_2}) * P', \quad (\text{Eqs. 12 e 13})$$

substituindo a Eq. 12 na Eq. 13, obtém-se:

$$P'' = S(s_{x_2}, s_{y_2}) * [S(s_{x_1}, s_{y_1}) * P] = [S(s_{x_2}, s_{y_2}) * S(s_{x_1}, s_{y_1})] * P.$$

O produto de matrizes $S(s_{x_2}, s_{y_2}) * S(s_{x_1}, s_{y_1})$ - isto é, a matriz resultante da composição das transformações $S(s_{x_1}, s_{y_1})$ e $S(s_{x_2}, s_{y_2})$ - é:

$$\begin{bmatrix} s_{x_2} & 0 & 0 \\ 0 & s_{y_2} & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} s_{x_1} & 0 & 0 \\ 0 & s_{y_1} & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s_{x_2} * s_{x_1} & 0 & 0 \\ 0 & s_{y_1} * s_{y_2} & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

sendo a confirmação que sucessivos escalonamentos são de facto multiplicativos.

De igual modo as equações de rotação, *Eqs. 3a e 3b*, podem ser representadas como:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \quad (\text{Eq. 14})$$

Definindo-se:

$$R(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}, \text{ obtém-se } P' = R(\theta) * P.$$

Sucessivas rotações são, como no caso de sucessivas translações, aditivas. Isto é, dado:

$$P' = R(\theta) * P, P'' = R(\phi) * P', \quad (\text{Eqs. 15 e 16})$$

substituindo a *Eq. 15* na *Eq. 16* obtém-se $P'' = R(\phi) * [R(\theta) * P] = [R(\phi) * R(\theta)] * P$.

O produto de matrizes $R(\phi) * R(\theta)$ - isto é, a matriz resultante da composição das transformações $R(\theta)$ e $R(\phi)$ - é:

$$\begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos(\phi + \theta) & -\sin(\phi + \theta) & 0 \\ \sin(\phi + \theta) & \cos(\phi + \theta) & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

sendo a confirmação que sucessivas rotações são de facto aditivas.

Finalmente, as equações de distorção, *Eqs. 7a e 7b*, podem ser representadas como:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} d_1 & d_2 & 0 \\ d_3 & d_4 & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}. \quad (\text{Eq. 17})$$

Definindo-se:

$$D(d_1, d_2, d_3, d_4) = \begin{bmatrix} d_1 & d_2 & 0 \\ d_3 & d_4 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \text{ obtém-se } P' = D(d_1, d_2, d_3, d_4) * P.$$

Todas as transformações apresentadas até este ponto, mantêm sempre o paralelismo entre os lados que constituem os objectos, sendo por isso, designadas por transformações afim.

As transformações translação e rotação são também designadas por transformações corpo-rígido, devido ao corpo ou objecto não ser destorcido de nenhum modo quando transformado. Isto é, são mantidos os ângulos, o paralelismo entre lados e os comprimentos.

1.2 - Composição de transformações geométricas em 2D

A composição é utilizada para combinar as matrizes representativas das transformações fundamentais R , S , T e D , de modo a produzir o resultado desejado. O objectivo básico da composição de transformações é aumentar a eficiência pela aplicação de uma transformação composta a um ponto, em vez da aplicação de uma série de transformações, uma após a outra.

Considere-se o caso da rotação de um objecto em torno de um ponto arbitrário P_I . Como a transformação rotação apresentada anteriormente é relativamente à origem, tem de converter-se o problema original, mais complexo, em três problemas separados, mais simples. Deste modo, para se proceder à rotação do objecto em questão em torno do ponto P_I , uma sequência de três transformações fundamentais torna-se necessária:

- 1) Translação de modo que P_I se localize na origem;
- 2) Rotação;
- 3) Translação de modo que P_I retorne à sua posição original.

Esta sequência de transformações fundamentais é ilustrada na Fig. 9, na qual o quadrado é rodado em torno do ponto $P_I(x_I, y_I)$. A primeira translação é de $(-x_I, -y_I)$, enquanto a última translação é, pelo inverso, de (x_I, y_I) . A composição de transformações resultante é:

$$T(x_I, y_I) * R(\theta) * T(-x_I, -y_I) = \begin{bmatrix} 1 & 0 & x_I \\ 0 & 1 & y_I \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -x_I \\ 0 & 1 & -y_I \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \cos\theta & -\sin\theta & x_I*(1 - \cos\theta) + y_I*\sin\theta \\ \sin\theta & \cos\theta & y_I*(1 - \cos\theta) - x_I*\sin\theta \\ 0 & 0 & 1 \end{bmatrix}$$

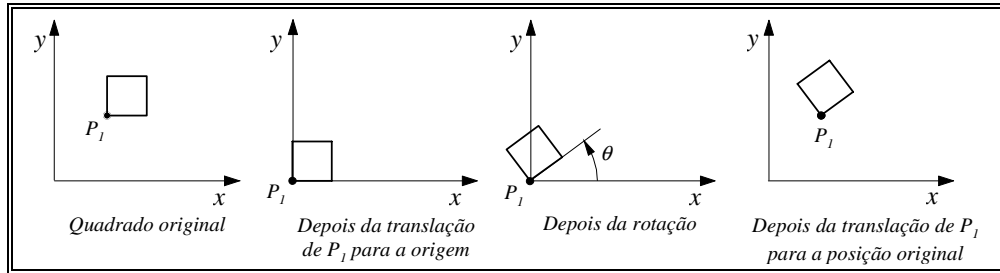


Fig. 9 - Rotação de um quadrado de um ângulo θ relativamente ao ponto P_I .

Um procedimento similar é utilizado para se proceder ao escalonamento de um objecto relativamente a um ponto arbitrário P_I , Fig. 10. A sequência pretendida consiste em: primeiro, translação de modo a P_I se localizar sobre a origem; seguidamente escalonamento, e por fim, translação de modo a P_I regressar à sua posição original. Neste caso, a composição de transformações é:

$$T(x_I, y_I) * S(s_x, s_y) * T(-x_I, -y_I) = \begin{bmatrix} 1 & 0 & x_I \\ 0 & 1 & y_I \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & -x_I \\ 0 & 1 & -y_I \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} s_x & 0 & x_1(1-s_x) \\ 0 & s_y & y_1(1-s_y) \\ 0 & 0 & 1 \end{bmatrix}.$$

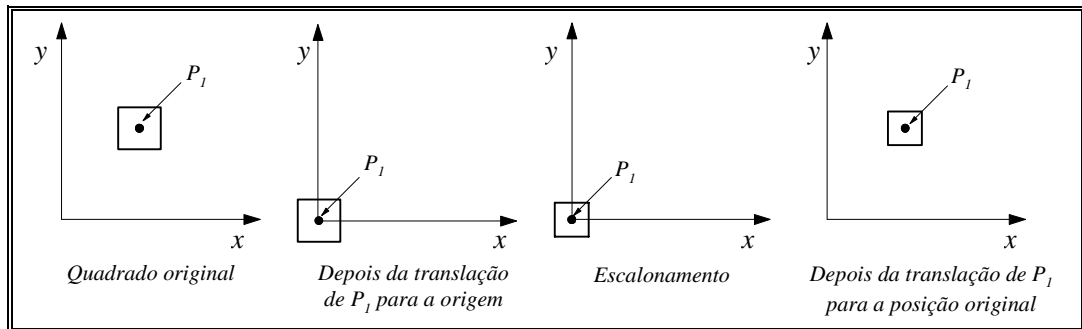


Fig. 10 - Escalonamento de um quadrado relativamente ao ponto P_1 .

Saliente-se que, no geral, a multiplicação de matrizes de transformações geométricas não é comutativa. Contudo, é facilmente demonstrado que, para os casos indicados na Tab. I, a comutatividade existe.

M_1 :	M_2 :
Translação	Translação
Escalação	Escalação
Rotação	Rotação
Escalação uniforme ($s_x = s_y$)	Rotação
<i>Nestes casos, não é necessário ter atenção na ordem da manipulação das respectivas matrizes.</i>	

Tab. I - Casos para os quais existe comutatividade na multiplicação das respectivas matrizes.

2 - Representação matricial de transformações geométricas em 3D

Tal como as transformações geométricas em 2D podem ser representadas por matrizes 3×3 , utilizando coordenadas homogéneas, as transformações geométricas em 3D podem ser representadas por matrizes 4×4 , desde que também se utilizem coordenadas homogéneas para a representação de pontos no espaço 3D. Desta forma, em vez de se representar um ponto por um triplo (x, y, z) , representa-se por um quádruplo (x, y, z, W) ; dois destes quádruplos representam o mesmo ponto se e só se um é múltiplo diferente de zero do outro: o quádruplo $(0, 0, 0, 0)$ não é permitido. Como em 2D, uma representação *standard* de um ponto (x, y, z, W) , com $W \neq 0$, é dada por $(x/W, y/W, z/W, 1)$; esta operação é designada como anteriormente por homogeneização. Também os pontos para os quais W é igual a zero são designados por pontos no infinito. Existe também uma interpretação geométrica: cada ponto no espaço 3D é representado por uma recta que passa pela origem do espaço 4D e a representação homogeneizada destes pontos forma um subespaço 3D do espaço 4D definido pela equação única $W = 1$.

O sistema de coordenadas 3D utilizado neste texto é o “da mão direita”, como representado na Fig. 11. Por convenção, neste sistema, rotações positivas são tais que: quando se observa a partir da parte positiva de um eixo para a sua origem, uma rotação de 90° no sentido contrário ao dos ponteiros do relógio transformará a parte positiva de um segundo eixo na parte positiva do terceiro eixo. A Tab. II indica esta convenção. Estas direcções positivas estão também indicadas na Fig. 11.

<i>Eixo de rotação:</i>	<i>Direção de rotação positiva:</i>
<i>x</i>	de <i>y</i> para <i>z</i>
<i>y</i>	de <i>z</i> para <i>x</i>
<i>z</i>	de <i>x</i> para <i>y</i>

Tab. II - Direção de rotação positiva para o sistema de coordenadas 3D “da mão direita”.

Existem textos que seguem outras convenções, nomeadamente a convenção da “mão esquerda”. Neste caso, Fig. 12, rotações positivas são definidas no sentido dos ponteiros do relógio quando se observa da parte positiva de um eixo para a origem.

Estas definições de rotações positivas permitem que as matrizes apresentadas sejam utilizadas quer para sistemas da “mão direita” quer para sistemas da “mão esquerda”. Conversões entre estes dois sistemas são discutidas no ponto 2.2.

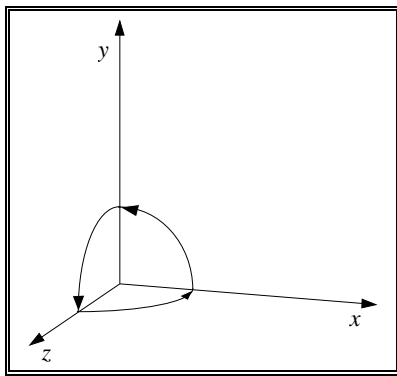


Fig. 11 - O sistema de coordenadas da “mão direita”.

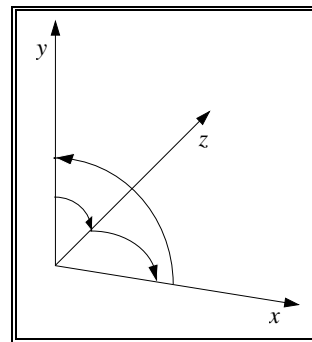


Fig. 12 - O sistema de coordenadas da “mão esquerda”.

A translação no caso 3D, é uma simples extensão do caso 2D:

$$T(t_x, t_y, t_z) = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{Eq. 18}$$

Isto é, $T(t_x, t_y, t_z) * [x \ y \ z \ 1]^T = [x+t_x \ y+t_y \ z+t_z \ 1]^T$.

Escalação é do mesmo modo uma extensão do caso 2D:

$$S(s_x, s_y, s_z) = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{Eq. 19}$$

Isto é, $S(s_x, s_y, s_z) * [x \ y \ z \ 1]^T = [x*s_x \ y*s_y \ z*s_z \ 1]^T$.

A rotação em 2D, Eqs. 3a e 3b, é apenas a rotação 3D em torno do eixo z:

$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{Eq. 20}$$

A matriz de rotação 3D em torno do eixo x é:

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (Eq. 21)$$

A matriz de rotação 3D em torno do eixo y é:

$$R_y(\theta) = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (Eq. 22)$$

Todas estas matrizes de transformação têm inversas. A inversa da matriz de translação T é obtida negando t_x , t_y e t_z ; a inversa da matriz de escalonamento S é obtida substituindo s_x , s_y e s_z , pelos respectivos inversos; as inversas das matrizes de rotação são obtidas através da negação do ângulo de rotação. No caso das matrizes de rotação, devido a estas serem ortogonais, para se obter as respectivas matrizes inversas basta realizar a transposição das mesmas.

Qualquer número de matrizes de rotação, escalonamento e translação, podem ser multiplicadas entre si. O resultado final terá sempre a forma:

$$M = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (Eq. 23)$$

Como no caso 2D, a submatriz 3×3 superior esquerda, corresponde à rotação e ao escalonamento associados, enquanto os três primeiros elementos da quarta coluna correspondem à translação. A eficiência computacional é aumentada executando a transformação de forma explícita como:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = R * \begin{bmatrix} x \\ y \\ z \end{bmatrix} + T,$$

onde R é a rotação e o escalonamento associados, e T é a translação.

A matriz distorção em 3D é a extensão simples da matriz de distorção em 2D, sendo:

$$D(d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8, d_9) = \begin{bmatrix} d_1 & d_2 & d_3 & 0 \\ d_4 & d_5 & d_6 & 0 \\ d_7 & d_8 & d_9 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (Eq. 24)$$

Isto é, aplicando a transformação $D(d_1, d_2, d_3, d_4, d_5, d_6, d_7, d_8, d_9)$ ao ponto $[x \ y \ z \ 1]^T$ obtém-se $[d_1*x + d_2*y + d_3*z \ d_4*x + d_5*y + d_6*z \ d_7*x + d_8*y + d_9*z \ 1]^T$.

Esta transformação com $d_1 = d_5 = d_9 = 1$ e $d_2 = d_4 = d_7 = d_8 = 0$, é equivalente à transformação distorção segundo os eixos x e y , dados por d_3 e d_6 , respectivamente. Neste caso a Eq. 24 passa ter a forma:

$$SH_{xy}(sh_x, sh_y) = \begin{bmatrix} 1 & 0 & sh_x & 0 \\ 0 & 1 & sh_y & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (Eq. 25)$$

Isto é, aplicando SH_{xy} ao ponto $[x \ y \ z \ 1]^T$, obtém-se $[x + sh_x * z \ y + sh_y * z \ z \ 1]^T$. Distorções ao longo dos eixos x e y têm efeitos similares.

Até esta altura, só se atendeu à transformação de pontos individuais. Transformam-se rectas, que tenham sido definidas por dois pontos, pela transformação dos seus pontos extremos. Os planos, se forem definidos por três pontos, podem ser tratados do mesmo modo; mas, usualmente, estes são definidos por uma equação do plano, e os coeficientes desta equação têm de ser transformados diferentemente. Pode-se também ter a necessidade de transformar a normal do plano. Seja um plano, representado por um vector coluna dos coeficientes da equação do plano $N = [A \ B \ C \ D]^T$. Então, um plano é definido por todos os pontos P tais que $N \bullet P = 0$, onde \bullet é o produto escalar e $P = [x \ y \ z \ 1]^T$. Este produto escalar dá origem à equação usual do plano $A * x + B * y + C * z + D = 0$, que pode ser representada como o produto do vector linha dos coeficientes da equação do plano, pelo vector coluna P : $N^T * P = 0$. Suponha-se que se transformam todos pontos P do plano por uma matriz M . Para manter $N^T * P = 0$ para os pontos transformados tem-se de transformar N por alguma (a ser determinada) matriz Q , dando origem à equação $(Q * N)^T * M * P = 0$. Esta expressão pode ser rescrita como $N^T * Q^T * M * P = 0$, utilizando-se a identidade $(Q * N)^T = N^T * Q^T$. Esta equação será válida se $Q^T * M$ for múltipla da matriz identidade. Se o múltiplo é 1 , tem-se $Q^T = M^{-1}$ ou $Q = (M^{-1})^T$. Isto significa que o vector coluna N' dos coeficientes para o plano transformado por M , é dado por:

$$N' = (M^{-1})^T * N. \quad (Eq. 26)$$

A matriz M^{-1} em geral não existe, devido ao determinante de M poder ser igual a zero. Isto acontecerá se M incluir uma projecção (ponto 2.3). No entanto é possível utilizar-se, em vez de $(M^{-1})^T$, a matriz dos cofactores de M , determinados na procura da inversa de M pela regra de Cramer.

Se apenas a normal ao plano é para ser transformada, e se M consiste apenas na composição das matrizes translação, rotação e escalonamento uniforme, então matematicamente ainda é mais simples. O vector N' da Eq. 26, pode ser simplificado para $N' = [A' \ B' \ C' \ 0]^T$.

2.1 - Composição de transformações em 3D

Neste ponto, a composição das transformações geométricas básicas em 3D será discutida, usando para o efeito um exemplo. Neste exemplo, o objectivo é transformar os segmentos de recta P_1P_2 e P_1P_3 , na Fig. 13, das suas posições iniciais (Fig. 13a) para as suas posições finais (Fig. 13b). Então, o ponto P_1 deverá ser transladado para a origem, P_1P_2 deverá estar sobre a parte positiva do eixo z e P_1P_3 deverá situar-se no plano (y, z) , na parte positiva do eixo y . O comprimento dos segmentos de recta não deverá ser alterado pela transformação.

Duas abordagens para encontrar a transformação desejada serão apresentadas. A primeira, consiste em compor as transformações primitivas T , R_x , R_y e R_z . Esta abordagem, apesar de ser algo fastidiosa, é facilmente ilustrada e o seu entendimento auxiliará a encontrar a compreensão do funcionamento das transformações. A segunda abordagem consiste em utilizar as propriedades das matrizes ortogonais e explica-se mais rapidamente; no entanto, é mais abstracta.

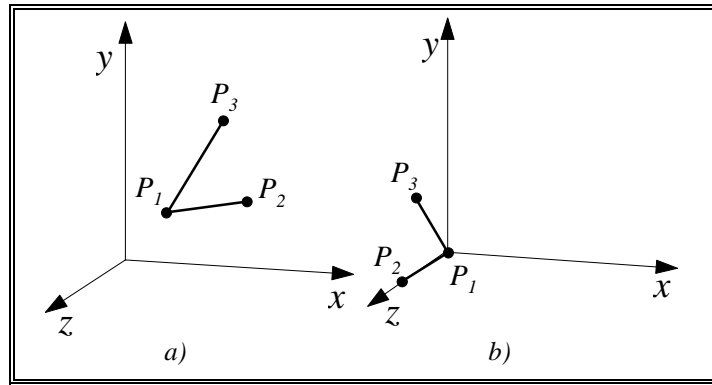


Fig. 13 - Transformação desejada: P_1 , P_2 e P_3 , das suas posições iniciais (a) para as suas posições finais (b).

Ao utilizar-se transformações primitivas divide-se o problema original, mais complexo, em subproblemas mais simples. Neste caso, a transformação pretendida pode ser realizada em quatro passos:

- 1) Translação de P_1 para a origem;
- 2) Rotação em torno do eixo y , de tal modo que P_1P_2 se situe no plano (y, z) ;
- 3) Rotação em torno do eixo x , de tal modo que P_1P_2 coincida com o eixo z ;
- 4) Rotação em torno do eixo z , de tal modo que P_1P_3 se situe no plano (y, z) .

Passo 1: Translação de P_1 para a origem

Considerando-se $P_1 = [x_1 \ y_1 \ z_1 \ 1]^T$, $P_2 = [x_2 \ y_2 \ z_2 \ 1]^T$ e $P_3 = [x_3 \ y_3 \ z_3 \ 1]^T$, a translação pretendida é:

$$T(-x_1, -y_1, -z_1) = \begin{bmatrix} 1 & 0 & 0 & -x_1 \\ 0 & 1 & 0 & -y_1 \\ 0 & 0 & 1 & -z_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Aplicando T a P_1 , P_2 e P_3 , obtém-se:

$$P'_1 = T(-x_1, -y_1, -z_1) * P_1 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, P'_2 = T(-x_1, -y_1, -z_1) * P_2 = \begin{bmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \\ 1 \end{bmatrix},$$

$$P'_3 = T(-x_1, -y_1, -z_1) * P_3 = \begin{bmatrix} x_3 - x_1 \\ y_3 - y_1 \\ z_3 - z_1 \\ 1 \end{bmatrix}$$

Passo 2: Rotação em torno do eixo y

A Fig. 14 apresenta P_1P_2 depois do passo 1 assim como a sua projecção no plano (x, y) . O ângulo da rotação pretendida é $-(90^\circ - \theta) = \theta - 90^\circ$. Então:

$$\cos(\theta - 90^\circ) = \sin\theta = \frac{z_2'}{D_1} = \frac{z_2 - z_1}{D_1}, \quad \sin(\theta - 90^\circ) = -\cos\theta = \frac{x_2'}{D_1} = \frac{x_2 - x_1}{D_1},$$

$$\text{onde } D_1 = \sqrt{(z_2')^2 + (x_2')^2} = \sqrt{(z_2 - z_1)^2 + (x_2 - x_1)^2}.$$

Quando estes valores são substituídos na Eq. 22 obtém-se $P_2'' = R_y(\theta - 90^\circ) * P_2' = \begin{bmatrix} 0 & y_2 - y_1 & D_1 & 1 \end{bmatrix}$. Como se esperava, a componente x de P_2'' é igual a zero e a componente z é o comprimento D_1 .

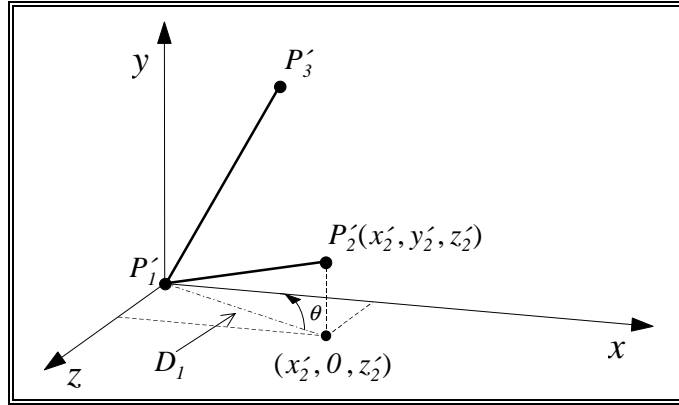


Fig. 14 - Rotação em torno do eixo y : a projecção de $P_1'P_2'$, que tem comprimento D_1 , é rodada de forma a situar-se no plano (y, z) ; o ângulo θ representa a direcção positiva da rotação em torno do eixo y ; o ângulo de rotação utilizado é $-(90^\circ - \theta)$.

Passo 3: Rotação em torno do eixo x

A Fig. 15 apresenta P_1P_2 depois do passo 2. O ângulo da rotação pretendida é ϕ , para o qual:

$$\cos\phi = \frac{z_2''}{D_2}, \quad \sin\phi = \frac{y_2''}{D_2},$$

onde $D_2 = |P_1''P_2''|$ ou seja, o comprimento do segmento de recta $P_1''P_2''$. Mas o comprimento do segmento de recta $P_1''P_2''$ é o mesmo do segmento P_1P_2 - devido às transformações rotação e translação preservarem o comprimento; deste modo:

$$D_2 = |P_1''P_2''| = |P_1P_2| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}.$$

O resultado da rotação no passo 3 é:

$$P_2''' = R_x(\phi) * P_2'' = R_x(\phi) * R_y(\theta - 90^\circ) * P_2' = R_x(\phi) * R_y(\theta - 90^\circ) * T * P_2 = \begin{bmatrix} 0 & 0 & |P_1P_2| & 1 \end{bmatrix}.$$

Isto é, P_1P_2 situa-se agora na parte positiva do eixo z .

Passo 4: Rotação em torno do eixo z

A Fig. 16 apresenta P_1P_2 e P_1P_3 após o passo 3 com P_2''' sobre o eixo z e P_3''' na posição:

$$P_3''' = \begin{bmatrix} x_3''' & y_3''' & z_3''' & 1 \end{bmatrix} = R_x(\phi) * R_y(\theta - 90^\circ) * T(-x_1, -y_1, -z_1) * P_3.$$

A rotação pretendida é segundo o ângulo positivo α , com $\cos\alpha = \frac{y_3'''}{D_3}$, $\sin\alpha = \frac{x_3'''}{D_3}$, $D_3 = \sqrt{x_3'''^2 + y_3'''^2}$.

Com o passo 4 obtém-se o resultado apresentado na Fig. 13b.

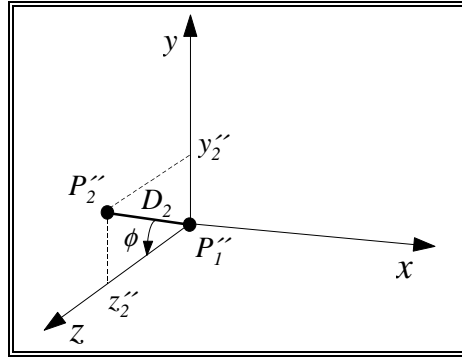


Fig. 15 - Rotação em torno do eixo x: $P_1''P_2''$ é rodado de forma a situar-se na parte positiva do eixo z segundo o ângulo positivo ϕ . D_2 é o comprimento do segmento de recta. O segmento de recta P_1P_3 não é representado, devido a não ser utilizado na determinação dos ângulos de rotação. Ambos os segmentos de recta são rodados por $R_x(\phi)$.

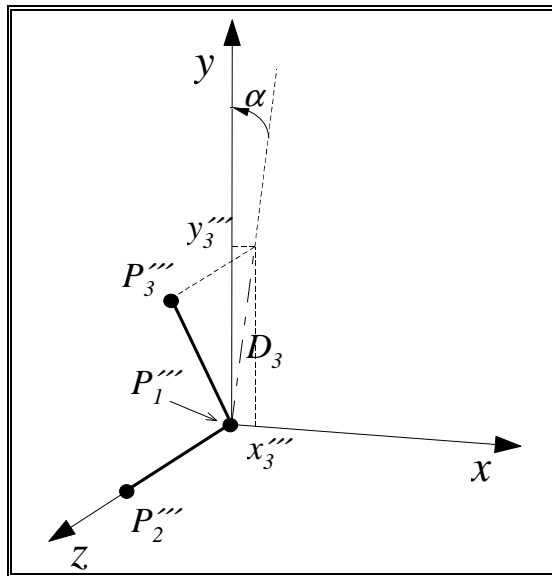


Fig. 16 - Rotação em torno do eixo z: a projecção de $P_1'''P_3'''$, cujo comprimento é D_3 , é rodada segundo o ângulo positivo α de forma a coincidir com o eixo y; passando a situar-se o segmento de recta no plano (y, z).

A composição de matrizes resultante é então:

$$R_z(\alpha) * R_x(\phi) * R_y(\theta - 90^\circ) * T(-x_1, -y_1, -z_1) = R * T, \quad (Eq. 27)$$

com $R = R_z(\alpha) * R_x(\phi) * R_y(\theta - 90^\circ)$.

A segunda abordagem para a obtenção da matriz R , consiste em utilizar as propriedades das matrizes ortogonais. Substituindo o segundo índice na Eq. 23 por x , y e z , por conveniências de notação, obtém-se:

$$R = \begin{bmatrix} r_{1x} & r_{2x} & r_{3x} \\ r_{1y} & r_{2y} & r_{3y} \\ r_{1z} & r_{2z} & r_{3z} \end{bmatrix}.$$

Devido a R_z ser um vector unitário ao longo de P_1P_2 que rodará para a parte positiva do eixo z tem-se:

$$R_z = [r_{1z} \quad r_{2z} \quad r_{3z}]^T = \frac{P_1P_2}{|P_1P_2|}.$$

O vector unitário R_x é perpendicular ao plano de P_1, P_2 e P_3 , e rodará para a parte positiva do eixo x : R_x deverá ser igual ao produto vectorial normalizado de dois vectores do plano. Isto é:

$$R_x = [r_{1x} \ r_{2x} \ r_{3x}]^T = \frac{P_1 P_3 \times P_1 P_2}{|P_1 P_3 \times P_1 P_2|}$$

Finalmente, $R_y = [r_{1y} \ r_{2y} \ r_{3y}]^T = R_z \times R_x$ rodará para a parte positiva do eixo y .

A matriz da composição resultante é dada por:

$$\begin{bmatrix} r_{1x} & r_{2x} & r_{3x} & 0 \\ r_{1y} & r_{2y} & r_{3y} & 0 \\ r_{1z} & r_{2z} & r_{3z} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * T(-x_1, -y_1, -z_1) = R * T, \text{ onde } R \text{ e } T \text{ são como na Eq. 27.}$$

A Fig. 17 apresenta os vectores individuais R_x, R_y e R_z .

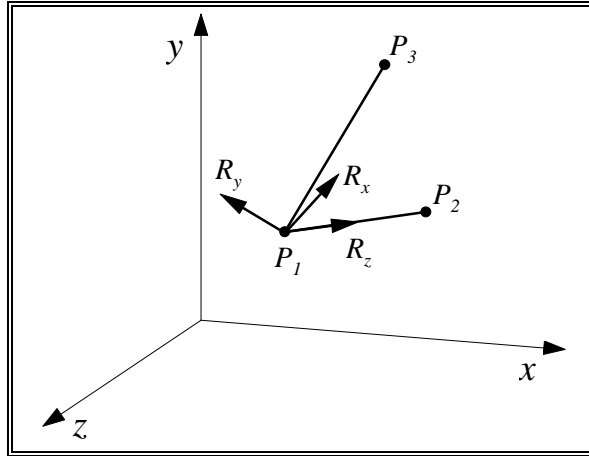


Fig. 17 - Os vectores unitários R_x, R_y e R_z , que são transformados em eixos principais.

2.2 - Transformações para mudança de sistema de coordenadas

Até este ponto, foram discutidas transformações de um conjunto de pontos pertencentes a um objecto noutro conjunto de pontos, quando ambos os conjuntos pertencem ao mesmo sistema de coordenadas. Com esta abordagem, o sistema de coordenadas permanece inalterado e o objecto é transformado relativamente à origem do sistema. Uma alternativa a esta abordagem, mas equivalente, é uma mudança do sistema de coordenadas. Esta abordagem é útil quando múltiplos objectos, cada qual expresso no seu sistema de coordenadas local, são combinados e se pretende expressá-los num sistema global.

Define-se $M_{i \leftarrow j}$ como a transformação que converte a representação de um ponto no sistema de coordenadas j , na sua representação no sistema i .

Definindo-se $P^{(i)}$ como a representação do ponto P no sistema de coordenadas i , $P^{(j)}$ como a representação do mesmo ponto no sistema de coordenadas j e $P^{(k)}$ como a representação do mesmo ponto no sistema de coordenadas k , então: $P^{(i)} = M_{i \leftarrow j} * P^{(j)}$, $P^{(j)} = M_{j \leftarrow k} * P^{(k)}$. Substituindo, obtém-se $P^{(i)} = M_{i \leftarrow j} * P^{(j)} = M_{i \leftarrow j} * M_{j \leftarrow k} * P^{(k)} = M_{i \leftarrow k} * P^{(k)}$; deste modo $M_{i \leftarrow k} = M_{i \leftarrow j} * M_{j \leftarrow k}$.

A Fig. 18 apresenta quatro sistemas de coordenadas diferentes. Verifica-se, pela observação da Fig. 18, que a transformação do sistema de coordenadas 2 para o sistema 1 é $M_{1 \leftarrow 2} = T(4, 2)$. Do mesmo modo, $M_{2 \leftarrow 3} = T(2, 3) * S(0.5, 0.5)$ e $M_{3 \leftarrow 4} = T(6.7, 1.8) * R(-45^\circ)$. Assim, $M_{1 \leftarrow 3} = M_{1 \leftarrow 2} * M_{2 \leftarrow 3} = T(4, 2) * T(2, 3) * S(0.5, 0.5)$. A figura apresenta também um ponto que é

$P^{(1)} = (10, 8)$, $P^{(2)} = (6, 6)$, $P^{(3)} = (8, 8)$ e $P^{(4)} = (4, 2)$, nos sistemas de coordenadas 1 a 4, respectivamente. É facilmente verificado que $P^{(i)} = M_{i \leftarrow j} * P^{(j)}$ para $1 \leq i, j \leq 4$.

Verifica-se também que $M_{i \leftarrow j} = M_{j \leftarrow i}^{-1}$. Assim, $M_{2 \leftarrow 1} = M_{1 \leftarrow 2}^{-1} = T(-4, -2)$. Porque $M_{1 \leftarrow 3} = M_{1 \leftarrow 2} * M_{2 \leftarrow 3}$, $M_{1 \leftarrow 3}^{-1} = M_{2 \leftarrow 3}^{-1} * M_{1 \leftarrow 2}^{-1} = M_{3 \leftarrow 2} * M_{2 \leftarrow 1}$.

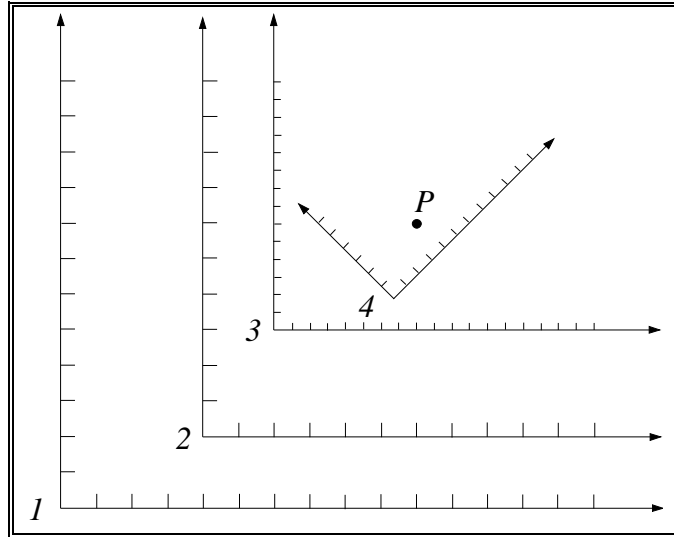


Fig. 18 - O ponto P e os sistemas de coordenadas 1, 2, 3 e 4.

No início desta secção, os sistemas de coordenadas da “mão esquerda” e da “mão direita” foram apresentados. A matriz que converte pontos representados num sistema de coordenadas para o outro, e vice-versa, é:

$$M_{D \leftarrow E} = M_{E \leftarrow D} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

A abordagem utilizada nos pontos anteriores - definir todos os objectos no sistema de coordenadas mundo e depois transformá-los para a localização desejada - implica uma noção irrealista; isto é, que todos os objectos são inicialmente definidos em cima uns dos outros no mesmo sistema de coordenadas mundo. É mais natural pensar que cada objecto esteja definido no seu sistema de coordenadas e depois sejam escalonados, rodados e transladados, pela redefinição das suas coordenadas no novo sistema de coordenadas mundo.

Considere-se o caso simples de translação do conjunto de pontos que definem o quadrado apresentado na Fig. 6 para a origem. Esta transformação é $T(-x_1, -y_1)$. Utilizando os dois sistemas de coordenadas apresentados na Fig. 19, verifica-se que a transformação que relaciona o sistema 1 com o sistema 2 - isto é, $M_{2 \leftarrow 1}$ - é igual a $T(x_1, y_1)$, que é apenas $T(-x_1, -y_1)^{-1}$.

Na realidade, a regra geral é a seguinte: a transformação a aplicar a um conjunto de pontos num único sistema de coordenadas, é apenas o inverso da transformação correspondente à mudança do sistema de coordenadas no qual os pontos estão representados. Esta relação pode ser observada na Fig. 20, que é derivada directamente da Fig. 21.

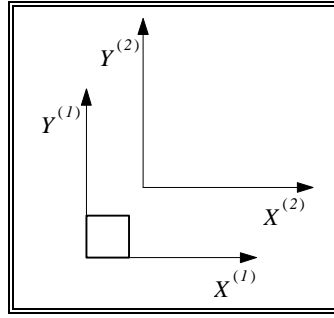


Fig. 19 - O quadrado e dois sistemas de coordenadas. Coordenadas dos pontos do quadrado podem ser representados em ambos os sistemas de coordenadas.

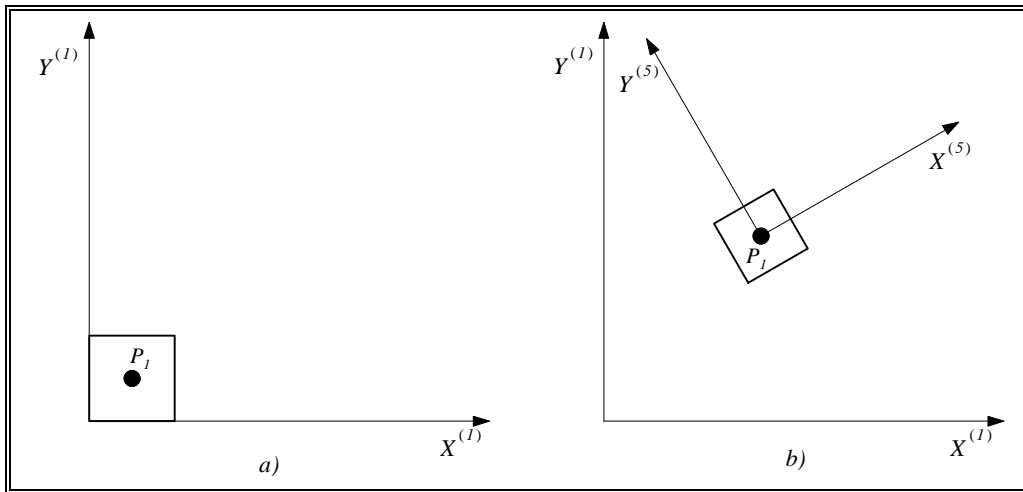


Fig. 20 - O quadrado original (a) no seu sistema de coordenadas e o quadrado transformado (b) no seu sistema relativamente ao sistema original.

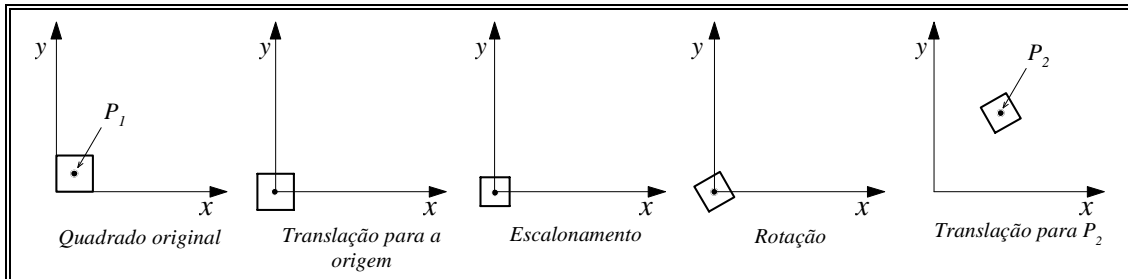


Fig. 21 - Rotação e escalaonamento do quadrado em torno do ponto P_1 e translação do quadrado transformado de modo a P_1 coincidir com P_2 .

A transformação para os pontos representados no único sistema de coordenadas, Fig. 21, é precisamente $T(x_2, y_2) * R(\theta) * S(s_x, s_y) * T(-x_1, -y_1)$.

Na Fig. 20, a transformação do sistema de coordenadas é precisamente:

$$M_{5 \leftarrow 1} = M_{5 \leftarrow 4} * M_{4 \leftarrow 3} * M_{3 \leftarrow 2} * M_{2 \leftarrow 1} = (T(x_2, y_2) * R(\theta) * S(s_x, s_y) * T(-x_1, -y_1))^{-1} \\ = T(x_1, y_1) * S(s_x^{-1}, s_y^{-1}) * R(-\theta) * T(-x_2, -y_2),$$

deste modo, $P^{(5)} = M_{5 \leftarrow 1} * P^{(1)} = T(x_1, y_1) * S(s_x^{-1}, s_y^{-1}) * R(-\theta) * T(-x_2, -y_2) * P^{(1)}$.

Uma questão importante, relacionada com a mudança do sistema de coordenadas, é a mudança da transformação. Suponha-se que $Q^{(j)}$ é uma transformação no sistema de coordenadas j . Pode por exemplo corresponder a uma composição de transformações discutida anteriormente. Suponha-se que

é desejado encontrar a transformação $Q^{(i)}$ no sistema de coordenadas i , que sendo aplicada aos pontos $P^{(i)}$ no sistema i , produzirá os mesmos resultados como quando $Q^{(j)}$ é aplicado aos pontos correspondentes $P^{(j)}$ no sistema j . Esta igualdade, é expressa por $Q^{(i)} * P^{(i)} = M_{i \leftarrow j} * Q^{(j)} * P^{(j)}$. Substituindo $P^{(i)} = M_{i \leftarrow j} * P^{(j)}$, a expressão anterior torna-se $Q^{(i)} * M_{i \leftarrow j} * P^{(j)} = M_{i \leftarrow j} * Q^{(j)} * P^{(j)}$. Simplificando, obtém-se $Q^{(i)} = M_{i \leftarrow j} * Q^{(j)} * M_{i \leftarrow j}^{-1}$.

2.3 - Outras transformações geométricas em 3D

Em 3D, existem outras transformações geométricas bastante utilizadas em computação gráfica e em visão por computador; nomeadamente, as transformações projecção e perspectiva.

No caso da transformação projecção, tudo se passa como se o observador projectasse a cena ou o objecto segundo uma direcção paralela à direcção de projecção (por exemplo, paralelamente ao eixo x , y ou z), sobre um plano que é perpendicular à respectiva direcção de projecção. Neste caso, diz-se que o centro de projecção se localiza no infinito.

Assim, a projecção segundo x é dada por:

$$M_x = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (\text{Eq. 28})$$

isto é, a projecção segundo x do ponto $P = [x \ y \ z \ 1]^T$ é $M_x * [x \ y \ z \ 1]^T = [0 \ y \ z \ 1]^T$.

A projecção segundo y é dada por:

$$M_y = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (\text{Eq. 29})$$

isto é, a projecção segundo y do ponto $P = [x \ y \ z \ 1]^T$ é $M_y * [x \ y \ z \ 1]^T = [x \ 0 \ z \ 1]^T$.

Finalmente, a projecção segundo z é dada por:

$$M_z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad (\text{Eq. 30})$$

isto é, a projecção segundo z do ponto $P = [x \ y \ z \ 1]^T$ é $M_z * [x \ y \ z \ 1]^T = [x \ y \ 0 \ 1]^T$.

Na Fig. 22, é representado um ponto P e as suas projecções segundo a direcção dos eixos x , y e z .

No caso da transformação perspectiva, o centro de projecção já não se localiza no infinito, mas a uma distância d do plano de projecção. Note-se que a transformação perspectiva é um caso especial da transformação paralela, para o qual o centro de projecção não se encontra no infinito. Existem duas variantes para esta transformação; numa, o plano de projecção encontra-se à frente do centro de projecção e na outra, o plano de projecção está atrás do centro de projecção. Neste texto, considera-se que, em ambas as variantes, o plano de projecção se situa em $z = 0$.

Na primeira variante, apresentada na Fig. 23, utiliza-se a semelhança de triângulos para escrever as relações:

$$\frac{x_p}{d} = \frac{x}{d+z}, \quad \frac{y_p}{d} = \frac{y}{d+z}.$$

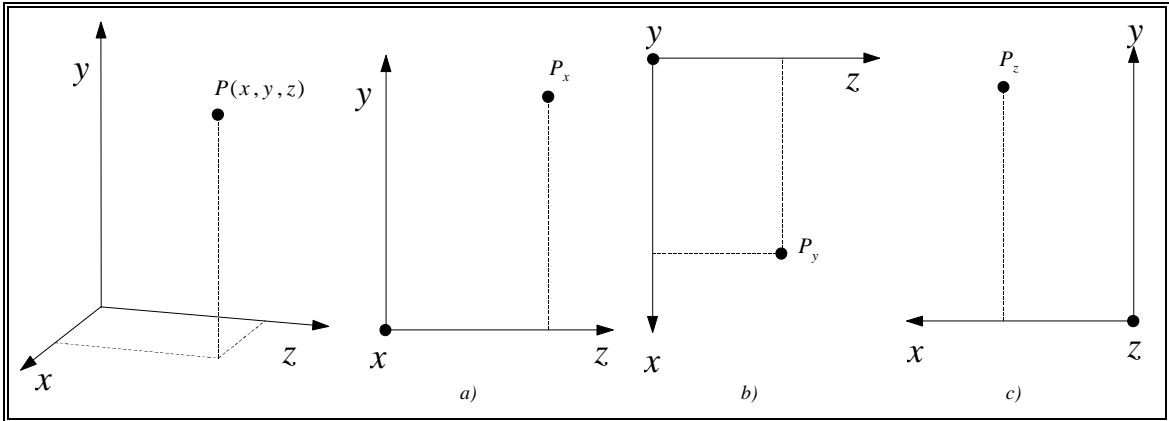


Fig. 22 - Um ponto P e as suas projecções segundo a direcção dos eixos: a) - x , b) - y e c) - z .

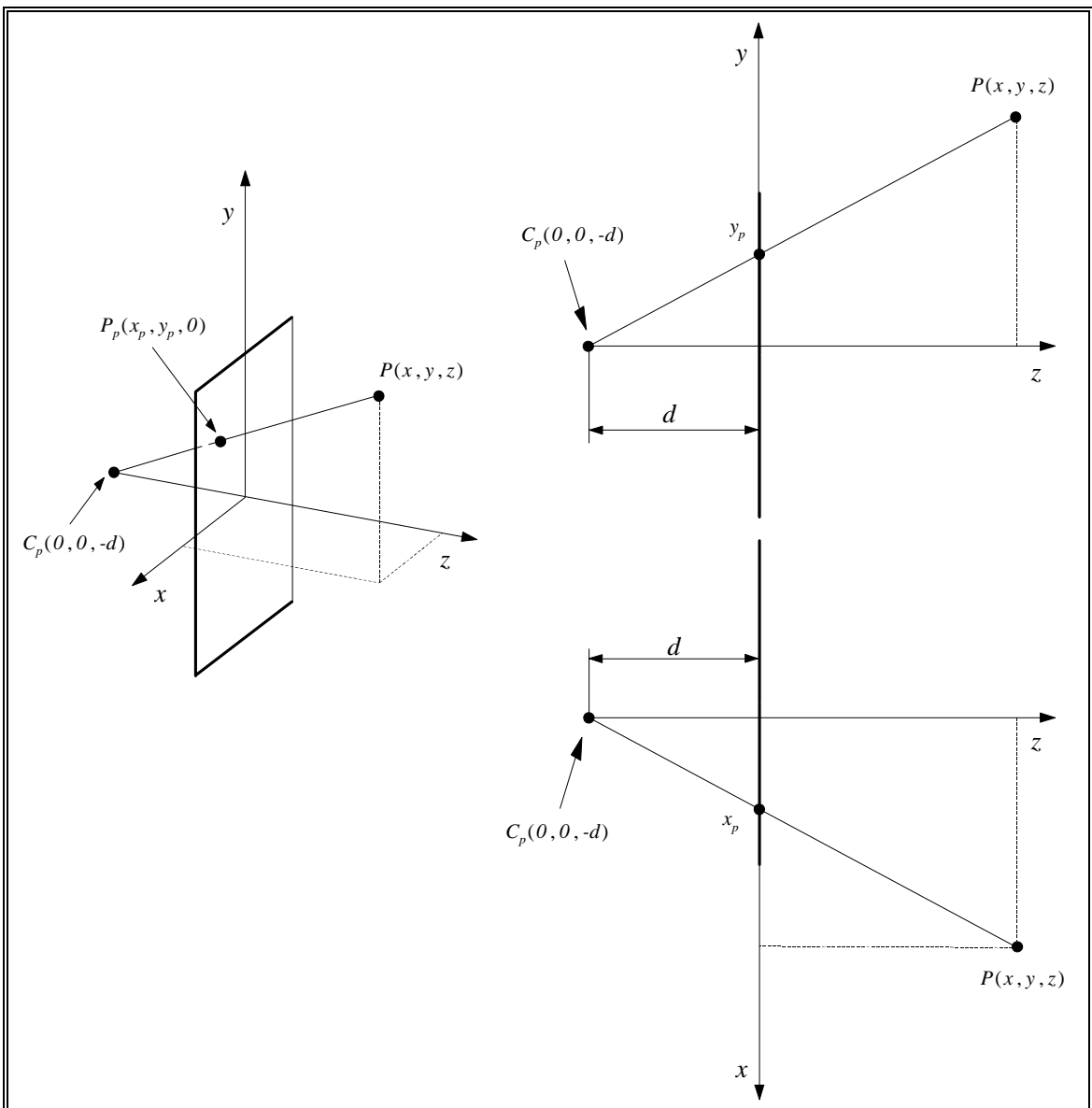


Fig. 23 - Projecção perspectiva quando o plano de projecção é anterior ao centro de projecção, C_p .

Multiplicando ambos os membros das relações anteriores por d , obtém-se:

$$x_p = \frac{x*d}{d+z} = \frac{x}{1+z/d}, \quad y_p = \frac{y*d}{d+z} = \frac{y}{1+z/d}. \quad (\text{Eqs. 31a e 31b})$$

A distância d é precisamente um factor de escala aplicado a x_p e a y_p . A divisão por z provoca que a projecção perspectiva de objectos mais distantes é mais pequena do que a dos mais próximos. A transformação dada pelas Eqs. 31a e 31b, pode ser representada mais compactamente pela matriz 4×4 :

$$M_{per} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/d & 1 \end{bmatrix}.$$

Deste modo, multiplicando o ponto $P = [x \ y \ z \ 1]^T$ pela matriz M_{per} resulta o ponto homogéneo $P = [X \ Y \ Z \ W]^T$:

$$\begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} = M_{per} * P = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/d & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix},$$

ou seja: $[X \ Y \ Z \ W]^T = [x \ y \ 0 \ z/d+1]^T$. Dividindo por W (que é igual a $z/d+1$) e desprezando-se a quarta coordenada para se obter o ponto 3D, obtém-se:

$$\left[\frac{X}{W}, \frac{Y}{W}, 0 \right] = [x_p, y_p, 0] = \left[\frac{x}{z/d+1}, \frac{y}{z/d+1}, 0 \right].$$

Note-se que, se d tender para o infinito, está-se no caso de projecção paralela segundo o eixo z .

Na segunda variante, em que o plano de projecção é anterior ao centro de projecção, para o qual a Fig. 24 representa a projecção perspectiva, utiliza-se a semelhança de triângulos para escrever:

$$-\frac{x_p}{d} = \frac{x}{d+z}, \quad -\frac{y_p}{d} = \frac{y}{d+z}.$$

Multiplicando ambos os membros por d , obtém-se:

$$x_p = -\frac{x*d}{d+z} = -\frac{x}{1+z/d}, \quad y_p = -\frac{y*d}{d+z} = -\frac{y}{1+z/d}. \quad (\text{Eqs. 32a e 32b})$$

A distância d é precisamente um factor de escala aplicado a x_p e a y_p . A divisão por z provoca que a projecção perspectiva de objectos mais distantes é mais pequena do que a dos mais próximos. Repare-se, que a projecção perspectiva, neste caso, apresenta um factor multiplicativo -1 nas coordenadas x_p e y_p , ocorrendo portanto, uma inversão do objecto no plano de projecção, sendo em tudo semelhante à transformação que ocorre no olho humano. A transformação dada pelas Eqs. 32a e 32b, pode ser representada mais compactamente pela matriz 4×4 :

$$M_{per} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/d & 1 \end{bmatrix}.$$

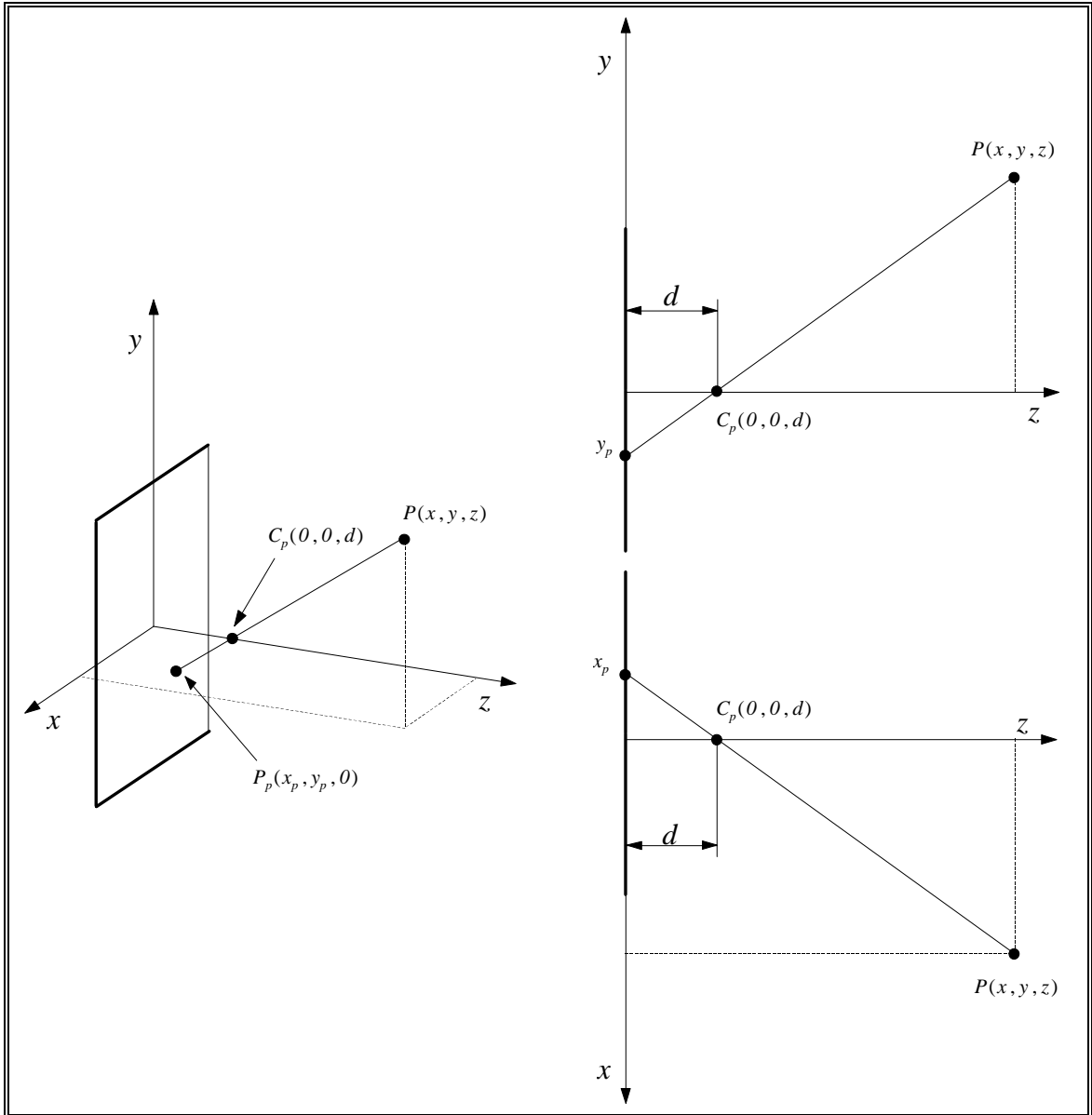


Fig. 24 - Projecção perspectiva quando o plano de projecção é posterior ao centro de projecção, C_p .

Deste modo, multiplicando o ponto $P = [x \ y \ z \ 1]^T$ pela matriz M_{per} resulta o ponto homogéneo $P = [X \ Y \ Z \ W]^T$:

$$\begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} = M_{per} * P = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/d & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix},$$

ou seja: $[X \ Y \ Z \ W]^T = [-x \ -y \ 0 \ z/d + 1]^T$. Dividindo por W (que é igual a $z/d + 1$) e desprezando-se a quarta coordenada para se obter o ponto 3D, obtém-se:

$$\left[\frac{X}{W}, \frac{Y}{W}, 0 \right] = [x_p, y_p, 0] = \left[-\frac{x}{z/d + 1}, -\frac{y}{z/d + 1}, 0 \right].$$

Note-se que, se d tender para o infinito, está-se no caso de projecção paralela segundo o eixo z , com um factor multiplicativo -1 nas coordenadas x_p e y_p .

Saliente-se que as matrizes anteriores podem apresentar outras formas, pois, como estão representadas utilizando coordenadas homogêneas, são equivalentes se multiplicadas por uma constante diferente de zero. Então, por exemplo:

$$M_{per} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1/d & 1 \end{bmatrix} \text{ é equivalente a } M'_{per} = \begin{bmatrix} d & 0 & 0 & 0 \\ 0 & d & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & d \end{bmatrix}, \text{ onde o factor multiplicativo é}$$

igual a d .

3 - Implementação*

Para apoiar as necessidades do trabalho descrito nesta dissertação, foi desenvolvida uma implementação¹ capaz de aplicar as diferentes transformações geométricas apresentadas neste texto, a pontos em 2D e em 3D. Nomeadamente, as transformações:

- ✓ rotação em torno do eixo x ;
- ✓ rotação em torno do eixo y ;
- ✓ rotação em torno do eixo z ;
- ✓ translação ao longo dos três eixos principais;
- ✓ escalonamento segundo os três eixos principais;
- ✓ distorção segundo os três eixos principais;
- ✓ projecção perspectiva quando o plano de projecção é anterior ao centro de projecção;
- ✓ projecção perspectiva quando o plano de projecção é posterior ao centro de projecção.

A referida implementação foi desenvolvida em linguagem C em ambiente $UNIX$ numa estação de trabalho $SunSparc$ e integra-se no ambiente de processamento de imagem $XITE$ ², do qual utiliza algumas funções e tipos de variáveis já previamente definidas.

Esta implementação é constituída pelos seguintes módulos:

- a) **principal**, responsável pela interface implementação/utilizador e gestor de toda a implementação;
- b) **rotationx**, responsável pela transformação rotação em torno do eixo x ;
- c) **rotationy**, responsável pela transformação rotação em torno do eixo y ;
- d) **rotationz**, responsável pela transformação rotação em torno do eixo z ;
- e) **translation**, responsável pela transformação translação ao longo dos três eixos principais;
- f) **scale**, responsável pela transformação escalonamento segundo os três eixos principais;
- g) **distortion**, responsável pela transformação distorção segundo os três eixos principais;
- h) **perspective1**, responsável pela transformação perspectiva quando o plano de projecção é anterior ao centro de projecção e tem coordenada z igual a zero;
- i) **perspective2**, responsável pela transformação perspectiva quando o plano de projecção é posterior ao centro de projecção e tem coordenada z igual a zero;

* Esta secção foi introduzida por razões de completude.

¹ Em [Tavares, 1995] esta implementação está descrita de forma mais detalhada.

² *X-based Image processing Tools an Environment*, [Lønnestad, 1992].

j) *multmatvec*, responsável pela multiplicação de uma matriz 4×4 por um vector coluna 4×1 .

Para usar esta implementação, o utilizador apenas necessita de introduzir as coordenadas do ponto pretendido, a designação da transformação geométrica desejada e, se pretender, o nome de um ficheiro de texto para escrita dos resultados obtidos.

No próximo ponto, são apresentados alguns resultados experimentais obtidos pela referida implementação.

3.1 - Alguns resultados experimentais

Na Fig. 25, apresentam-se alguns resultados experimentais obtidos pela referida implementação.

Results for geotransf with point P(10, 20, 30), for rotationx with teta=-1.24:
Point P'(10, 34.8694, -9.17179) with W=1.

Results for geotransf with point P(10, 20, 30), for rotationy with teta=-1.24:
Point P'(-25.1256, 20, 9.45784) with W=1.

Results for geotransf with point P(10, 20, 30), for rotationz with teta=-1.24:
Point P'(22.1636, -2.96191, 30) with W=1.

Results for geotransf with point P(10, 20, 30), for scale with sx=-1.24, sy=10, and sz=20:
Point P'(-12.4, 200, 600) with W=1.

Results for geotransf with point P(10, 20, 30), for distortion with d1=-1.24, d2=10, d3=20,
d4=30, d5=40, d6=50, d7=60, d8=70 and d9=80:
Point P'(787.6, 2600, 4400) with W=1.

Results for geotransf with point P(10, 20, 30), for perspective with d=-1.24:
Point P'(0.396927, 0.793854, 0) with W=25.1935.

Fig. 25 - Alguns resultados experimentais obtidos pela referida implementação.

4 - Conclusão

Neste anexo são apresentadas as diversas transformações em 2D e em 3D utilizadas em computação gráfica e em visão por computador. A utilização de coordenadas homogéneas e de matrizes para a representação das transformações é também discutida e justificada.

Uma implementação, capaz de aplicar estas transformações a pontos em 2D e em 3D, desenvolvida no âmbito desta dissertação, é referida de forma muito sucinta, assim como alguns exemplos de resultados experimentais. Esta implementação é, sem dúvida, de grande utilidade, pois no dia a dia e em vários domínios, surgem as mais diversas situações em que é necessário proceder à aplicação de alguma destas transformações.

Neste domínio recomendam-se as referências [Foley, 1991], [Schalkoff, 1989] e [Hall, 1993].

1 - Transformações geométricas em 2D	195
1.1 - Representação matricial das transformações geométricas em 2D e coordenadas homogéneas	197
1.2 - Composição de transformações geométricas em 2D	201
2 - Representação matricial de transformações geométricas em 3D	202
2.1 - Composição de transformações em 3D	205
2.2 - Transformações para mudança de sistema de coordenadas	209
2.3 - Outras transformações geométricas em 3D	212
3 - Implementação*	216
3.1 - Alguns resultados experimentais	217
4 - Conclusão	217

* Esta secção foi introduzida por razões de completude.

Obtenção de Estrutura Tridimensional a Partir de Movimento de Câmara

João Manuel Ribeiro da Silva Tavares

Errata

Página	Local	Onde se lê ...	Deve ler-se ...
24	1º parágrafo	De facto, ... equações lineares com cinco incógnitas, ...	De facto, ... equações lineares com sete incógnitas (ou cinco incógnitas, no caso do método equivalente para pontos de calibração coplanares, ver ponto I.4), ...
37	3º parágrafo	Os valores obtidos para ... e ψ , apresentam ...	Os valores obtidos para ... e ϕ , apresentam ...
39	Eq. 37	$[f] = \begin{bmatrix} \sqrt{x_i^2 + y_i^2} \\ z_i - f \end{bmatrix} = \begin{bmatrix} z_i \sqrt{X_{u_i}^2 + Y_{u_i}^2} \\ z_i f \end{bmatrix}$	$[f] = \begin{bmatrix} \sqrt{x_i^2 + y_i^2} \\ z_i - f \end{bmatrix} = \begin{bmatrix} z_i \sqrt{X_{u_i}^2 + Y_{u_i}^2} \\ z_i f \end{bmatrix}$
47	1º parágrafo	Trata-se do método ... um sistema de câmara e lente com distância focal ajustável ...	Trata-se do método ... um sistema de câmara e lente cuja distância é variável ...
50	1º parágrafo	Um dos métodos ... na(s) imagens(s) de calibração ...	Um dos métodos ... na(s) imagem(s) de calibração ...
126	Último parágrafo	A estimação da matriz de incerteza ... interior da qual a entidade cujas ...	A estimação da matriz de incerteza ... interior da qual o ponto médio da entidade cujas ...
138	No algoritmo apresentado	iii) Determinação das coordenadas dos pontos inicial e final da parte visível pela câmara, para a respectiva posição e orientação do segmento de recta.	iii) Determinação das coordenadas dos pontos inicial e final da parte do segmento de recta visível pela câmara, para a respectiva posição e orientação.
166	Fig. 12	Para os eixos x e y do referencial mundo adoptado os sentidos indicados correspondem às partes positivas, de cada eixo.	Para os eixos x e y do referencial mundo adoptado os sentidos indicados correspondem às partes negativas, em vez de corresponderem às partes positivas, de cada eixo.
202	2º parágrafo	Tal como as transformações ... em 3 D pode ser representadas ...	Tal como as transformações ... em 3D pode ser representadas ...
221	8ª referência	[Deriche, 1987] - Rachild Deriche	[Deriche, 1987] - Rachid Deriche