



Dissertação

Mestrado em Engenharia Eletrotécnica

Bin Picking – Reposicionamento de Peças usando um Manipulador

DAVID ALEXANDRE AREIA CRUZ

Leiria, Setembro de 2014



Dissertação

Mestrado em Engenharia Eletrotécnica

Energia e Automação

Bin Picking – Reposicionamento de Peças usando um Manipulador

DAVID ALEXANDRE AREIA CRUZ

Dissertação de Mestrado realizada sob a orientação do Doutor Hugo Filipe Costelha de Castro, e co-orientação do Doutor Carlos Fernando Couceiro de Sousa Neves, Professores da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria.

Leiria, Setembro de 2014

À Minha Família

Esta página foi intencionalmente deixada em branco.

Agradecimentos

A realização desta tese comportou um processo de enriquecimento pessoal, pela investigação, conhecimentos adquiridos e laços de amizade efetuados. Os meus agradecimentos dirigem-se especialmente a todos os que estiveram comigo neste percurso académico.

Em primeiro lugar quero prestar um reconhecimento à minha família, em particular: mãe, pai, mana, um enorme agradecimento por me apoiarem incondicionalmente, transmitindo-me força, ânimo e coragem perante as dificuldades.

Ao professor doutor Hugo Costelha, orientador desta tese, e ao professor doutor Carlos Neves, co-orientador desta tese, um obrigado pelo conhecimento transmitido, sugestões e apoio manifestado.

À empresa Eletrofer por disponibilizar um caso de estudo prático.

Às comunidades *Robotic Operating System* (ROS), *OpenCV*, *Point Cloud Library* (PCL) e *Boost* pela disponibilização de código e documentação, e ferramentas de visualização que permitem acelerar a colocação em prática de ideias, agilizando o teste e desenvolvimento de aplicações. Agradeço também à comunidade *Eigen* e *VTK* utilizados pela PCL para transformações em nuvens de pontos e visualização de resultados.

Manifesto o meu agradecimento a todos os meus amigos que de forma positiva me apoiaram na execução deste estudo.

Esta página foi intencionalmente deixada em branco.

Abstract

The need to insert raw-material in a production system is found across industry. When the objects arrive unordered, in a bin we are faced with a bin-picking problem. In this context, the manual handling of objects is associated with the repetition of similar movements during a high number of hours. As a result, musculoskeletal disorders such as tendinitis occur. The need for bin-picking applications arises from this problem. Furthermore, this type of applications, depending on the hardware and control system implemented, offers the potential to boost productivity and also to enable the inspection of all the material inserted in a production system. This dissertation involves the development and implementation of an automatic system for repositioning parts using low-cost sensing systems for a case study from a metallurgical company from Leiria region, in the center of Portugal. The parts to reposition, have a metallic aspect and are approximately flat. In addition to the challenges inherent to the shiny aspect of metallic parts, the challenges derived from their superposition and the thickness of the parts, which is of the order of magnitude of the capacity to measure depth, are addressed. In order to solve this problem, a method with a detection phase so as to pick up the and a phase to determine the pose of the parts were used. A detection method which uses the concept of searching a sub-region was developed for the picking up phase. To determine the pose the Iterative Closest Point algorithm was used. This algorithm, applicable in real-time, was tested successfully in a wide range of non-controlled illumination conditions (using the normal illumination of the room without any change), with the absence and presence of superposition.

Keywords: bin-picking, industrial robotics, logistics, metallic parts, low-cost sensing systems, Kinect.

Esta página foi intencionalmente deixada em branco.

Resumo

A necessidade de introduzir matéria-prima num sistema produtivo é transversal à indústria. No caso em que os objetos chegam de modo desorganizado, dentro de um recipiente estamos perante o problema "*bin-picking*". Neste contexto, a introdução manual de objetos encontra-se associada à execução do mesmo tipo de movimentos, de um modo repetitivo, durante um elevado número de horas. Como consequência surgem doenças de trabalho, destacando-se as tendinites. Surge, desta forma, a necessidade de uma aplicação de *bin-picking*. Este tipo de aplicações apresenta potencial, dependendo do *hardware* e controlo implementado, para aumentar a produtividade, assim como possibilitar a inspeção da totalidade do material inserido no sistema produtivo. A presente dissertação comporta o desenvolvimento e implementação de um sistema automático de reposicionamento de peças, utilizando *hardware* sensorial de baixo custo e partindo de um caso de estudo associado a uma empresa metalúrgica da região de Leiria. Os objetos a reposicionar, possuem aspeto metálico e são aproximadamente planos. Aos desafios inerentes ao brilho exibido pelas peças acresce a sobreposição entre as mesmas e o erro associado ao sensor de baixo custo e à espessura das peças ser da ordem de grandeza da capacidade de medição de profundidade. Para a resolução deste problema foi utilizado um método com uma etapa de deteção com vista a captura e uma etapa de determinação de pose das peças. Para a etapa de captura foi desenvolvido um método de deteção que utiliza o conceito de procura de uma sub-região. Para a etapa de determinação de pose foi utilizado o algoritmo *Iterative Closest Point*. O algoritmo, aplicável em tempo real, foi testado com sucesso numa vasta gama de condições de iluminação não controlada (utilizando iluminação normal de uma sala sem qualquer alteração), com a ausência e a presença de sobreposição.

Palavras-chave: *bin-picking*, robótica industrial, logística, objetos metálicos, sistemas sensoriais de baixo custo, Kinect.

Esta página foi intencionalmente deixada em branco.

Lista de Figuras

1.1	Exemplos de sobreposição de objetos.	2
2.1	Simplificação de forma e determinação de orientação de polígono plano convexo.	9
2.2	Ilustração de segmentação e alinhamento baseado no cálculo de descritores e <i>Iterative Closest Point</i> (ICP).	10
2.3	Ilustração de melhoria da estimativa de eixo para um objeto com simetria rotacional.	12
3.1	Aparência de caixa com peças utilizadas no caso de estudo de diferentes pontos de vista.	26
3.2	Irregularidades no aspeto de peças utilizadas no caso de estudo.	26
3.3	Diferentes peças utilizadas no caso de estudo.	27
3.4	Interligação dos componentes do sistema.	29
3.5	Disposição dos elementos do sistema associados à perceção e manipulação física das peças.	29
3.6	Padrão xadrez.	30
3.7	Fundo controlado na garra robótica.	30
3.8	Aspeto do padrão de infravermelhos produzido pela Kinect para peças de madeira e metal.	31
3.9	Posição das luminárias, janelas e manipulador robótico na sala	32
3.10	Sistema de <i>software</i> desenvolvido.	33
3.11	Referenciais do sistema.	38
3.12	Histograma de distância de pontos a plano para a calibração 22.	39
3.13	Diferença entre conversão de pontos com os parâmetros iniciais e parâmetros relativos à calibração.	40

3.14	Gradiente da diferença entre conversão de pontos com os parâmetros iniciais e parâmetros relativos à calibração.	40
3.15	Vetores com direção x e y.	42
3.16	Fluxograma que descreve as etapas de obtenção de segmentos.	44
3.17	Exemplo de uma região.	45
3.18	Ilustração das métricas utilizadas sobre uma peça.	47
3.19	Ilustração do conceito de uma sequência de segmentos obtidos incrementalmente, com a adição de um novo elemento.	48
3.20	Influência da inclinação do vetor normal de uma superfícies na estimativa de métricas numa imagem bidimensional.	49
3.21	Correspondência entre regiões.	55
3.22	Peça num ambiente de fundo controlado.	57
3.23	Vetores próprios e centroide numa peça.	58
3.24	Fluxograma geral do processo implementado.	59
3.25	Fluxograma da captura de peças.	61
3.26	Fluxograma da determinação de pose de uma peça.	62
4.1	Iluminação solar.	64
4.2	Iluminação artificial.	64
4.3	Aspetto de caixa vazia de acordo com a iluminação.	65
4.4	Aspetto de caixa com peças sem sobreposição de acordo com a iluminação.	67
4.5	Deteções em caixa com peças sem sobreposição.	68
4.6	Deteções em caixa com peças sem sobreposição ignorando falhas na estimativa da normal e consideração de folga nas medidas do lado do retângulo.	69
4.7	Deteção de peças com iluminação solar.	69
4.8	Deteção de falso positivo com iluminação difusa.	70
4.9	Aspetto de caixa com peças com sobreposição de acordo com a iluminação.	70
4.10	Deteções em caixa com peças com sobreposição.	71
4.11	Deteções de falsos positivos em caixa com peças com sobreposição	72
4.12	Deteção de falso positivo devido a reflexões de iluminação e correta deteção de zona com aspecto similar.	74
4.13	Variação da altura associada a uma possível captura de uma peça.	76

4.14	Variação da altura associada a uma possível captura de peças num mesmo plano.	76
4.15	Deteções de peças com vista ao cálculo de pose.	78
4.16	Cálculo de pose inicial.	79
4.17	Pose calculada.	79
4.18	Comparação entre dois tipos de peças diferentes.	79
4.19	Cálculo de pose para peça de dimensões similares.	80
A.1	Manipulador Industrial.	93
A.2	Controlador robótico utilizado.	94
A.3	Ligações entre controlador robótico, sensor fim de curso e eletroválvula.	94
A.4	Ferramenta utilizada.	95
A.5	Sensor de fim de curso.	95
B.1	Exemplo de projeção em câmaras <i>pinhole</i> , adaptado de http://www.ics.uci.edu/~majumder/vispercep/cameracalib.pdf	102
B.2	Gráficos de parâmetros referentes à $Matriz_{Câmara}$ das calibrações da câmara de infravermelhos.	104
B.4	Gráficos que evidenciam a estreita relação entre parâmetros obtidos nas calibrações da câmara de infravermelhos.	104
B.3	Gráficos de parâmetros referentes aos parâmetros de distorção das calibrações da câmara de infravermelhos.	105
B.5	Valores associados às métricas de distância entre duas conversões com parâmetros distintos de um mesmo ponto.	107
B.6	Valores associados às métricas de distância entre duas conversões com parâmetros distintos de um mesmo ponto, sem considerar o desvio médio.	108
B.7	Histograma de distância de pontos a plano para a calibração 22.	113
B.8	Diferença entre conversão de pontos com os parâmetros iniciais e parâmetros relativos à calibração.	113
B.9	Gradiente da diferença entre conversão de pontos com os parâmetros iniciais e parâmetros relativos à calibração.	113
C.1	Fluxograma descritivo do processo de comunicação por parte do computador.	118
C.2	Fluxograma descritivo do processo de comunicação por parte do controlador.	119

D.1	Exemplo de atualização de máxima distância a aresta mais próximo numa região.	132
D.2	Exemplo de diminuição de um dos lados do retângulo de menor área que engloba uma região com a adição de um ponto a uma região.	134
D.3	Exemplo de obtenção de um retângulo com dimensões iguais aos limites máximos para um retângulo de menor área que contém uma região. . .	135

Lista de Tabelas

4.1	Parâmetros utilizados no algoritmo de procura de peças.	73
B.1	Correlação entre parâmetros de distorção das calibrações da câmara de infravermelhos.	105
B.2	Correlação entre parâmetros de distorção das calibrações da câmara de cores.	108
B.3	Câmara de Cores - Parâmetros Distorção.	109
B.4	Câmara de Cores - Parâmetros da Matriz Câmara.	110
B.5	Câmara de cores - métricas convergência.	111
B.6	Câmara de cores - métricas convergência sem considerar o desvio médio.	112
B.7	Medições utilizadas para comparar calibrações	114
B.8	Parâmetros Extrínsecos.	115
B.9	Parâmetros Extrínsecos.	115
C.1	Valores de Eco.	117
C.2	Palavras Chave.	120
C.3	Serviços ROS.	123

Esta página foi intencionalmente deixada em branco.

Lista de Abreviaturas

- AAM** *Active Appearance Models*
- CAD** *Computer Aided Design*
- ESTG** *Escola Superior de Tecnologia e Gestão de Leiria*
- ICLK** *Inverse Compositional Lucas-Kanade*
- ICP** *Iterative Closest Point*
- IDE** *Integrated Development Environments*
- FPFH** *Fast Point Feature Histograms*
- FPGA** *Field Programmable Gate Arrays*
- GPM** *Geometric Pattern Matching*
- GPU** *Graphical Processing Unit*
- NIST** *National Institute of Standards and Technology*
- PCL** *Point Cloud Library*
- RAM** *Random Access Memory*
- RANSAC** *Random Sampling Consensus*
- RANSAM** *Random Sample Matching*
- ROS** *Robotic Operating System*
- SAC-IA** *Sample Consensus Initial Alignment*
- SDK** *Software Development Kit*
- SLR** *Single-Lens Reflex*
- SVD** *Single Value Decomposition*
- SVM** *Support Vector Machine*

Esta página foi intencionalmente deixada em branco.

Índice

Agradecimentos	iii
Abstract	v
Resumo	vii
Lista de Figuras	ix
Lista de Tabelas	xiii
Lista de Abreviaturas	xv
Índice	xviii
1 Introdução	1
1.1 Caso de Estudo	3
1.2 Organização do Documento	4
2 Revisão da Literatura	5
2.1 Perceção do Meio	6
2.1.1 Utilização de primitivas geométricas	7
2.1.2 Utilização de características ou aspeto visual	13
2.1.3 Segmentação e Determinação de Contornos	18
2.1.4 Detecção de pose	21
2.2 Adaptação através de aprendizagem	23
3 Metodologia	25
3.1 Caso de Estudo	26
3.2 Configuração e Constituição Física	28

xvii

3.2.1	Constituição	28
3.2.2	Especificações relativas à implementação	30
3.2.3	Iluminação	32
3.3	Desenvolvimento do software	33
3.3.1	Aplicação de processamento e controlo do processo	33
3.3.2	Aplicação do controlador	37
3.4	Calibração do Sistema	38
3.4.1	Calibração da Microsoft Kinect	39
3.4.2	Obtenção da correspondência entre os referenciais	40
3.5	Estimativa de Pose	43
3.5.1	Deteção Orientada à Captura de Peças	43
3.5.2	Deteção de Pose de uma Peça	56
3.6	Controlo do Processo	59
4	Testes e Resultados	63
4.1	Estimativa de pose com vista a captura de peças	63
4.2	Captura de peças	75
4.3	Determinação de pose com vista a recolocação de peças	76
5	Conclusões e trabalho futuro	81
	Bibliografia	83
A	Especificações dos Componentes do Sistema Físico	93
A.1	Sensor Microsoft Kinect para a Xbox 360	96
A.1.1	Características do dispositivo Microsoft Kinect	96
B	Calibração da Câmara <i>Microsoft Kinect</i>	101
C	Comunicação entre Controlador e Computador	117
D	Estimação de Pose	131

Capítulo 1

Introdução

Hoje em dia, cada vez mais, os robôs encontram-se presentes nos processos industriais. Sem sentir o mundo exterior, os robôs, apenas repetem tarefas pré-programadas, muito rígidas em relação ao ambiente. As capacidades sensoriais permitem estender a gama de funcionalidades dos robôs. Neste âmbito, a visão revela-se um tipo importante de sensor por conseguir recolher informação global sobre o ambiente sem o perturbar (na medida em que é um sensor sem contacto)[1]. Com vista a aumentar a flexibilidade e capacidade de adaptação dos sistemas robóticos, o controlo por realimentação visual foi introduzido e largamente investigado [2, 3]. A visão artificial é, atualmente, uma tecnologia bem estabelecida na indústria, sendo, cada vez mais utilizada [4]. A área primária de sucesso para a visão artificial são as aplicações de tapete rolante, sendo exemplos aplicações para guiar robôs ou controlo de qualidade [4]. Normalmente, o sistema de visão máquina é combinado com dispositivos mecânicos que asseguram que apenas um objeto é presente ao sistema de cada vez, eliminando a presença de sobreposições [4]. Muitas aplicações industriais estão a ser utilizadas com sucesso todos os dias empregando visão computacional e robôs, contudo, para esses sistemas os objetos têm de possuir uma forma (2d) simples ou estarem organizados de um modo estruturado [5].

O problema de *bin-picking* pertence a um campo de investigação substancial no domínio da robótica, contando já com avanços significativos [6, 7]. Do ponto de vista da automação a tarefa de *bin-picking* é muito desafiante, constituindo um tema de investigação em curso há décadas [8]. Parte do interesse científico de *bin-picking* advém da conjugação de vários assuntos específicos de robótica, tais como reconhecimento e localização de objetos, planeamento da forma de agarrar, planeamento de trajetória e desvio de obstáculos, numa tarefa comum [9, 10]. Por esta razão, *bin-picking* é, de

um modo consistente, escolhido para demonstrar a eficácia dos algoritmos mencionados [9], constituindo uma importante aplicação de exibição durante várias décadas [9]. Contudo, embora este problema seja já estudado há um número considerável de anos, existem ainda poucas soluções industriais [6, 11]. A maioria das soluções encontram-se ainda mais na fase de investigação que de tecnologia padrão pronta para aplicações industriais [6], constituindo uma omissão nas cadeias de fluxo de materiais automatizadas [8, 9], motivo pelo qual possui um elevado interesse económico [9]. A introdução manual de matéria-prima num sistema produtivo encontra-se associada à execução de movimentos repetitivos durante um número elevado de horas, o que origina doenças músculo-esqueléticas. É importante referir que estas aplicações abarcam motivações sociais e de bem-estar. O problema de *bin-picking*, onde os objetos têm formas 3D e se encontram aleatoriamente dispostos como mostra a Figura 1.1 ainda não foi resolvido de um modo geral, primeiramente devido a severos problemas de sobreposição [5], existindo uma multitude de aplicações dependentes de especificidades relativas ao utilizador, processo e produto, ao invés uma única aplicação *bin-picking* [12].



(a) Imagem do caso de estudo. (b) Imagem do artigo [5]. (c) Imagem do artigo [13].

Figura 1.1: Exemplos de sobreposição de objetos.

A perceção em ambiente fabril é, segundo o *National Institute of Standards and Technology* (NIST) [12], o fator que limita o desenvolvimento de tecnologias *bin-picking*, considerando-se resolvidos aspetos físicos como manipuladores robóticos ou garras. Para o caso em que o ambiente se encontra estruturado numa matriz, exigindo pouco esforço por parte do sistema de perceção, o problema de *bin-picking*, em vasta medida [12], é considerado como resolvido com um grau de maturidade relativamente elevado. Para o caso não estruturado, onde se concebem postura e forma aleatórias para os objetos, a tecnologia não atingiu ainda o ponto de maturidade desejado [12]. Contrastando com soluções específicas inflexíveis, o custo de uma solução robótica *bin-picking* não é conduzido pelo custo do robô [12]. É o custo da integração no sistema de produção a maior fatia desse valor, de modo mais particular questões de segurança e equipamento auxiliar específico ao processo (de modo a simplificar o processo de perceção), o que

influencia a relação custo-benefício e flexibilidade¹ dos sistemas [12]. Uma solução geral para o problema de *bin-picking* apresenta um elevado mercado potencial devido às inúmeras situações onde este é realizado manualmente [4, 6, 7].

1.1 Caso de Estudo

A Eletrofer é uma empresa metalúrgica da região de Leiria. Uma das suas atividades é a galvanização de peças metálicas. Neste processo, o material chega desorganizado em caixas, sendo inserido manualmente no sistema produtivo. Devido à execução do mesmo tipo de movimentos durante um elevado número de horas surgem doenças de trabalho com especial relevo para tendinites.

É neste contexto que surge a necessidade de uma aplicação de *bin-picking*. Este tipo de aplicações apresenta potencial, dependendo do *hardware* e controlo implementado, para aumentar a produtividade, assim como possibilitar a inspeção da totalidade do material inserido no sistema produtivo.

¹No sentido em que limita a adaptação do sistema a alterações (o equipamento auxiliar específico deve permitir o correto reconhecimento dos objetos e a consequente manipulação).

1.2 Organização do Documento

Este documento é composto por cinco capítulos:(1) Introdução, (2) Revisão da literatura, (3) Metodologia e (4) Conclusão e trabalho futuro. No primeiro capítulo faz-se um breve enquadramento do tema proposto, onde se aborda a sua relevância na atualidade, o seu estado de desenvolvimento, os objetivos e a organização estrutural do documento. O segundo capítulo contempla as abordagens que têm sido utilizadas ou propostas para a resolução de problemas de *bin-picking* a nível da percepção, seguindo-se a exposição de algoritmos não aplicados especificamente no âmbito de *bin-picking*, que apresentam potencial para contribuir para a resolução de partes do problema. Expõem-se algumas propostas para a manipulação das peças ou melhoramento dos resultados utilizando métodos de aprendizagem por parte do sistema. No terceiro capítulo revelam-se as peças utilizadas e os componentes que constituem o sistema implementado, seguindo-se a exposição da metodologia desenvolvida para a resolução do problema de *bin-picking*. Adicionalmente indica-se, através de argumentos dedutivos, o raciocínio que sustenta teoricamente o sistema de deteção que antecede a captura de peças. No quarto capítulo apresentam-se os resultados obtidos para várias condições de teste, discutindo-se observações pertinentes. O quinto capítulo compreende as conclusões extraídas do trabalho realizado e indica direções para desenvolvimento de trabalhos futuros.

Capítulo 2

Revisão da Literatura

A tarefa de *bin-picking* consiste na captura de objetos de uma caixa ou recipiente. Neste recipiente as peças podem encontrar-se desorganizadas e sobrepostas. Para peças muito pequenas podem-se utilizar painéis de vibração, com o seu movimento vibratório as peças tendem a afastar-se, resolvendo-se o problema da sobreposição. O alinhamento das peças é obtido através da geometria da panela e das peças. As painéis de vibração não se aplicam a peças com dimensões relevantes ou pesadas, sendo necessários sistemas de visão e aquisição de informação 3D para proceder à sua manipulação automática.

A tarefa de *bin-picking* pode, de um modo geral, ser dividida nas subtarefas de:

1. Perceção do meio, compreendendo a deteção da presença de um objeto no meio e a sua localização no espaço tridimensional, assim como a estimativa da sua orientação;
2. Decisão do objeto a capturar e da forma de o agarrar (de uma lista de hipóteses geradas);
3. Planeamento de trajetórias;
4. Atuação e aprendizagem, englobando o controlo do manipulador robótico de modo a este executar o planeamento estabelecido, a obtenção de dados relativos ao sucesso ou insucesso da abordagem utilizada e a utilização desses dados de forma a melhorar o sucesso de perceção, planeamento e atuação futuros.

Devido à perceção ser o fator que limita o desenvolvimento de tecnologias de *bin-picking*, sendo considerados resolvidos aspetos físicos como manipuladores robóticos ou

garras [12], neste capítulo destaca-se a etapa de percepção. São ainda expostos algoritmos referentes à manipulação e propostas para adaptação através de aprendizagem, utilizados no âmbito de *bin-picking* e considerados pertinentes.

2.1 Percepção do Meio

Numa aplicação de *bin-picking* a percepção do meio ocupa-se de varias funções: (1) determinar obstáculos ao movimento do robô; (2) identificar objetos de interesse; (3) determinar a postura de objetos de interesse com o rigor suficiente para ser agarrado e depositado no local desejado, segundo os critérios estabelecidos para a aplicação. A identificação e determinação da postura do objeto, a nível algorítmico, podem encontrar-se interligadas, não sendo efetuada distinção, de modo explícito, neste capítulo. Por se verificar a ausência de constrangimentos à movimentação do manipulador robótico na aplicação desenvolvida (ver capítulo 3.2), a função (1) não é abordada neste trabalho.

Existem várias abordagens para resolver o problema de deteção de pose de objetos e várias formas de as classificar [8]. Entre outras, encontram-se divisões entre abordagens: (1) baseadas em modelos ou em vistas dos objetos [4][14][15]; (2) baseadas em características¹ ou aparência de objetos [16]; (3) abordagens de baixo para cima² ou de cima para baixo³ [8].

Com base nestas divisões é possível referir, de modo geral, vantagens e desvantagens relativas à sua adoção. Na abordagem (1) a utilização de modelos permite descrições compactas, robustez face a oclusões e algum grau de invariância perante a iluminação e variações de pose [4], apresentando como desvantagens a necessidade da utilização de um processo de extração de características [4]. A possibilidade de lidar com objetos sem determinar explicitamente características, como linhas ou cantos, é apresentada como vantagem da abordagem baseada nas vistas, onde se exprime o ambiente e as vistas nos mesmos termos (valor de píxeis) [17]. As desvantagens apresentadas para métodos baseados em vistas englobam elevada dependência de iluminação, ponto de vista e pose do objeto [17]. Com base nestas divisões também é possível apontar fragilidades. Na abordagem (3) são apresentadas as abordagens de baixo para cima, tendo como pontos fracos a possibilidade de o ruído e as obstruções poderem influenciar a repetibilidade, enquanto que às abordagens de cima para baixo se associa falta de generalidade, sendo comum aplicar-se um valor limite ou procurar a melhor correspondência [8]. Estas

¹Tradução do termo inglês *feature*.

²Criando hipóteses baseadas em dados recolhidos.

³Verificando hipóteses com os dados recolhidos.

divisões de abordagens não são mutuamente exclusivas nem necessariamente complementares, constituindo hipóteses de classificação bibliográfica tal como outros modelos, como o proposto por [18, 19]. Nestes, onde se apresentam como componentes usados num sistema de reconhecimento de objetos típico: (1) extração de características, seguido de (2) agrupamento de características, sucedendo-se (3) geração de hipóteses relativas ao objeto e (4) uma fase de verificação. Com o advento da popularidade de abordagens baseadas em *machine learning* e Conjuntos de características⁴, a distinção entre as etapas anteriores esbateu-se, não sendo incomum encontrar uma abordagem ao reconhecimento de objetos que consista numa única fase de extração de características, seguida de aplicação encadeada de um ou mais classificadores⁵ [20].

Nas subsecções seguintes abordam-se em primeiro lugar algoritmos utilizados no âmbito de *bin-picking*. Estes algoritmos são divididos nas categorias: (1) utilização de primitivas geométricas e (2) utilização de características discriminatórias e aspeto visual. Segue-se a exposição de algoritmos de segmentação e determinação de contornos que, não estando diretamente associados a um processo de *bin-picking* na literatura, apresentam potencial para serem testados como parte integrante de um. Segue-se uma discussão sobre algoritmos de deteção de pose de peças. Finaliza-se apresentando algoritmos de reconhecimento de objetos cujo desenvolvimento não foi direcionado para aplicações de *bin-picking*.

2.1.1 Utilização de primitivas geométricas

A utilização de primitivas geométricas, como é abordada neste capítulo, significa aproveitamento de características geométricas de partes de um objeto para o seu reconhecimento. De um modo geral estes algoritmos possuem cinco fases:

1. Identificação de componentes geométricos modelados, onde são extraídas medições que caracterizem ou discriminem o objeto, de acordo com o(s) modelo(s) disponíveis numa base de dados;
2. Cálculo de métrica(s) ou informação associada ao grau de semelhança entre a deteção e o modelo;
3. Inserção de cada identificação efetuada numa lista ordenada de acordo com a(s) métrica(s) utilizada(s), ou rejeição do(s) objetos associado(s);

⁴Tradução do termo em inglês *bags-of-features*.

⁵Tradução do inglês *cascades of one or more classifiers*.

4. Estimativa da pose, caso a identificação da(s) primitiva(s) não implique ou compreenda essa operação;
5. Verificação da pose ou do resultado obtido (opcional).

Devido à sua aplicação prática em soluções de *bin-picking*, de seguida abordam-se, as primitivas: (1) superfícies planas e (2) primitivas cilíndricas ou elípticas.

Superfícies Planas

Na literatura recente foram utilizadas superfícies planas para reconhecimento numa aplicação de *bin-picking* [11],[21]. No algoritmo apresentado em [11] utiliza-se como fonte de informação um mapa de profundidade⁶ obtido com recurso ao método de luz estruturada. Na identificação de planos é efetuada através do cálculo do vetor normal para o segmento plano que descreve a vizinhança em cada ponto. De seguida, agrupam-se os pontos de acordo com a diferença entre os seus vetores normais e um vetor normal médio de uma região em expansão através de operação limiar (*threshold*). É efetuada uma pré ordenação das regiões de acordo com a sua área, começando das de maior área para as de menor, sendo excluídas as regiões de menor área dos passos que se seguem. Antes de calcular métricas, a superfície é simplificada para um polígono plano convexo. Para isto, em primeiro lugar, todos os pontos pertencentes a uma região são projetados para o plano que melhor descreve a superfície. De seguida, desde o centro do plano são geradas 360 pesquisas em linhas igualmente espaçadas, retendo-se, para cada linha, os pontos pertencentes ao plano mais distantes ao centro, considerando-se que pontos cuja distância seja inferior a um valor limite pertencem à linha. O último passo desta simplificação consiste em descartar todos os pontos responsáveis por tornar o polígono côncavo. A informação recolhida engloba a área do polígono convexo e a posição dos pontos referentes ao seu centro e ao vértice pertencente ao mais distante deste centro. De seguida, de modo a estimar a semelhança entre a identificação e o modelo, é efetuada comparação entre a área, a distância da peça e do modelo segundo $erro = \frac{|valor_1 - valor_2|}{valor_2}$, onde se compreende, $valor_2$ é uma característica do modelo.

Para haver identificação, o valor associado a este erro deve ser inferior a um valor limite, sendo referido em [11] o valor 0.1, em ambos os casos. Deve-se notar que cada plano convexo é comparado com um número de planos convexos extraídos do modelo, assegurando a funcionalidade do algoritmo para diferentes pontos de vista,

⁶Matriz bidimensional onde a posição de cada elemento fornece informação no plano XoY e cada elemento contém informação sobre a profundidade.

não sendo no entanto referido o número ou forma sistemática de os obter. Segue-se a estimativa da pose da peça. Neste algoritmo é efetuado o alinhamento entre os pontos do centro do polígono convexo, as normais associadas ao plano do polígono convexo e o versor entre o ponto do centro do plano e o ponto mais distante deste. Este processo encontra-se ilustrado na Figura 2.1. Para verificar o resultado obtido é efetuada uma amostragem do modelo na qual se faz a interpolação da profundidade com base num *kernel* gaussiano. De seguida são contados os pontos para os quais a diferença entre a profundidade da amostragem do modelo e da peça é inferior a um valor limiar. Se a razão entre o número de pontos contados e o número de pontos obtidos através do modelo for superior a 0.9 considera-se que a identificação corresponde ao modelo.

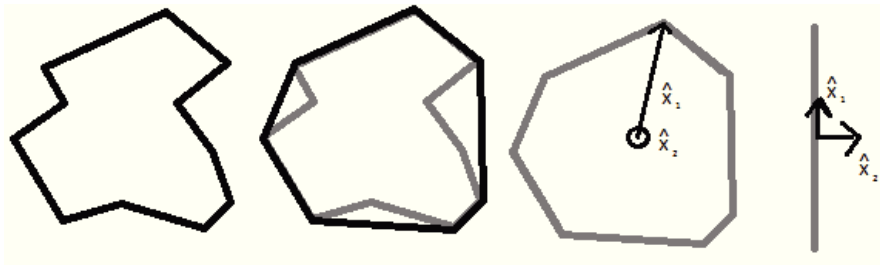


Figura 2.1: Simplificação de forma e determinação de orientação de polígono plano convexo.

Para avaliar o desempenho do algoritmo, os autores efetuaram três testes para 19 caixas com um único tipo de objetos empilhados aleatoriamente. Considerando como sucesso a verificação visual da melhor correspondência, obtiveram sucesso em 33 dos 57 testes. Como fragilidades conhecidas os autores indicam a extração de planos com áreas pequenas e a presença de sobreposições em objetos de topologia complexa, não sendo possível a extração dos seus planos com a totalidade da sua área. O algoritmo é considerado rápido pelos autores, ocorrendo cada iteração em menos de 3s num PC de 2.0G Hz podendo ser utilizada em tempo real se for implementada em *Field Programmable Gate Arrays* (FPGA). Este algoritmo necessita de peças com um volume considerável, várias vezes superior ao ruído associado ao sensor. A utilização de objetos simétricos leva também a piores resultados.

Outra abordagem baseada na deteção de planos é exposta em [21]. Os autores utilizaram o *scanner laser* ModelMaker D100, retirando amostras com distância entre si de 0.4mm a 1.0mm. O método utiliza como fonte de informação nuvens de pontos⁷ obtidas. Nesta abordagem procede-se ao cálculo das normais das superfícies que descrevem os dados recolhidos, seguido-se a segmentação dos dados através de um método de regiões em crescimento, com base nos ângulos associados à normal, à distância ex-

⁷Estrutura de dados composta por um conjunto de pontos e a sua localização no espaço.

pectável entre pontos no espaço devido à discretização da medição pelo sensor e às descontinuidades associadas à borda dos objetos. As superfícies referentes a segmentos constituem planos. Os autores calculam os descritores *Fast Point Feature Histograms* (FPFH) [22] para cada uma das regiões, utilizando o método *Sample Consensus Initial Alignment* (SAC-IA) descrito em [23] e [22] para encontrar a correspondência e a transformação entre as regiões segmentadas e o modelo. A melhor correspondência é utilizada para a inicialização do algoritmo *Iterative Closest Point* (ICP) que é utilizado para obter a pose (final) do objeto. A Figura 2.2 contém ilustrado o conceito utilizado. Os autores não apresentam resultados de forma quantitativa, utilizando imagens para mostrar resultados com bom alinhamento. Os autores apontam como fragilidades a possibilidade de obter má inicialização devido à semelhança dos descritores em superfícies planas. Os autores identificaram este fenómeno no caso em que peças se encontram com sobreposições ou amostragem de pontos não uniforme. Como os descritores são baseados em informação espacial é importante obter boa resolução para utilizar este algoritmo.

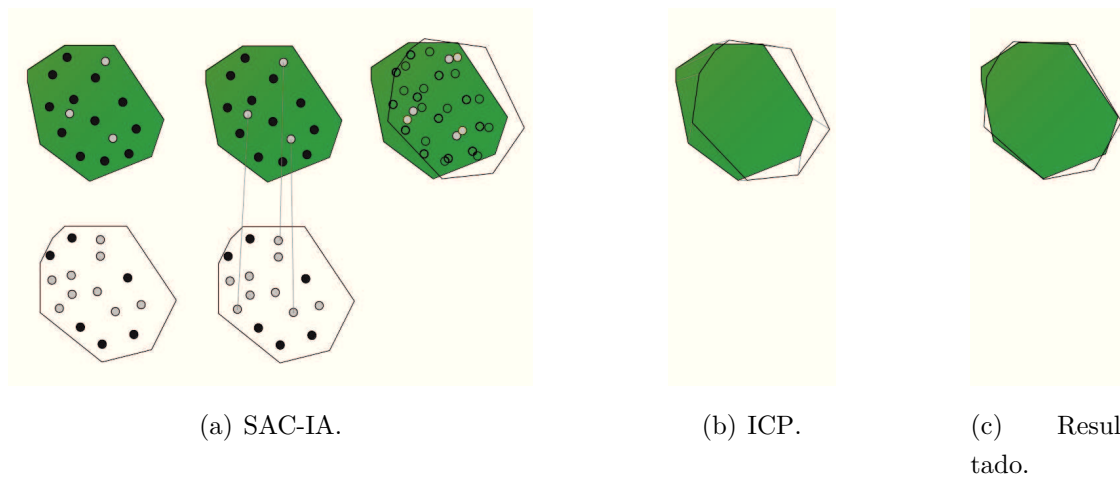


Figura 2.2: Ilustração de segmentação e alinhamento baseado no cálculo de descritores e ICP.

Primitivas Cilíndricas

De modo a reconhecer objetos cilíndricos no âmbito de aplicações de *bin-picking* foram utilizadas características elípticas associadas à vista de um dos topos de um cilindro [24] [4] e propriedades inerentes à simetria rotacional [25]. Para o caso de estudo utilizado na abordagem [24], deve-se ter em atenção que a utilização de cilindros com cores diferentes para os lados(preto) e para a base(branco), assim como a forma dos cilindros

que possuem altura razoável e raio que aparenta ser superior à altura⁸, simplifica o processo de reconhecimento de formas, o que em muitos cenários não é aplicável.

Em [24], a identificação de elipses começa com a determinação de contornos fechados numa imagem. De acordo com uma estratégia quasi-*Random Sampling Consensus* (RANSAC). De seguida, os autores estimam os parâmetros dessa mesma elipse, nomeadamente os seus eixos maior e menor, de acordo com o método [26]. É depois feita uma classificação das elipses e, das elipses não rejeitadas é escolhida a que corresponde ao objeto mais elevado e considerado possível de capturar. O algoritmo demora 37ms num computador *dual-core* de 3.0GHz, com um erro associado a uma deteção de 1 pixel com a distância de 20cm entre a câmara e o objeto. Não são especificados pelos autores os parâmetros focais da câmara, não sendo possível obter, em unidades métricas, a distância à qual esse erro corresponde.

Outra abordagem para deteção de elipses [4] baseia-se em segmentos de arco em vez de contornos fechados. Este algoritmo começa com a deteção de arestas recorrendo ao método Canny[27], gerando depois um conjunto de estimativas de arcos de circunferência baseado na avaliação da curvatura dos segmentos e na sua vizinhança. De seguida estimam-se os parâmetros da elipse que melhor correspondem ao arco segundo o método em [28]. Devido a uma elipse poder ser representada por vários segmentos de aresta, o processo origina múltiplas deteções de elipse com parâmetros muito semelhantes. As elipses com parâmetros semelhantes são encontradas, associando-se os segmentos de arestas correspondentes sendo utilizados para a determinação de uma nova elipse. Destas novas elipses são descartadas as elipses que se considera conterem parâmetros fora do esperado. Segue-se a determinação da circunferência tridimensional a que correspondem as elipses. Isto é feito com base no processo [29], utilizando-se o raio do círculo que se pretende detetar como um dos parâmetros. A validação das circunferências obtidas é efetuada através verificação de correspondência entre circunferências detetadas de dois pontos de vista diferentes. A escolha do objeto a capturar é efetuada com base:(1) na sua visibilidade, medida através da distância entre os pontos do modelo com os parâmetros estimados e a imagem detetada, (2) na normal das faces do cilindro se encontrarem alinhadas com a direção normal ao plano da câmara e (3) o objeto se encontrar no topo. Com este processo obtiveram-se erros de 2mm na determinação do centro de círculos e erros de cerca de 4×10^{-3} radianos para a orientação dos círculos.

A abordagem descrita em [25] é baseada na simetria rotacional. Esta abordagem utiliza como fonte de informação um mapa de profundidade obtido através de iluminação es-

⁸Não são fornecidas medidas, esta observação advém das análise das imagens documentais.

truturada. Esta abordagem começa com a extração de cantos e arestas, baseando-se no método de detecção de descontinuidades de Wani [30]. As descontinuidades encontradas são utilizadas para segmentação dos objetos. Segue-se a detecção de simetria rotacional utilizando a noção de a normal de um ponto numa circunferência ser colinear ao raio da mesma para esse ponto, considerando-se que a interseção de duas normais identifica o centro de uma circunferência. Identifica-se o cilindro que melhor aproxima uma região e divide-se em segmentos segundo planos perpendiculares ao eixo de simetria rotacional. Para cada segmento estima-se a circunferência que melhor o representa, obtendo-se um novo centro. Determina-se a linha que melhor representa os centros de circunferência de todas as secções, obtendo-se um novo eixo. A estimativa do novo eixo encontra-se ilustrada na Figura 2.3. O eixo e o seu centro são utilizados para estimar a pose do objeto. O raio em cada segmento é utilizado para identificar o objeto numa base de dados de objetos. Obtiveram-se erros de 0.15mm para o raio e de cerca de um grau para a orientação. Não é fornecida informação relativa ao tempo de execução.

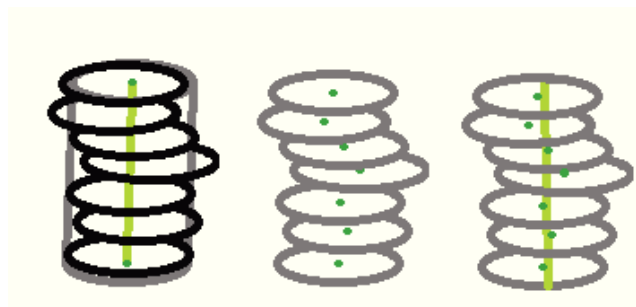


Figura 2.3: Ilustração de melhoria da estimativa de eixo para um objeto com simetria rotacional.

Utilização de múltiplas primitivas

Existem abordagens que utilizam conjuntos de várias primitivas para a identificação e determinação de pose de objetos como [31] e [32]. Estes algoritmos possuem associado um elevado tempo de computação, o que o limita a utilização em aplicações em tempo real. Para peças descritas de modo aproximado por apenas uma primitiva, tornam-se desnecessários ou de mais difícil implementação conceitos como a pesquisa de pontos de interseção entre primitivas utilizado em [31], ou a pesquisa de transformação entre primitivas de um modelo utilizado em [32]. Ambos utilizam informação de profundidade para encontrar as primitivas, pelo que a resolução de profundidade seja bastante inferior à dimensão das peças de modo a garantir uma correta aproximação das mesmas.

Em [31] recolhe-se do ambiente um mapa de profundidade. De seguida segmenta-se este mapa de acordo com as normais ao plano que melhor descreve a vizinhança de cada ponto. Posto isto procura-se, com os segmentos obtidos, primitivas geométricas cilíndricas e planas. Pontos identificáveis no mapa de profundidade e no modelo como pontos de interceção entre primitivas são utilizados para inicialização do algoritmo ICP. O algoritmo demora 24,36s num computador com processador Intel core 2 duo E 7500 a 2.93 GHz com 2GB de *Random Access Memory* (RAM). Verificaram-se erros de posição de cerca de 22mm, e de rotação de cerca de 2 graus. Em [32], recorre-se a uma Kinect para encontrar primitivas geométricas como esferas, planos e cilindros, que são utilizadas para descrever um modelo de objeto. O modelo dos objetos é constituído por uma árvore na qual cada nó corresponde a uma primitiva e cada arco conecta primitivas próximas no espaço e identifica a pose relativa entre as primitivas. Na informação sensorial do robô são identificadas primitivas, sendo de seguida criadas árvores de acordo com a proximidade das primitivas detetadas. As árvores criadas são comparadas com o modelo, começando por se procurar um arco aleatório correspondente nas duas árvores a partir do qual se expande a pesquisa aos arcos a si ligados na árvore. A melhor correspondência entre árvores é utilizada para obter a estimativa de transformação a fornecer ao algoritmo ICP (de acordo com a implementação de convergência quadrática [33]) utilizado para obter a correta transformação entre modelo e deteção. No caso de apenas ser detetada uma primitiva, não se podendo encontrar arcos entre primitivas, o algoritmo é adaptado. Os investigadores utilizaram o robô Cosero que incorpora um *notebook quadcore* com o processador Intel i7-Q720. A componente de perceção e deteção de peças foi executada em 83s. Não são fornecidos resultados para a o erro na determinação de pose.

2.1.2 Utilização de características ou aspeto visual

Em [34] utilizou-se um "SICK LMS 400 laser line scanner", com um erro angular de $\pm 0.1^\circ$ e $\pm 4\text{mm}$ de posição, para obter nuvens de pontos⁹ do ambiente. Extraem-se pares de pontos de uma nuvem e as respetivas normais¹⁰ e calculam-se, como características, a distância entre dois pontos, o ângulo entre as normais e os ângulos entre cada uma das normais e o vetor entre os dois pontos. Para procurar correspondências entre as características associadas a um par de pontos do modelo e de uma nuvem de pontos obtida, utiliza-se o método *Random Sample Matching* (RANSAM). Para acelerar o

⁹Estrutura de dados que contém um conjunto de pontos, aos quais se encontra associada uma posição.

¹⁰Entenda-se por normais de uma nuvem de pontos os vetores normais ao plano que melhor descreve a vizinhança de cada ponto.

processo que visa encontrar correspondências, é construída uma tabela indexada com os valores já processados, o que leva os autores a afirmar que, com o decorrer das sucessivas iterações, o tempo de busca tende para o valor constante de pesquisa de uma entrada na tabela indexada. A relação encontrada é verificada através da percentagem de pontos que, após transformação, não se encontram a uma distância superior a um limite definido. O algoritmo demora 4s num computador com um processador de 2,4GHz. O algoritmo cometeu erros referentes a uma translação de 9mm para suportes de vigas de aspeto brilhante e 3mm para os mesmos objetos pintados com tinta não refletora.

Em [8] utilizam-se os sensores Sick LMS 400 e Leuze LPS 36 do tipo "*line scanner*". O segundo sensor possui resolução entre 1mm, a 3mm. Obtêm-se características baseadas na comparação da profundidade entre um pixel e a sua vizinhança, obtendo-se, para cada comparação, os resultados: maior, menor ou igual. Com base na ideia de ocorrências desta característica influenciarem a probabilidade de o objeto se encontrar em determinadas poses, utiliza-se a acumulação de evidências relativas às múltiplas ocorrências destas características para estimar poses prováveis de objetos [35]. As poses mais prováveis são verificadas através de *template-matching*. Os autores afirmam que o processo demora, em média, menos de 350ms, notando-se nos dados fornecidos que, por vezes, o valor chega aos 1,5s com a utilização de um computador *Quadcore*, não mencionando a sua velocidade. Devido à comparação de profundidade com pontos na sua vizinhança, a resolução de profundidade deve ser bastante inferior à dimensão das peças de modo a evitar falsas deteções.

Em [36] e em [37] é utilizado o método de *Geometric Pattern Matching* (GPM) [38] para encontrar correspondências entre imagens e modelo, obtendo uma estimativa da pose do objeto. Esta pose é posteriormente refinada de acordo com diferentes processos. Em [36] utilizam-se os vetores próprios associados à nuvem de pontos para inicializar o algoritmo ICP. Em [37] a melhoria da deteção de pose é efetuada através do cruzamento dos resultados obtidos de dois pontos de vista diferentes. O algoritmo [37] obteve erros médios de 1mm e 0.3graus e máximos de 2.5mm e 1.5graus. Estes algoritmos são parte de um produto comercial não disponível durante a execução do projeto. Para além disto, um teste de uma aplicação com estes algoritmos deve ter em atenção a possível ausência de características utilizadas na pesquisa¹¹, devido ao aspeto brilhante associado às peças e a oclusões [39]. Outro aspeto a ter em consideração é o elevado espaço de procura em rotação e translação resultar num processo de reconhecimento mais lento [39].

¹¹Arestas, sentido entre a zona de menor brilho e maior brilho ao longo da aresta, arestas fechadas ou abertas.

O aspeto visual de um objeto numa imagem varia de acordo com o seu ponto de vista. Uma abordagem para contornar esta variação é a obtenção de uma base de dados de imagens do objeto desejado a partir de múltiplos pontos de vista. Esta abordagem foi implementada em várias aplicações, sendo um exemplo [40]. Nesta abordagem são gerados mapas de profundidade virtuais correspondentes a várias vistas simuladas de um objeto, geradas de acordo com o sensor utilizado na aplicação. Estas vistas são obtidas para diferentes orientações e guardadas numa base de dados. Os mapas de profundidade capturados pelo sensor utilizado são comparados com os mapas virtuais. As melhores correspondências são utilizadas como estimativas para algoritmo ICP. O erro obtido foi de 1mm e menos de 5graus. Para este método é expectável um elevado tempo de execução. Sendo este método baseado em informação de profundidade, a resolução de profundidade deve ser bastante inferior à dimensão das peças de modo a evitar falsas deteções devido ao ruído.

Nas aplicações baseadas no aspeto visual uma das principais limitações é o espaço de procura. Em [41] a identificação da estimativa da pose é acelerada com recurso a computação massivamente paralela numa *Graphical Processing Unit* (GPU). Utiliza-se como entrada um mapa de profundidade obtido através do método de luz estruturada. Segmenta-se o mapa de profundidade, determina-se o segmento mais alto e utiliza-se o ponto com maior distância a arestas para inicializar um algoritmo de verificação de pose. Este algoritmo consiste no teste em simultâneo de todas as poses de uma base de dados, aplicando translações com vista a diminuir o erro. A melhor pose é escolhida para inicialização do algoritmo ICP. O algoritmo foi executado num computador com processador de 2.83GHz Intel Core2 Quad (Q9550) e placa gráfica NVIDIA GTX 280 com 1GB de memória vídeo, demorando em média 0.89s com erro médios de 0,68mm de posição e de menos de 1 grau em rotação. Para este método devem-se utilizar peças com espessura bastante superior à resolução do mapa de profundidade. Oclusões podem levar à obtenção de uma pose incorreta com o algoritmo de verificação de pose, resultando numa errada determinação de pose.

A abordagem utilizada em [42]e consiste em restringir o espaço de procura através da identificação de características na imagem. Estas características devem ser escolhidas de forma a serem robustas a ruído e de rápida computação. Estas características são armazenadas numa base de dados, servindo de mecanismo de indexação. Ao calcular estas características na informação fornecida pelos sensores e filtrar as entradas da base de dados que não as contêm (sendo considerada pouco provável), os autores afirmam aumentar a velocidade dos algoritmos sem diminuir a sua robustez e fiabilidade.

Em [43], utiliza-se como fonte de informação uma imagem proveniente de uma câmara,

de onde se extrai o gradiente. Neste algoritmo, utiliza-se a ideia de espaço de procura em escala¹² de informação referente a arestas observadas a partir de vários pontos de vista para reconhecimento de objeto. Procuram-se correspondências numa estrutura constituída por várias árvores nas quais as diferentes vistas se encontram agrupadas por semelhança. Em primeiro lugar, procuram-se correspondências no nível hierárquico superior, onde o nível de detalhe utilizado para descrição do modelo é inferior e se permitem correspondências mais ténues. De seguida, em todas as árvores nas quais se encontrou uma correspondência, tenta-se seguir a correspondência no nível inferior. O resultado obtido é melhorado através da minimização do quadrado do erro estimado para a correspondência de arestas segundo direções perpendiculares a estas. Neste algoritmo a comparação entre imagens e modelo é efetuada através da média do cosseno entre as orientações das arestas no modelo e peça. Devido a não serem calculadas arestas na imagem de pesquisa e os gradientes serem normalizados, os autores consideram que esta comparação suporta variações no contraste, com a ressalva de inversões de contraste locais (sendo um exemplo metade do fundo preto e outra metade branco), o que pode acontecer em casos reais devido a sombras e posicionamento relativo face a fontes de luz e câmara. Os autores também consideram esta medida robusta face a pequenas deslocações nas arestas pois a direção do gradiente não varia abruptamente na vizinhança de uma aresta. Esta comparação é efetuada para grupos de píxeis correspondentes a segmentos de aresta do modelo, considerando que o contraste não varia deste modo na sua proximidade, somando-se posteriormente as contribuições individuais de cada grupo. Esta comparação é robusta face a oclusões devido a se considerar que, na ausência de visibilidade de uma região de uma imagem sensorial ou do modelo, o ruído provoca direções aleatórias para a orientação dos gradientes contribuindo em média com soma zero, pelo que o valor da medida aproxima de modo grosseiro a percentagem de objeto não visível. As oclusões podem decorrer das arestas não se encontrarem visíveis devido a sombras e reflexões. Com este algoritmo obteve-se erro médio de 0.58mm e 0.48 graus em 0.33s num computador com um processador Intel Core 2 Duo de 3 GHz, para um cilindros. Um aspeto pertinente no trabalho [43] é a iluminação do objeto de interesse a partir de três pontos de vista diferentes com três cores diferentes (vermelho, verde e azul), de modo a melhorar a visibilidade das arestas. Este método utilizou uma câmara de 640 por 480 píxeis a distâncias entre 12cm e 22cm às peças a detetar.

Em [44] a iluminação é utilizada para encontrar, numa pilha de objetos, o objeto a capturar. Isto deve-se à ideia de o objeto superior possuir maior intensidade luminosa

¹²Com o afastamento da câmara obtêm-se representações dos objetos com menores dimensões. A noção de espaço de escala está associada ao nível de detalhe na medida em que uma imagem de maior resolução permite um maior nível de detalhe.

que os restantes por estes objetos se encontrarem sombreados. Em [13], utiliza-se um sistema de iluminação controlada onde as posições relativas entre as fontes de luz, a câmara e a caixa com material são conhecidas. São utilizadas três fontes de luz, uma vermelha, uma verde e outra azul. Na implementação do algoritmo as três luzes são ligadas uma de cada vez de modo a aumentar a robustez¹³. Na aplicação descrita em [13], onde as peças são bastante refletoras, a luz dirigida a uma câmara depende da orientação do material que a reflete. Este princípio é utilizado para obter simultaneamente a deteção de objetos e a sua orientação. Os autores criam uma base de dados com múltiplos pontos de vista de uma peça. Através de um processo de voto com porções de imagem extraídas da base de dados estima-se a pose mais provável. A pose obtida é refinada através do processo iterativo descrito em [45]. Caso o número de correspondências entre píxeis de arestas do modelo e da peça seja inferior a um valor limiar, a deteção é rejeitada. Este algoritmo, executado num computador com um processador Intel QuadCore de 3.2GHz demora 0.445s. Os valores de erro posição e orientação não são apresentados quantitativamente. Para objetos com aspeto brilhante não é razoável esperar a identificação do objeto no topo recorrendo apenas à intensidade luminosa como em [44]. Para peças com número de sulcos inferior às utilizadas em [13], é expectável um número menor de informação na forma de "porções de imagem" disponível para a determinação de pose. A presença de sujidade pode introduzir informação não desejada na forma de "porções de imagem", sendo necessários testes para aferir a capacidade de adaptação para estes casos.

Alguns algoritmos dividem o processo de deteção de pose em duas etapas, a captura na qual a determinação de pose pode ser apenas uma aproximação, seguindo-se um refinamento da pose com o objeto isolado. Os autores de [6] e [7] colocam esta divisão de um modo mais subtil referindo-se a ela como apenas uma hipótese, sendo apenas a determinação da pose com o objeto isolado efetuada caso se afigure necessário. O processo descrito em [46] pesquisa, com recurso a um mapa de profundidade, em primeiro lugar, apenas a informação necessária para capturar uma peça, determinando a sua pose posteriormente. Esta atitude é adotada com vista a reduzir o tempo de processamento e evitar que as hipóteses para geração de poses de captura referentes a deteções sejam inválidas¹⁴, pelo que não é necessário repetir o processo de determinação de pose várias vezes.

A procura pela maneira de agarrar um objeto consiste em percorrer uma imagem de profundidade com um *kernel* com a forma de círculo no qual se garante que a garra,

¹³Sem aumentar a complexidade no modelo.

¹⁴Procurando poses com o intuito de agarrar parte-se do principio que todas as deteções obtidas são válidas.

neste caso tipo pinça, cabe em qualquer orientação, podendo capturar uma região. O máximo deste *kernel* fornece a posição segundo \hat{x} e \hat{y} , o mapa de profundidade fornece a altura e a orientação é encontrada através da estimativa da linha que melhor representa o esqueleto topológico¹⁵ da região. Segue-se a comparação das hipóteses geradas, onde as medidas de qualidade apontadas são a altura da descontinuidade ou protuberância, a qual deve ser superior a um valor limite de modo a permitir uma captura e sem colisões entre a garra e outros objetos. Estas colisões são detetadas através da análise da área da imagem correspondente ao retângulo de ocupação da garra aquando da captura da peça. Uma vez agarrada uma peça segue-se a determinação da sua pose. A determinação de pose é obtida através da medição de parâmetros inerciais associados ao transporte da peça a partir dos quais são calculados os eixos principais de inercia e o centro de massa [47]. Com base nestes valores são geradas hipóteses para a pose do objeto. A ambiguidade presente nas hipóteses geradas é resolvida através da incorporação de informação geométrica sobre o objeto, a garra e características visuais adicionais. Num computador com processador de 3.6GHz o processamento de imagem para captura de uma peça demora 18ms. Devido à procura de descontinuidades num mapa de profundidade, os objetos a utilizar devem possuir dimensões bastante superiores à resolução de profundidade. Este método, publicado muito recentemente junho de 2014, constitui, no que respeita o processo de perceção que antecede a captura de uma peça, uma versão complementar do algoritmo proposto neste documento, incorporando informação sobre uma ferramenta do tipo pinça ao invés de uma ferramenta do tipo ventosa.

As abordagens que dividem o processo em duas etapas podem ainda beneficiar de deteção de colisões devido à monitorização da força exercida nas várias juntas do manipulador robótico, por parte do controlador deste, como o processo descrito em [9], para recuperar de pequenos erros em tentativas de captura de peças.

2.1.3 Segmentação e Determinação de Contornos

Nesta secção são abordados algoritmos de segmentação, deteção de contornos e determinação de pose entre objetos, que apresentam potencial interesse para a tarefa de *bin-picking*. Para além destes, são referidas algumas estratégias de deteção de objetos no domínio de distinção entre bases de dados de objetos diferentes, tipicamente sem sobreposição.

De acordo com a avaliação de algoritmos de segmentação com a base de dados de Ber-

¹⁵Tradução do termo em inglês *topological skeleton*.

keley¹⁶, o segundo melhor algoritmo para segmentação é o "gPb-ucm(color)"[48]. Neste algoritmo consideram-se variações em brilho, cor, textura, assim como a hierarquia entre regiões. O melhor algoritmo, segundo esta mesma página é o "*Discriminatively trained sparse code gradients for contour detection*"[49]. Este algoritmo utiliza *Orthogonal Matching Pursuit*[50] e *K-SVD*¹⁶ [51] para aprender representações de arestas como porções de imagens. A relação espacial dessas detecções é considerada e utilizada com *Support Vector Machine* (SVM) para detecção de arestas. O resultado é suavizado, procedendo-se à supressão de pontos não correspondentes a máximos nas arestas, para produzir o resultado final da detecção. Em [52] obtêm-se resultados semelhantes ao algoritmo [49] em intervalos de tempo três ordens de magnitude¹⁷ menores. Este algoritmo, também baseado em *machine learning*, utiliza florestas de decisão¹⁸. A maior limitação deste algoritmo, segundo os autores, é a incapacidade de incorporar nova informação relativa a pistas sobre arestas após a etapa de treino da floresta de decisão, aconselhando-se que a informação de treino se assemelhe à utilizada na execução do algoritmo. Um aspeto relevante para a tarefa de *bin-picking* é a duração do treino de florestas de decisão demorar cerca de 30 minutos por árvore de decisão (o processo pode ser paralelizado por árvore) num computador *Quad-core*. De acordo com os resultados nas bases de dados testadas, estes algoritmos correspondem ao estado da arte atual. O elevado tempo de execução dos algoritmos [48] e [49], invalida a sua utilização em aplicações de tempo real, não sendo contudo excluídas do auxílio à construção de segmentações consideradas "de referência". A necessidade de construção de base de dados anotada para treino dos algoritmos [49] e [52] é um fator com potencial para limitar a sua utilização em sistemas produtivos com constante substituição de matéria prima, sobretudo caso se considere voltar a treinar o sistema para objetos diferentes, caixas diferentes ou sistemas de iluminação diferentes.

Outras abordagens para a detecção de arestas utilizam múltiplos elementos sensíveis a: (1) arestas de diferentes orientações como [53], onde se utiliza o gradiente morfológico¹⁹ segundo oito direções diferentes ou (2) ao conceito de hierarquia na detecção de arestas [54] [55]. Em [54] considera-se que os contornos mais relevantes são detetados mesmo em imagens mais esbatidas (através de um filtro gaussiano). Por este motivo detetam-se primeiro arestas numa imagens mais esbatida seguindo-se arestas detetadas neste nível hierárquico para níveis de menor esbatimento de imagem, aos quais se associa uma

¹⁶Disponível em <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/bench/html/algorithms.html>.

¹⁷Segundo [52], [49] é executado a 1/280 imagens por segundo (fps), [48] a 1/240 fps e [52] entre 3 a 60 fps de acordo com o número de árvores da floresta de decisão e utilização ou não de múltiplas escalas.

¹⁸No sentido em que múltiplas árvores (de decisão) constituem uma floresta, tradução livre do termo em inglês *decision forests*.

¹⁹Diferença entre o resultados das operações de dilatação e erosão morfológica sobre uma imagem.

melhor localização. Deste modo os autores afirmam obter boa localização nas arestas e eliminar arestas associadas a ruído. Em [55], para além da hierarquia, é utilizada informação relativa à direção das arestas devido à utilização do domínio *contourlet*²⁰.

Para encontrar arestas em imagens pode-se recorrer ao algoritmo Canny [27]. Para este algoritmo são necessários dois valores limite, um para começar uma aresta e outro para continuar as arestas obtidas, ligando-as a novos elementos. Em [27] não são descritas formas de obter estes valores automaticamente. Vários métodos para este fim foram propostos [56],[57]. Em [56] utiliza-se o método de *Otsu* [58] para definir o limite superior e metade para definir o limite inferior. Em [57] combina-se a informação relativa ao gradiente da imagem utilizado no algoritmo Canny, com um processo de continuação de arestas.

Os resultados apresentados em [53] apontam para este algoritmo fornecer maior quantidade de arestas fechadas que algoritmos como Canny, notando-se nos resultados que esta propriedade surge com o custo de maior representação de arestas associadas a ruído. Isto não é de todo desejável para aplicações onde a distância a arestas é um parâmetro importante pois pode ser seriamente comprometida. Em [53] mostram-se resultados que combinam a omissão de ruído resultante da aplicação de filtros gaussianos sem a deterioração da localização associada. Considera-se que com a aplicação do algoritmo Canny já se obtém uma resposta razoável face ao ruído, não se afigurando necessária a aplicação dos conceitos desenvolvidos em [53]. No algoritmo [55] a adição de ruído altera significativamente o resultado da deteção de arestas. No exemplo fornecido, com a adição de ruído, novas regiões foram encontradas. A intervenção em termos de parametrização consiste num valor de *threshold* por escala. Para se encontrar este valor, os autores apontam para um algoritmo adaptativo, como este não é especificado compromete a replicação dos resultados. Apesar de parecer mais robusto a ruído gaussiano que o algoritmo Canny para o exemplo ilustrado, não se possui informação suficiente para afirmar que a aplicação de um algoritmo deste tipo seja vantajosa para outras aplicações do que o algoritmo Canny.

Em [59] é proposto um método para segmentação em três regiões com base no algoritmo *Otsu*[58], procurando-se aferir dois valores para a separação destes três segmentos. O processo de cálculo destes valores é efetuado por regiões da imagem. O algoritmo de cálculo dos dois valores limite é baseado num processo iterativo no qual cada iteração corresponde à divisão da região em duas através do algoritmo *Otsu* e ao cálculo do valor médio de cada uma das duas regiões. O intervalo entre os dois valores médios

²⁰Associado à transformada *contourlet*, generalização da transformada *wavelet* multidimensional, incorporando informação relativa a direção, consultar [55].

obtidos é considerado "por determinar", sendo o processo repetido para este intervalo enquanto a diferença entre os dois valores obtidos for superior a um valor limite. Este algoritmo foi publicado recentemente, em março de 2014, quando foi lido já se encontrava implementada uma aplicação de segmentação com resultados satisfatórios, não existindo a necessidade de testar este algoritmo.

A abordagem descrita em [60] é baseada somente em informação de profundidade. Nesta abordagem, procura-se dividir um mapa de profundidade em segmentos planos. Para este efeito são calculadas a magnitude e a direção do gradiente da profundidade. Começa-se por agrupar regiões com a mesma direção de gradiente de profundidade com base no histograma desta grandeza numa vizinhança de n por n pontos, na qual não se consideram os pontos de elevada magnitude de gradiente (por estarem associados a descontinuidades em planos). De seguida, para cada região obtida, calcula-se o plano que melhor a descreve através de RANSAC [61]. Se a distância entre duas regiões de acordo com a normal dos planos que as descrevem for baixa, agrupam-se as regiões. Posto isto, com vista a melhorar a qualidade dos resultados, todos os pontos rodeados por pontos de apenas uma região são considerados parte dessa região e atribui-se aos pontos nas arestas a região para a qual estes pontos possuem menor distância segundo a normal dos planos. Com base neste algoritmo, para objetos curvos obtêm-se regiões planas que os descrevem de modo aproximado. Esta abordagem requer que a resolução do mapa de profundidade seja bastante inferior à dimensão da peça a analisar de modo a garantir uma correta aproximação da forma das peças através dos planos calculados.

2.1.4 Detecção de pose

Algumas propostas para deteção de pose de objetos foram desenvolvidas para modelos ou representações específicas de objetos. Para determinação de pose de objetos de simetria rotacional foi recentemente desenvolvido o algoritmo [62]. A determinação de pose de objetos cuja descrição consiste em contornos num plano pode ser efetuada através de [63]. Uma classe de algoritmos de aplicação genérica é o ICP [64]. Este tipo de algoritmos consiste na execução de várias iterações de determinação de correspondência entre pontos seguida da translação e rotação que melhor alinha a correspondência. Existem múltiplas variantes, encontrando-se algumas descritas em [65] que analisa a convergência do algoritmo ICP[64] e propõe variantes de maior velocidade de convergência. Segundo [65] o algoritmo ICP[64] converge lentamente quando é necessário efetuar movimentos tangenciais ao longo de uma superfície, sendo uma das propostas para acelerar o algoritmo a consideração da distância face à tangente de planos que descrevem localmente uma superfície. Em [66] utiliza-se a distância entre planos re-

ferentes a nuvens de pontos que se pretendem alinhar, o que pode ser utilizado para explicar a mais célere convergência descrita face ao ICP[64]. Outra vantagem apontada para [66] é a menor influência de incorretas correspondências entre pontos o que torna o algoritmo mais robusto face a fracas inicializações.

Em [67] descreve-se como obter descritores invariantes a rotação e translação através de momentos. Os momentos são calculados no domínio dos números complexos. Os descritores invariantes correspondem às componentes real e imaginária de diferentes momentos. Uma das propriedades destes descritores é a possibilidade de identificação de inversões segundo um eixo (do tipo espelho), através dos momentos correspondentes a componentes imaginárias. A informação referente a inversões segundo um eixo pode ser utilizada para encontrar em imagens bidimensionais rotações segundo eixos não perpendiculares à normal do plano da imagem. Isto pode ser útil para escolher inicializações para um algoritmo do tipo ICP.

Deteção de Objetos sem sobreposição

Na literatura um método bastante popular para encontrar objetos numa imagem é "A Discriminatively Trained, Multiscale, Deformable Part Model"[68]. Este método utiliza detetores, resultantes de um processo de treino, baseados em histogramas de orientação de gradientes, utilizando o conceito de hierarquia e partes móveis. O algoritmo *Inverse Compositional Lucas-Kanade* (ICLK), baseado em *gradient descent*, é utilizado para lidar com variação de aparência de objetos de uma classe. Neste algoritmo, de forma iterativa, minimiza-se o erro quadrático entre um "*template*" e uma imagem, utilizando diferentes funções de transformação (entre imagem e modelo), solucionando diferenças em aparência e alinhamento em simultâneo. Os *Active Appearance Models* (AAM) combinam a informação referente à forma de um objeto com a informação referente a textura. Um conjunto de possíveis representações de um objeto é construída com recurso a bases de dados devidamente anotadas, contendo modificações de aparência conhecidas (podem compreender movimentos entre o objeto e a câmara, iluminação, ou deformações aplicadas ao objeto). Ainda na fase de treino, estimam-se as adaptações a efetuar ao modelo (movimento, deformações ao aspeto ou à forma do objeto), para corrigir diferenças face a uma imagem alvo. Durante a execução do algoritmo, as adaptações estimadas são aplicadas iterativamente ao modelo até convergência entre o modelo e a imagem, obtendo-se deteção, localização e deformação de um objeto. O artigo [69] descreve a forma de obter alinhamento entre imagens com *gradient descent* no domínio de *Fourier*, que pode ser utilizada em ICLK ou em AAM. A ideia avançada neste artigo apresenta a vantagem do custo computacional ser invariante face ao

número de filtros aplicados em etapas de pré-processamento no sentido em que a convolução corresponde a uma multiplicação no domínio de *Fourier*. Em [70] é apresentado um método para pesquisa de imagens sem treino prévio.

Estes algoritmos não foram desenvolvidos com vista a aplicações de *bin-picking*, contudo, estes algoritmos apresentam potencial interesse para a identificação de um tipo de peças numa base de dados.

2.2 Adaptação através de aprendizagem

Para a melhoria de sucesso de capturas de peças com o número de capturas efetuadas, em [71] propõe-se a utilização de informação referente ao sucesso de capturas já efetuadas na escolha das sucessivas capturas. Esta informação é utilizada na forma de medida de probabilidade de sucesso de uma captura para um conjunto de informação disponível. O sucesso de uma captura na tarefa *bin-picking* depende de um vasto conjunto de fatores entre os quais: (1) a pose do objeto a capturar, (2) a pose dos objetos na sua proximidade (devido a colisões ou bloqueio físico da peça no momento da captura), (3) sensores utilizados e (4) iluminação ambiente. Em [71], considera-se que as contribuições destes parâmetros estão incluídas de modo implícito no sucesso de uma captura, sendo utilizada informação relativa a tentativas de captura como forma de mapear, de modo aproximado, a probabilidade de sucesso associada a cada captura. São delineadas duas formas para escolher o modo de agarrar as peças: (1) escolha aleatória de acordo com a probabilidade e (2) escolha da forma de agarrar com maior probabilidade de sucesso. O primeiro método de seleção garante a convergência da estimativa de probabilidade, sendo expectável que com o número de capturas se opte com maior frequência pelas melhores formas de capturar as peças. Isto é obtido com o custo de se escolherem ocasionalmente as piores abordagens para captura de peças. O segundo método de seleção não garante a convergência dos resultados, sendo apenas expectável que se atualize a probabilidade de sucesso de algumas formas de captura. Uma consequência direta é não se atualizar a probabilidade de sucesso de formas de captura hipoteticamente melhores que as capturas selecionadas. Com isto pode-se sistematicamente selecionar modos de captura de peças com baixa probabilidade de sucesso. Os autores indicam a escolha de um pequeno grupo de formas de captura de peças cuidadosamente selecionadas para contornar esta situação.

Esta página foi intencionalmente deixada em branco.

Capítulo 3

Metodologia

Atualmente, segundo NIST [12], não existem aplicações que solucionem o problema de *bin-picking* de um modo genérico, existindo uma panóplia de aplicações dependentes de especificidades relativas a utilizador, processo e produto, ao invés de uma única aplicação. Deste modo importa clarificar a conseqüente exposição do processo implementado e o raciocínio associado à sua construção, apresentando as peças utilizadas como caso de estudo. Conhecendo-se as peças utilizadas, segue-se a exposição da constituição e configuração dos componentes físicos assim como do ambiente de desenvolvimento por *software* associados à implementação. Posto isto aborda-se, o processo de calibração necessário ao correto funcionamento do sistema. Depois, aborda-se o algoritmo implementado e o processo (raciocínio) de desenvolvimento do mesmo. Finaliza-se esta secção com o funcionamento do sistema como um todo.

3.1 Caso de Estudo

O presente trabalho utiliza como caso de estudo as peças presentes na Figura 3.1. As peças pesam 100 gramas, possuem aspecto metálico com elevada reflexão e não apresentam textura (padrões coloridos, sequência de cores). Observa-se igualmente que as peças são aproximadamente planas, possuindo poucas saliências e pouca espessura face às restantes dimensões. Pode-se ainda observar que várias peças possuem manchas, tal como realçado na Figura 3.2. Existem vários tipos de peças utilizados, encontrando-se três presentes na Figura 3.3, onde é bastante visível o efeito da refletividade na aparência das peças. Na secção 3.5 são apresentados os modelos utilizados para a sua representação.



Figura 3.1: Aparência de caixa com peças utilizadas no caso de estudo de diferentes pontos de vista.



(a) Irregularidades no aspeto de peças utilizadas no caso de estudo (face 1).



(b) Irregularidades no aspeto de peças utilizadas no caso de estudo (face 2).

Figura 3.2: Irregularidades no aspeto de peças utilizadas no caso de estudo.



(a) Diferentes peças utilizadas no caso de estudo.



(b) Diferentes peças utilizadas no caso de estudo.

Figura 3.3: Diferentes peças utilizadas no caso de estudo.

3.2 Configuração e Constituição Física

3.2.1 Constituição

O sistema, ilustrado nas Figuras 3.4, 3.6 e 3.7, é composto por:

1. Um manipulador industrial;
2. Um controlador para o manipulador industrial;
3. Uma ferramenta de captura de peças através de contacto com regiões planas desta recorrendo a um sistema de vácuo;
4. Um sensor com o objetivo de detetar peças capturadas pela ferramenta;
5. Um sensor que fornece imagem de cores RGB e mapa de profundidade;
6. Um computador;
7. Um padrão xadrez;
8. Uma superfície disposta de forma a constituir fundo controlado para deteção e verificação de pose de peças capturadas.

A aplicação projetada, descrita com mais detalhe na Secção 3.5, é adaptável a um sistema físico genérico composto por estes componentes, desde que estejam asseguradas as seguintes condições¹:

1. O conjunto formado pelo manipulador e ferramenta se poder movimentar no espaço tridimensional e possuir a capacidade de rodar a ferramenta ao longo de, pelo menos, dois eixos perpendiculares;
2. A ferramenta deve ser capaz de sustentar a peça;
3. Possibilidade de transmissão de informação entre o controlador e um computador;
4. Deve ser possível obter imagem da peça sobre a superfície que constitui fundo controlado;
5. Deve ser possível separar com base em cor ou intensidade luminosa a peça do fundo controlado utilizado.

¹Devendo-se verificar se a resolução da informação obtida pelos sensores é adequada.

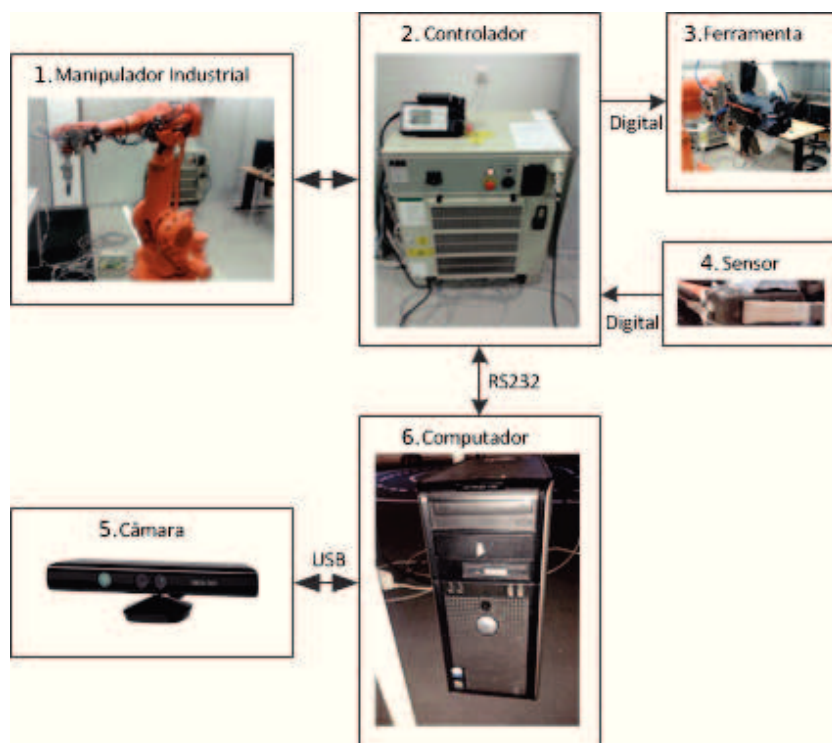
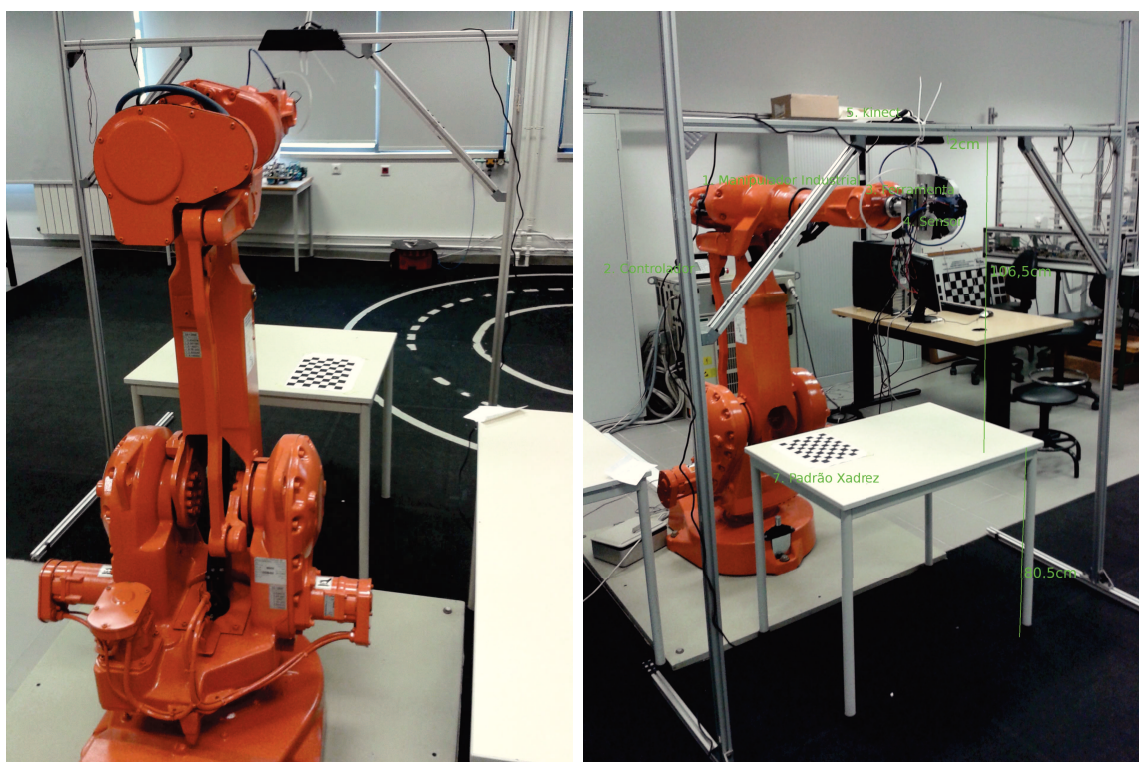


Figura 3.4: Interligação dos componentes do sistema.



(a) Disposição dos elementos do sistema associados à percepção e manipulação física das peças vista 1 .

(b) Disposição dos elementos do sistema associados à percepção e manipulação física das peças vista 2.

Figura 3.5: Disposição dos elementos do sistema associados à percepção e manipulação física das peças.

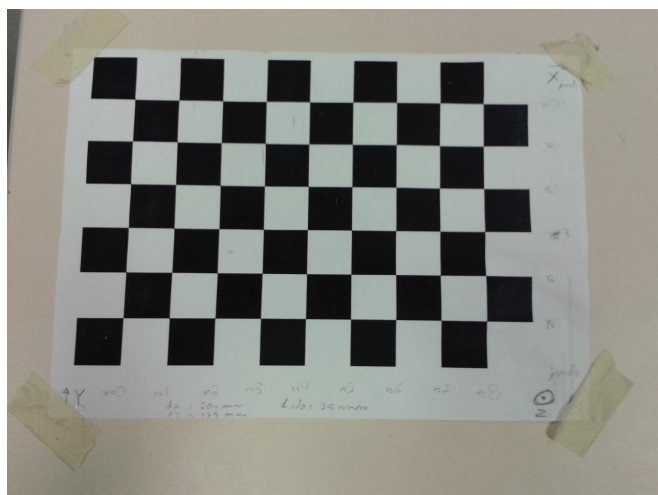


Figura 3.6: Padrão xadrez.

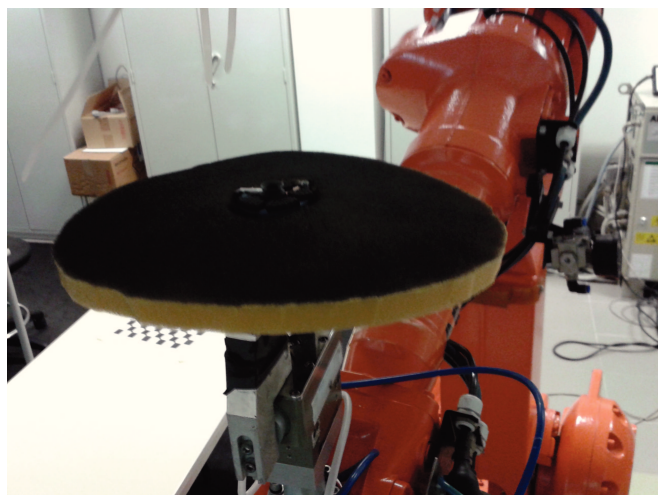


Figura 3.7: Fundo controlado na garra robótica.

3.2.2 Especificações relativas à implementação

A escolha do tipo de sensor para fornecimento da imagem de cores e mapa de profundidade, em conjunto com a escolha do manipulador robótico, influenciam a disposição da célula de trabalho. A configuração utilizada neste caso pode ser vista na Figura 3.5. O sensor utilizado é a *Kinect* para a Xbox360 da Microsoft. Este sensor exige uma distância mínima entre as peças alvo e a câmara de modo a obter leituras do mapa de profundidade. Peças com aspeto metálico apresentam dificuldades acrescidas ao sistema de deteção de profundidade baseado na triangulação dos pontos projetados através da sua leitura na câmara de infravermelhos, na Figura 3.8 pode-se visualizar a projeção do padrão de infravermelhos da Microsoft Kinect para Xbox 360 para blocos de madeira e peças metálicas utilizadas.

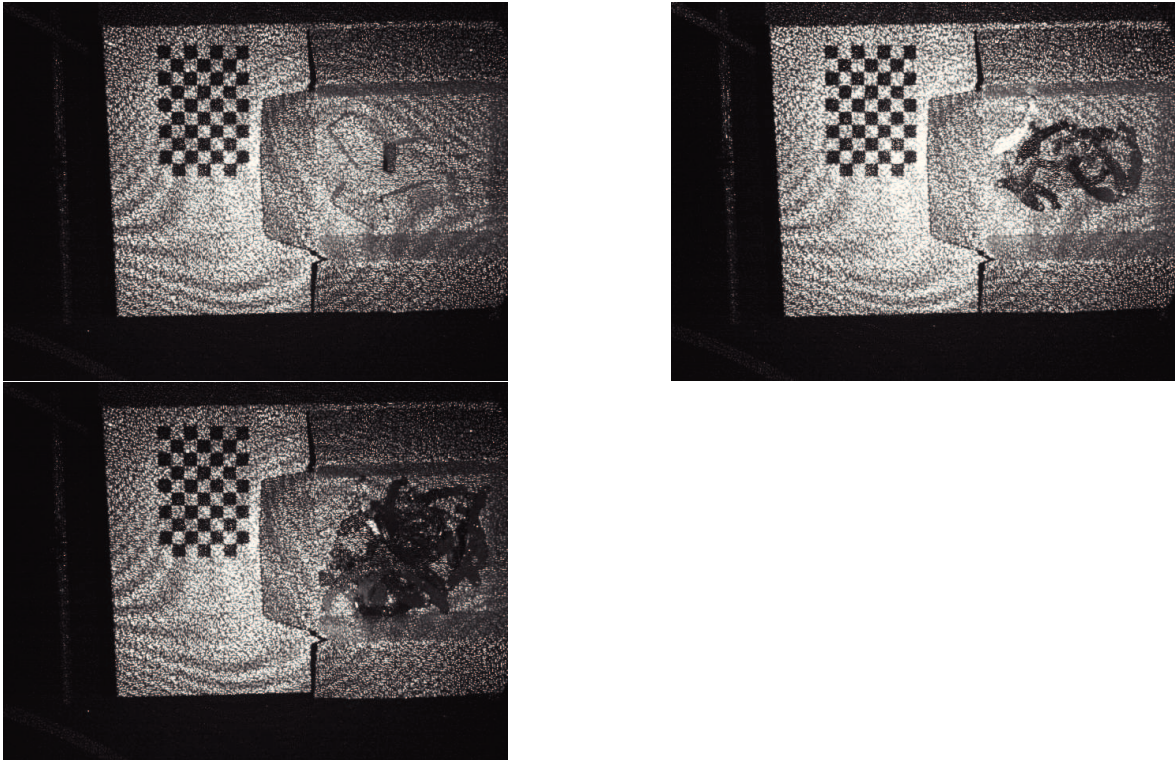


Figura 3.8: Aspeto do padrão de infravermelhos produzido pela Kinect para peças de madeira e metal.

O manipulador robótico utilizado é um IRB2400 de 1,5m da ABB. Devido ao campo de trabalho para este manipulador a caixa com as peças encontra-se próxima da base deste, em cima de uma mesa. As especificações detalhadas referentes aos elementos utilizados na implementação encontra-se disponíveis para consulta no anexo A.

A ferramenta consiste num circuito de vácuo ligado a uma ventosa, cujo diâmetro foi escolhido tendo por base a geometria das peças no processamento que antecede a captura de peças.

A folha com padrão xadrez é utilizada na calibração do sistema (ver Secção 3.4).

Como fundo controlado é utilizada uma superfície flexível e escura colocada em torno da ferramenta. A necessidade desta superfície será também explicada em mais detalhe posteriormente.

3.2.3 Iluminação

Na sala onde se encontra o sistema implementado, ilustrada na Figura 3.9, as janelas estão a altura de 1 metro do chão, e as luminárias, a 3 metros do chão. Não se utilizam técnicas de iluminação controlada, pelo que esta varia ao longo do dia.



Figura 3.9: Posição das luminárias, janelas e manipulador robótico na sala

3.3 Desenvolvimento do software

O sistema de *software* desenvolvido encontra-se esquematizado na Figura 3.10. Este sistema pode ser sub-dividido em (1) aplicação para o controlador robótico e (2) aplicação de processamento e controlo do processo.

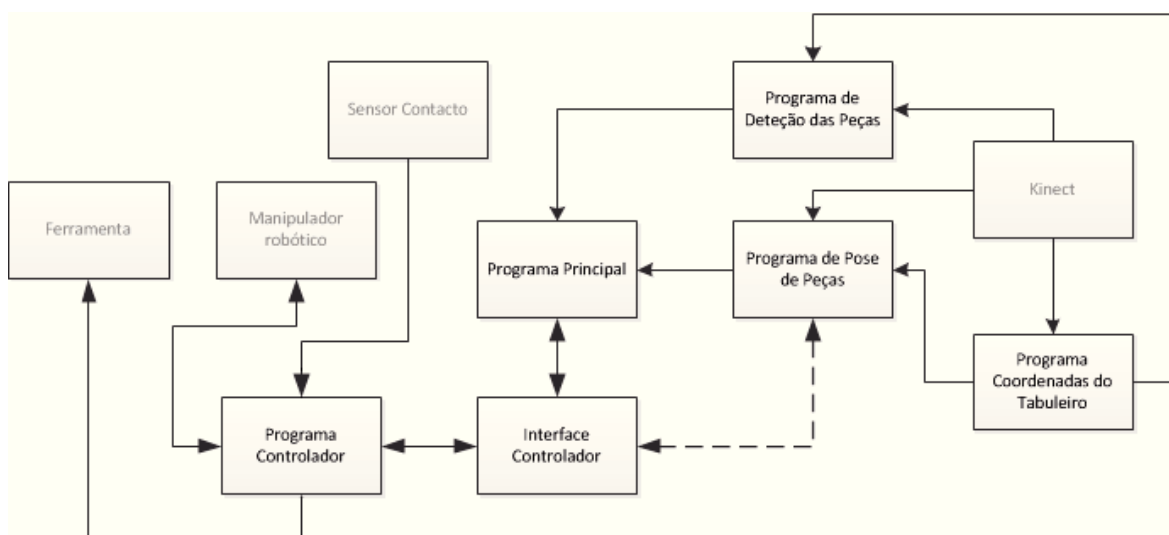


Figura 3.10: Sistema de *software* desenvolvido.

3.3.1 Aplicação de processamento e controlo do processo

Ambiente de desenvolvimento

Os programas da aplicação de processamento e controlo do processo comunicam entre si através das ferramentas *Robotic Operating System* (ROS). ROS é um ambiente de programação que fornece abstração de *hardware*. Engloba bibliotecas e convenções de *software* que facilitam a produção modular de *software*, a comunicação entre programas e a integração de funcionalidades previamente desenvolvidas. Para além disto possui ferramentas de visualização que facilitam a verificação dos resultados.

Optou-se por utilizar ROS, por este facilitar a construção de aplicações de modo modular, interligando diferentes componentes facilitando testes e desenvolvimento, contando com:

1. Funcionalidade de ligação a um grande conjunto de sensores, permitindo um

sistema de troca de dados normalizado entre programas, ou programas e sensores, bem como guardar a informação trocada para utiliza-la posteriormente;

2. Ferramenta em software aberto, com bastante documentação e tutoriais;
3. Vasta comunidade de utilizadores, possuindo plataformas como o sítio Web <http://answers.ros.org/questions/> onde membros desta comunidade se auxiliam;
4. Ferramentas de visualização de dados.

Como se optou por utilizar ROS, que apenas suporta oficialmente o sistema operativo do tipo Ubuntu, utilizou-se o Kubuntu. A versão utilizada foi a 12.04, por permitir o acesso às versões hydro, a mais recente no início do projeto, e fuerte (algumas aplicações no fuerte² não foram transferidas para groovy² ou hydro², como a aplicação de calibração extrínseca utilizada, descrita na Secção 3.4).

A linguagem de programação utilizada foi C++ pelas seguintes razões:

1. Compatível com ROS;
2. Largamente utilizada, possuindo vasta comunidade de utilizadores que prestam apoio mútuo;
3. Existem muitas bibliotecas, projetos e rotinas em código aberto como OpenCV, *Point Cloud Library* (PCL), Boost, Eigen que são desenvolvidas em C++ e/ou suportam desenvolvimento em C++;
4. Das linguagens suportadas pelo ROS, C++ e Python, possui a potencialidade para gerar código mais rápido/otimizado;
5. Vasto número de *Integrated Development Environments* (IDE), vários destes livres;

O IDE utilizado foi o QtCreator 4, pelas seguintes razões:

1. Plataforma de desenvolvimento livre;
2. Facilidade em criar aplicações gráficas (monitorização ou controle), se necessário;
3. Utilizado pela comunidade ROS e PCL.

Utilizou-se, para sistema de compilação, o cmake e gcc por ser utilizado em ROS.

²furete, groovy e hydro são três versões do ROS. A versão fuerte foi lançada em abril de 2012, groovy em dezembro de 2012 e hydro em setembro de 2013.

Escolha de *Software Frameworks* a usar com Microsoft Kinect para a Xbox 360

Para a Kinect existem vários *drivers* e *software frameworks* [72, 73]: (1) Microsoft *Software Development Kit* (SDK), (2) OpenNi, (3) Freenect da OpenKinect, (4) Matlab. Estas diferem entre si em vários aspetos, como a obtenção dos valores relativos ao mapa de profundidade sem qualquer linearização [72], como é o caso (3), ou após linearização dos valores expressando-os em métrica, como é o caso dos restantes [72, 73]. Outra diferença é gama de distâncias para as quais se obtêm medições de profundidade, para a Kinect da Xbox360 em específico, a Microsoft SDK e a *driver* utilizada com o Matlab, apenas devolvem valores correspondentes à profundidade na gama de 0,8 metros a aproximadamente 4 metros, enquanto que as restantes devolvem valores correspondentes à gama de 0,5 metros a mais de 9 metros [72, 73].

Para escolha do *driver* a utilizar foram utilizados as seguintes características: (1) Possibilidade de utilizar Kinect mais próxima dos objetos a filmar; (2) Distâncias do mapa de profundidade em escala métrica ou linearizada; (3) Compatibilidade com sistema operativo Ubuntu (único sistema operativo suportado oficialmente pelo ROS); (4) Facilidade de Integração com ROS e biblioteca PCL, utilização nas comunidades ROS e PCL. As duas primeiras características já foram abordadas. A primeira coloca OpenNi e Freenect em vantagem face às restantes enquanto que a segunda coloca Freenect em desvantagem às restantes. A terceira característica apenas não é cumprida pela Microsoft SDK. Tanto a Microsoft SDK como a *driver* utilizada com o Matlab não cumprem a quarta característica. A OpenNi oferece maior facilidade de integração, comparativamente à Freenect, com a biblioteca PCL devido à existência de tutorial para o efeito no sítio Web do PCL, o que indica maior propensão da sua utilização por esta comunidade. Na comunidade ROS é possível encontrar apoio à utilização da OpenNi e da Freenect, sendo que uma rápida pesquisa no sítio <http://answers.ros.org/questions/> permite aferir que atualmente ambas são utilizadas. Por estes motivos, optou-se pela OpenNi.

Programa de Coordenadas do Tabuleiro

Este programa recebe nuvens de pontos e imagens de cores da Kinect. O objetivo deste programa é produzir e disponibilizar uma matriz de transformação entre o referencial da câmara a cores da Kinect e o referencial associado ao padrão xadrez (ver Secção 3.2). Detalhes sobre a obtenção desta transformação podem ser consultados na Secção 3.4.2.

Programa de Detecção das Peças

O programa de detecção das peças recebe nuvens de pontos da Kinect, imagens de cores da Kinect e uma matriz de transformação entre o referencial associado à câmara e o referencial associado ao padrão xadrez. Este programa tem como objetivo detetar peças no mundo e enviar uma posição para a sua captura. O processo de detecção de peças encontra-se detalhado na Secção 3.5.1.

Programa Pose de Peças

Este programa recebe imagens de cores da Kinect, a posição da ferramenta no referencial associado ao padrão xadrez e a transformação entre o referencial associado à câmara e o referencial associado ao padrão xadrez. O objetivo deste programa é fornecer a pose de peças na ferramenta. Detalhes deste processo podem ser consultados na Secção 3.5.2.

Interface Controlador

Este programa serve de intermediário entre a aplicação do controlador e aplicações no ambiente ROS. É através deste programa que o Programa Principal comanda o manipulador robótico e que o Programa Pose de Peças obtém a posição da ferramenta no referencial associado ao padrão xadrez. As funcionalidades e características deste programa encontram-se detalhadas no Anexo C.

Programa Principal

Este programa recebe informações referentes à posição da ferramenta, às juntas do manipulador industrial, ao estado do sensor de contacto da ferramenta, à posição para captura de peças e à transformação referente a pose da peça na ferramenta. O objetivo deste programa é atuar a ferramenta e movimentar o manipulador robótico. O funcionamento deste programa encontra-se detalhado na Secção 3.6.

3.3.2 Aplicação do controlador

Ambiente de desenvolvimento

O controlador robótico pode influenciar a escolha do ambiente de desenvolvimento em *software* a utilizar. O controlador utilizado é um IRB S4C Plus da ABB, sendo a linguagem de programação associada o RAPID. A utilização desta linguagem torna a aplicação imediatamente portátil para outros produtos ABB. Devido à versão do controlador ser 4, não existe compatibilidade com as ferramentas para controlo proporcionadas pelo *ROS Industrial*, um projeto que utiliza as funcionalidades ROS com vista a aplicações destinadas à indústria como controlo de movimentos de manipuladores robóticos.

Como ferramentas de desenvolvimento foram utilizados:

1. RobotStudio da ABB;
2. Notepad++.

Ambos os programas são executados no sistema operativo Microsoft Windows 7. O RobotStudio permite acesso a documentação das funções disponíveis, verificação de erros e simular movimentações do manipulador robótico. Como alternativa, para pequenas alterações foi utilizado o Notepad++ por permitir a marcação de palavras chave e marcadores, facilitando a leitura de código.

Graças à utilização da linguagem RAPID, o sistema produzido pode ser utilizado com outros robôs do fabricante ABB, mesmo com controlador versão 5.

Programa Controlador

O programa do controlador funciona como um servo do programa principal através do programa de interface controlador, que serve de intermediário no processo. As funcionalidades do programa presente no controlador respondem, portanto ao programa de interface controlador, motivo pelo qual se encontram detalhadas no Anexo C as funcionalidades deste programa.

3.4 Calibração do Sistema

A calibração do sistema implementado engloba:(1) a calibração do(s) sensor(es) responsáveis pela obtenção da imagem de cores e mapa de profundidade (Microsoft Kinect para a Xbox360), assim como (2) obtenção da correspondência entre o sistema de coordenadas utilizado pelo controlador robótico e o sistema de coordenadas referente ao sensor de obtenção de mapa de profundidade e imagem de cores (Microsoft Kinect para a Xbox360). Estes referenciais encontram-se ilustrados na Figura 3.11.

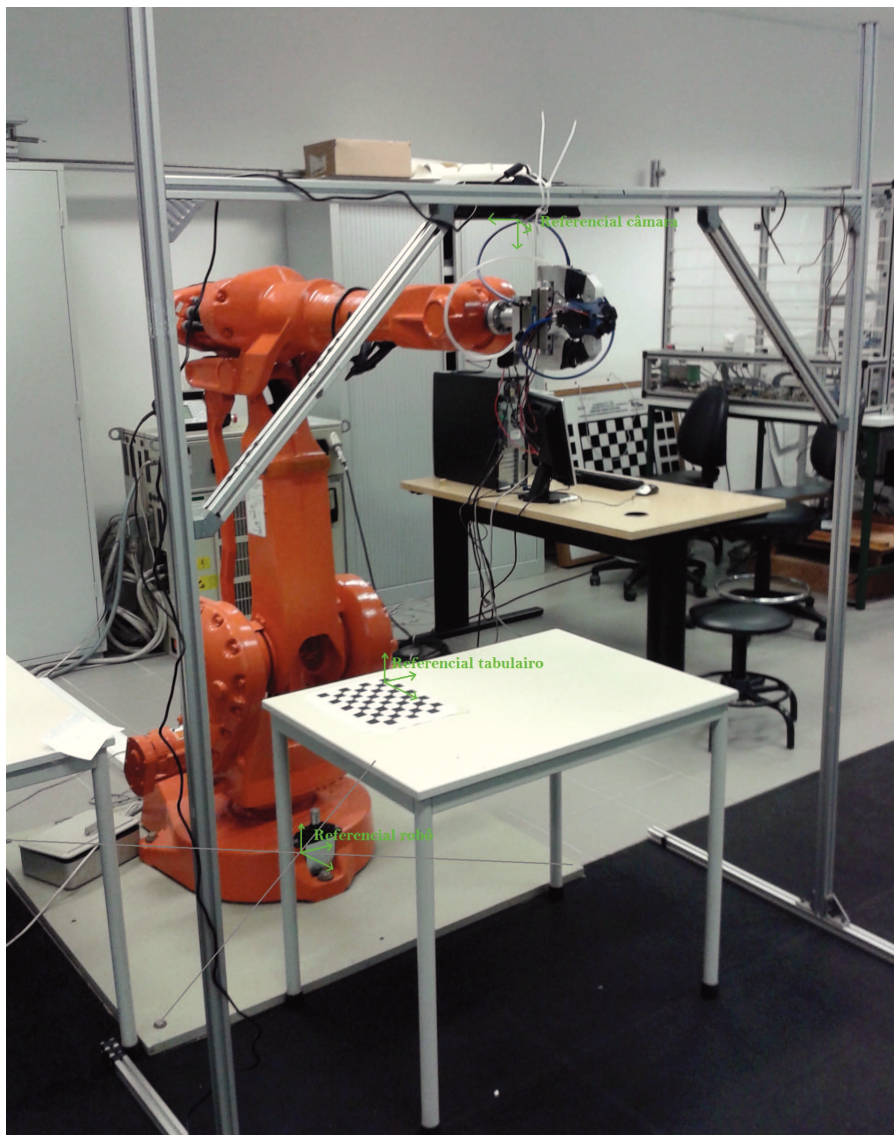


Figura 3.11: Referenciais do sistema.

3.4.1 Calibração da Microsoft Kinect

A calibração de câmaras visa estimar a distorção das imagens capturadas de forma a compensar ou corrigir os seus efeitos. A Kinect possui uma câmara de cores e uma câmara de infravermelhos. A câmara de infravermelhos é utilizada, em conjunto com um projetor de infravermelhos para obter um mapa de profundidade através de triangulação. Por este motivo, a calibração da câmara de infravermelhos encontra-se associada ao mapa de profundidade. Para conhecer a correspondência de píxeis de duas câmaras no mundo deve-se conhecer os respetivos parâmetros intrínsecos e extrínsecos. O processo de calibração encontra-se descrito no anexo B. Para a câmara de cores obteve-se o desvio padrão de 0,234 mm para os lado dos quadrados de 25mm de um padrão xadrez a uma distância de 82cm da câmara. Para a câmara de infravermelhos efetuou-se o mesmo teste obtendo-se o desvio padrão de 0,2455mm. Para avaliar a qualidade do mapa de profundidade com a câmara fixa, retiraram-se imagens de uma porção de uma mesa de 40cm de lado, a 112cm da câmara, tendo-se obtido o desvio padrão de 3,156mm. Na Figura B.7 pode ser visto o histograma da distância a plano para a calibração escolhida. Nas Figuras B.7 e B.9 pode-se observar a diferença entre a posição de pontos resultantes da conversão de pontos do mundo para a imagem com os parâmetros iniciais e com os parâmetros referentes à calibração, assim como o divergente da distância entre os pontos.

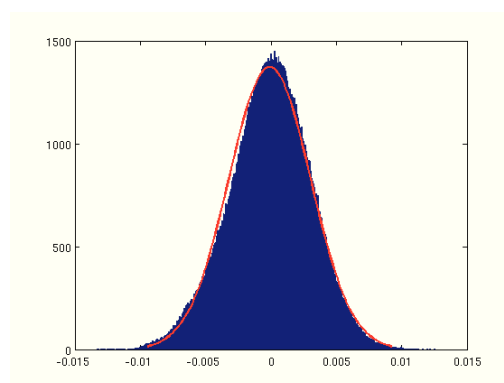
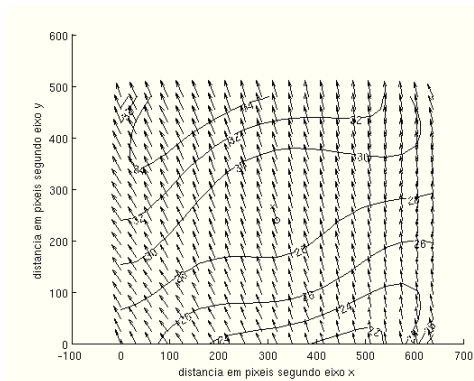
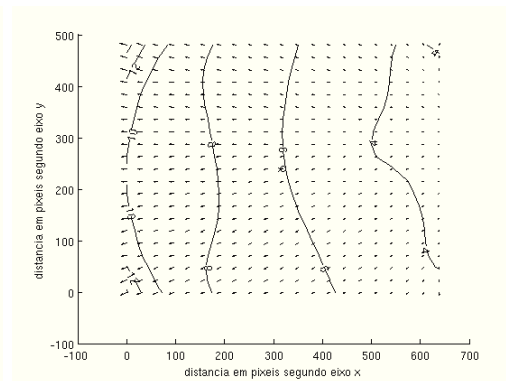


Figura 3.12: Histograma de distância de pontos a plano para a calibração 22.

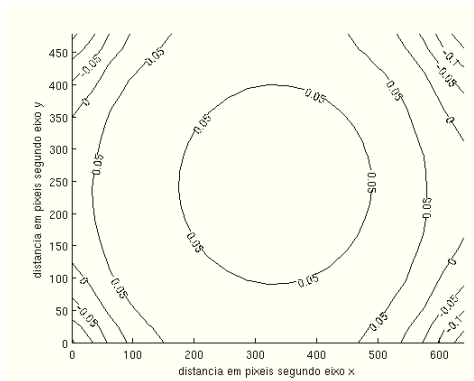


(a) câmara de cores

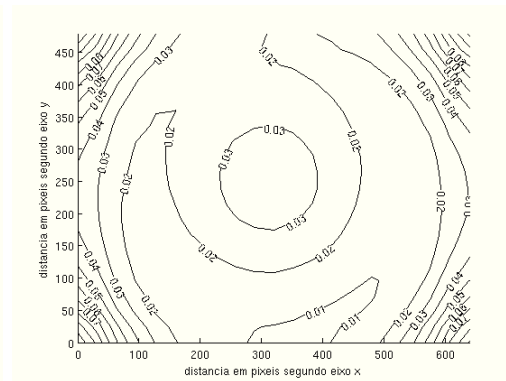


(b) câmara de infravermelhos

Figura 3.13: Diferença entre conversão de pontos com os parâmetros iniciais e parâmetros relativos à calibração.



(a) câmara de cores



(b) câmara de infravermelhos

Figura 3.14: Gradiente da diferença entre conversão de pontos com os parâmetros iniciais e parâmetros relativos à calibração.

3.4.2 Obtenção da correspondência entre os referenciais

Na aplicação existem três referenciais distintos, associados a: (1) imagem de cores, (2) mapa de profundidade e (3) manipulador robótico. A correspondência entre as duas primeiras encontra-se no Anexo B. Para encontrar a correspondência entre a primeira e a terceira, utiliza-se um quarto referencial cujo sistema de coordenadas se descreve nos sistemas de coordenadas associados à imagem de cores e manipulador robótico. É neste quarto referencial que se expressam as posições e orientações na aplicação implementada. A utilização deste referencial permite mover a Kinect entre testes, verificar a calibração do robô com a câmara desligada e facilita a arrumação de material ao longo do período de testes quando se afigura necessária.

A correspondência entre o referencial associado ao padrão xadrez e o referencial associado ao manipulador robótico compreende:

1. Movimentação da ferramenta até à origem do referencial constituído pelo padrão xadrez, registando o valor das coordenadas associadas ao manipulador robótico nesse ponto;
2. Mover ferramenta até aos cantos internos do padrão xadrez mais distantes do ponto que constitui a origem, segundo os versores \hat{x} e \hat{y} , registando os pontos;

Caso os versores \hat{x} , \hat{y} dos dois referenciais estejam alinhados, a transformação consiste apenas numa translação. Caso contrario, a transformação entre os dois referenciais consiste na multiplicação matricial expressa na Equação (3.1).

$$Ponto_1 = \begin{bmatrix} X_{2_{x_1}} & X_{2_{y_1}} & X_{2_{z_1}} & O_{2_{x_1}} \\ Y_{2_{x_1}} & Y_{2_{y_1}} & Y_{2_{z_1}} & O_{2_{y_1}} \\ Z_{2_{x_1}} & Z_{2_{y_1}} & Z_{2_{z_1}} & O_{2_{z_1}} \\ 0 & 0 & 0 & 1 \end{bmatrix} Ponto_2 \quad (3.1)$$

Para a correspondência entre o referencial associado ao padrão xadrez e o referencial associado à imagem de cores é necessário:

1. Encontrar padrão tipo xadrez, numa imagem da câmara de cores, identificando os pontos referentes aos seus cantos internos, obtendo-se as coordenadas x, y e z desse mesmo ponto recorrendo-se ao mapa de profundidade (com base na calibração da kinect);
2. O canto interno detetado no canto superior esquerdo da imagem corresponde à origem do referencial associado ao padrão xadrez. A diferença entre a origem e o canto interno superior direito corresponde ao versor \hat{x} . A diferença entre a origem e o canto interno inferior esquerdo corresponde a uma estimativa para o versor \hat{y} , denominado $\hat{y}_{estimado}$. Isto encontra-se ilustrado na Figura 3.15;
3. Calcular o versor \hat{z} através do produto externo entre os vetores \vec{x} e $\vec{y}_{estimado}$, garantindo que $\vec{z} \perp \vec{x} \wedge \vec{z} \perp \vec{y}_{estimado}$;
4. Calcular o versor \hat{y} através do produto externo entre os versores \hat{z} e \hat{x} , garantindo que $\vec{y} \perp \vec{x} \wedge \vec{y} \perp \vec{z}$;
5. Definir o sentido dos versores de acordo com o sentido destes no referencial robô;

6. Calcular a transformação entre pontos descritos no sistema de coordenadas associado à imagem de cores e a descrição equivalente do mesmo ponto no sistema de coordenadas do referencial associado à folha com padrão xadrez através da Equação (3.1).

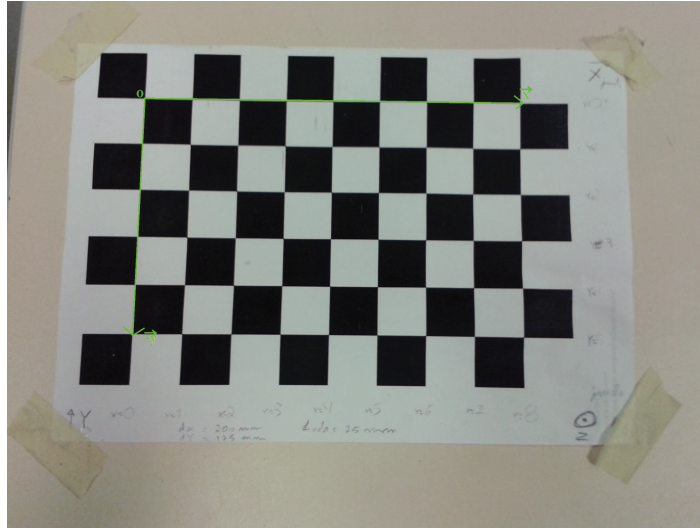


Figura 3.15: Vetores com direção x e y.

3.5 Estimativa de Pose

As peças são refletoras (ver secção 3.1), o que dificulta a deteção de profundidade através da triangulação ativa utilizada pela Kinect, verificando-se zonas nas quais a profundidade não é determinada. Para além disto, o erro médio associado a medições do mapa de profundidade é na ordem de grandeza da espessura das peças. Por estes motivos o mapa de profundidade não é utilizado como informação base para a discriminação entre peças ou aferição da pose. Os valores do mapa de profundidade são apenas utilizados como referência para a posição do objeto a agarrar. Considerando a incerteza associada ao mapa de profundidade e a dificuldade associada à identificação de características em imagens de peças refletoras e sem textura, como discutido em [13], optou-se pela divisão do processo de determinação da pose em duas fases. A primeira consiste na determinação da pose de peças na caixa apenas com o rigor necessário à captura de uma peça. Na segunda, deteta-se a pose da peça capturada, isolada das restantes, em fundo controlado, com o rigor necessário à correta introdução no sistema de destino com uma posição e orientação conhecidas. Caso esta determinação de pose não seja bem sucedida, a peça é rejeitada, considerando-se que a peça não corresponde ao modelo ou contém defeito(s).

3.5.1 Deteção Orientada à Captura de Peças

O algoritmo implementado é baseado na deteção de regiões planas. O sistema de deteção baseia-se em três componentes:(1) localização,(2) forma e (3) aspeto. A localização constitui uma simplificação das regiões a detetar e classificar, evitando-se regiões com forma ou aspeto que possam ser excluídas por não pertencerem à área de trabalho, mesmo que os parâmetros se aproximem dos modelados. Deste modo evita-se um esforço acrescido na parametrização de forma ou aspeto.

O sistema de deteção de peças, de um modo geral, compreende as seguintes tarefas:(1) segmentação e (2) classificação dos diferentes segmentos de acordo com a sua relevância ou interesse. Devido à incerteza associada à medição de profundidade optou-se por utilizar informação capturada pela câmara de cores para a segmentação. Procura-se que os segmentos correspondam a zonas de intensidade luminosa homogénea, encontrando-se limitados por zonas com variação de intensidade luminosa considerada elevada. A segmentação implementada está esquematizada no diagrama de blocos da Figura 3.16. A estimativa de arestas utiliza o detetor de arestas Canny[27] aplicado à intensidade luminosa de uma imagem. Para a execução deste detetor são

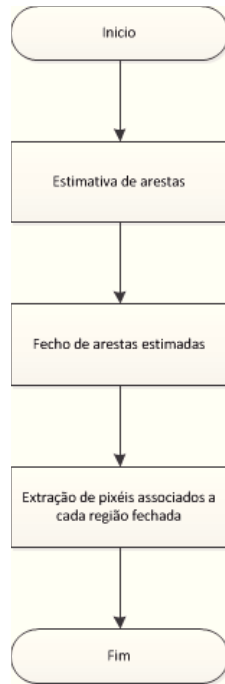


Figura 3.16: Fluxograma que descreve as etapas de obtenção de segmentos.

necessários dois valores limite. A determinação destes valores é efetuada de modo automático, seguindo o processo descrito em [56], considerando-se o gradiente da suavização obtida com a aplicação de um filtro gaussiano à imagem (com parâmetros iguais ao algoritmo Canny) em vez do gradiente da imagem. Esta alteração é um meio termo entre a abordagem [56] e a abordagem [57] que considera o processo de ligação entre pontos de limiar superior com os de limiar inferior utilizados no algoritmo Canny. De modo a encontrar os píxeis correspondentes a cada região é calculada a distância à aresta, consistindo na Equação (3.2). De seguida é iterativamente calculado o máximo desta grandeza e efetuada a marcação de toda a área dessa região fechada, repetindo-se o processo, sem considerar as regiões já marcadas, até que o valor máximo da distância à aresta seja inferior a um valor limite.

$$d_{aresta} = \underbrace{Max}_{\forall (x_{peça}, y_{peça}) \in peça} \underbrace{Min}_{\forall (x_{aresta}, y_{aresta}) \in aresta} \sqrt{(x_{peça} - x_{aresta})^2 + (y_{peça} - y_{aresta})^2} \quad (3.2)$$

Após a obtenção de regiões fechadas segue-se a análise. O cerne da análise da forma de regiões é a sua comparação com a forma da ventosa utilizada. Caso a região contenha uma zona de dimensão igual ou superior à da ventosa, a região pode ser capturada. Entre a deteção de um círculo da dimensão da ventosa e a deteção de uma peça completa, é adotado um sistema de classificação baseado nas dimensões da região. Deste

processo resulta a detecção de regiões capturáveis e discriminação positiva das regiões mais próximas de uma peça completa, sendo ignoradas regiões de dimensão superior. Segue-se uma explicação detalhada da classificação da forma de uma região.

Considere uma região um conjunto conexo de píxeis. Considere ainda que: (1) "Região(Peça)" é uma região, possivelmente virtual, constituída por todos os píxeis que pertencem a uma mesma peça (visíveis ou não); (2) "Região(Segmento)" é uma região em que todos os pontos se encontram visíveis e pertencem a uma mesma peça e (3) "Região(captura)" um segmento com dimensão suficiente para suportar a recolha. Uma região associada a um segmento s_2 está contida em outra, associada a um segmento s_1 , caso verifique a expressão 3.3. Com base nela, pode-se entender que a região associada ao s_2 é composta por todos os elementos de área pertencentes à região s_1 , podendo ou não conter elementos de área adicionais, o que implica que uma qualquer região associada a um segmento genérico se contém a si mesma, $Região(s_1) \subseteq Região(s_1) \forall s_1$. Esta operação pode ser visualizada com o auxílio da Figura 3.17. Nesta Figura os quadrados a branco representam elementos de uma região. Os quadrados cinzentos representam elementos cuja adição à "região branca" satisfaz a condição de conectividade e a Equação 3.3. A região constituída por todos os quadrados brancos e um quadrado cinzento contém a região formada por todos os quadrados brancos. A adição de um quadrado preto à "região branca" não satisfaz a condição de conectividade e a Equação 3.3. O conjunto obtido com esta operação não constitui uma região.

$$Região(s_1) \subseteq Região(s_2) \implies \forall s_1, s_2 \exists p_{s_2} : p_{s_2} \in Região(s_1) \quad (3.3)$$

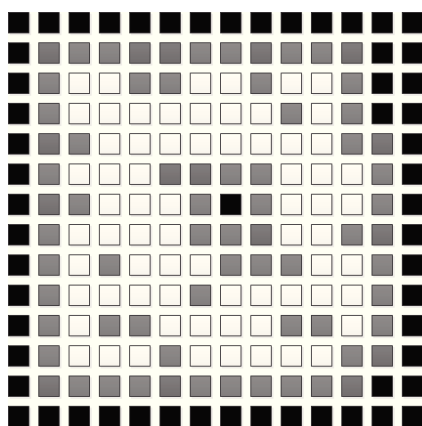


Figura 3.17: Exemplo de uma região.

A forma associada a uma região deve ser tal que a sua superfície contenha a forma necessária para a correta captura da peça. Devido à ferramenta possuir uma ventosa

redonda para este fim, a região deve conter um círculo de raio igual ou superior ao raio da ventosa. A superfície associada a uma região deve conter a forma necessária para que seja expetável que pertença a uma peça. Devido a sobreposições, partes de peças podem não se encontrar visíveis, podendo-se apenas aferir se a expetativa de a forma corresponder a uma parte da peça é provável. Considera-se que uma região cuja forma associada respeite estes princípios constitui uma região de interesse.

A abordagem à deteção da forma da peça encontra-se resumida na Equação (3.4), considerando-se que uma região possui interesse caso esta expressão seja satisfeita.

$$Região(peça) \subseteq Região(segmento) \subseteq Região(captura) \quad (3.4)$$

A classificação da forma consiste na utilização de indicadores referentes à Equação (3.4) que garantem, de modo estrito, que a região associada à captura está contida na região do segmento. A ausência de um elemento do círculo da região associada à captura de peças torna um segmento "inválido". Não se garante de modo estrito que a região associada ao segmento está contida na região associada à peça. Simplifica-se, com o auxílio dos indicadores referentes à Equação (3.4), a verificação desta condição, com o intuito de aumentar a robustez da classificação³. Os indicadores utilizados, para um segmento s_i , são:(1) Área " $a(s_i)$ ", (2) Máxima distância a aresta mais próxima " $d_{aresta}(s_i)$ ", (3) Menor lado " $l(s_i)$ " do retângulo de menor área que englobe a região e (4) Maior lado " $c(s_i)$ " do retângulo de menor área que englobe a região . O indicador " a " fornece informação sobre a dimensão da região. O valor associado a " a " é sensível a qualquer sobreposição a uma região. A " d_{aresta} " visa assegurar que a região associada a um segmento contém a região associada à captura, isto porque o seu valor depende da ausência de arestas na sua proximidade. A margem de segurança para a captura de um objeto corresponde à diferença entre o valor de " d_{aresta} " e o raio da ventosa⁴. A " d_{aresta} " é sensível a sobreposições no círculo de raio igual " d_{aresta} " da zona mais segura para a ventosa agarrar. A forma associada às peças é aproximadamente retangular. Os indicadores " c " e " l " visam complementar a descrição da forma associada à região, fornecendo dimensões segundo duas direções perpendiculares. Os indicadores " c " e " l " são sensíveis a sobreposições que atravessem completamente a região paralelamente a um dos lados do retângulo que engloba a região. Também são sensíveis a oclusões de pontos pertencentes a vértices ou arestas do polígono convexo que engloba a região.

³deste modo permitem-se pequenas deformações associadas a ruído, sujidade ou reflexões na forma da peça.

⁴se " d_{aresta} " for 1cm maior que o raio da ventosa então com um desvio até 1cm do ponto de maior distância a aresta é possível, efetuar uma captura com sucesso

As métricas " d_{aresta} ", " c " e " l " encontram-se ilustradas na Figura 3.18.

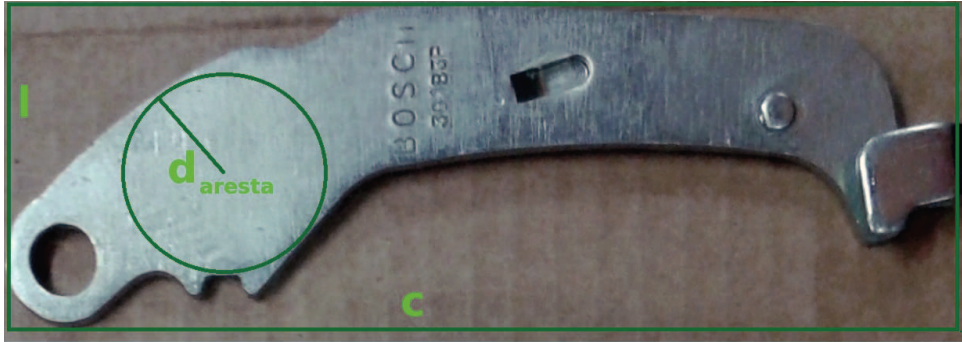


Figura 3.18: Ilustração das métricas utilizadas sobre uma peça.

Conclui-se, com base na Equação (3.5), que, para uma sequência de n segmentos s_1, s_2, \dots, s_n , na qual $Região(s_{i+1}) \subseteq Região(s_i) \forall i \in \mathbb{N}^*$, com $i \leq n$, representando: (1) a região correspondente a s_1 a região associada à captura e (2) a região correspondente a s_n a região associada à peça. Este conceito encontra-se ilustrado na Figura 3.19. Os valores associados aos indicadores: área, máxima distância à aresta mais próxima, assim como os limites⁵ superiores (*diâmetro de círculo de menor área que contém o segmento* $\geq c \geq l$) e inferiores ($c \geq l \geq 2 * d_{aresta}$) associados aos lados do retângulo de menor área que englobe a região, apenas progredirem no sentido crescente, o que faz das métricas boas estimativas da proximidade entre as regiões associadas a um segmento arbitrário s_i e as regiões associadas à captura da peça s_1 e forma da peça s_n . Apesar das métricas associadas aos lados do retângulo de menor área que engloba a região não estarem condicionadas a progredir apenas no sentido positivo, somente uma pode diminuir entre cada uma das iterações da sequência considerada, encontrando-se a magnitude da diminuição limitada quer pela dimensão do aumento do lado adjacente, quer pelos limites superiores e inferiores destas métricas. No Anexo D pode-se consultar a argumentação que permite deduzir a Equação (3.5).

$$Região(s_1) \subseteq Região(s_2) \implies \begin{cases} a(s_1) \geq a(s_2) \\ d_{aresta}(s_1) \geq d_{aresta}(s_2) \\ l(s_1) \geq l(s_2) \vee c(s_1) \geq c(s_2) \\ l(s_1) * c(s_1) \geq l(s_2) * c(s_2) \end{cases} \quad (3.5)$$

Utilizando a câmara de cores, para cada segmento obtido, apenas se tem acesso a uma projeção da superfície. Da determinação das métricas (a , d_{aresta} , c e l) na projeção

⁵ver Anexo D

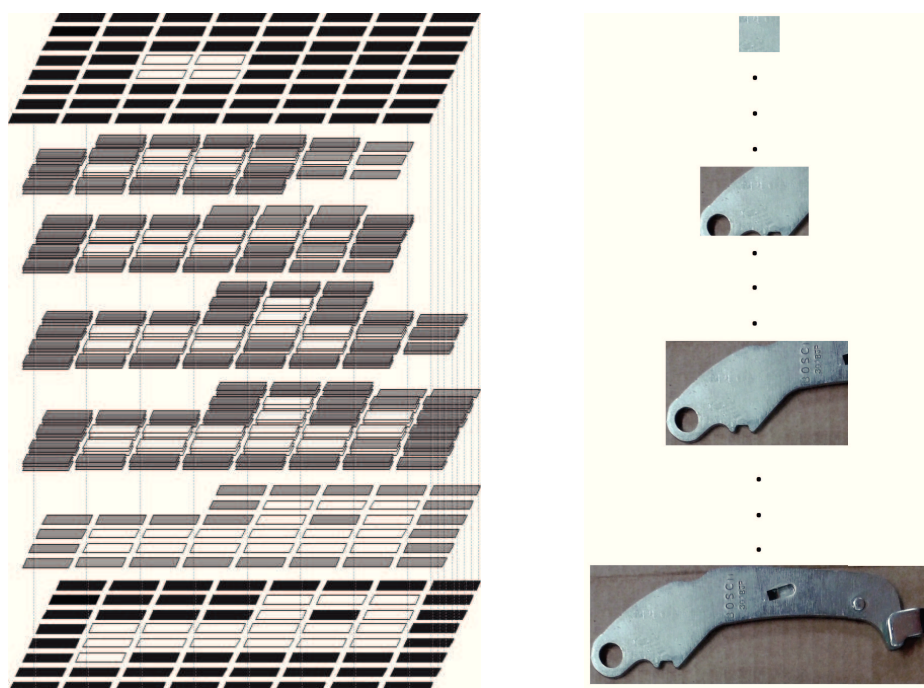


Figura 3.19: Ilustração do conceito de uma seqüência de segmentos obtidos incrementalmente, com a adição de um novo elemento.

de uma peça, podem resultar valores que diferem significativamente dos obtidos com medições ao longo do plano da peça (correspondente à projeção no caso em que a normal da superfície da peça e a normal do plano da câmara possuem a mesma direção). Na Figura 3.20 pode-se visualizar um caso onde a projeção dos contornos de uma peça inclinada para o plano da câmara diminui as métricas utilizadas (a , d_{aresta} , c e l). É possível obter uma estimativa da depreciação que a operação de projeção provoca em cada um dos indicadores, conhecendo-se a relação entre o vetor normal associado à superfície e à câmara. Não se considera a informação relativa ao mapa de profundidade na detecção associada à captura de peças, devido ao erro associado a uma medição ser próximo da espessura das peças e à presença de zonas com valores inválidos. Por este motivo, sendo a inclinação associada às peças um fator considerado não favorável, não é efetuada a retificação associada à expetável depreciação dos indicadores. Entende-se assim que este fator corresponde a um elemento não considerado de modo explícito, tornando o resultado do algoritmo sensível a operações de rotação da região face ao sensor utilizado para a sua captura (câmara).

O aspeto pretende complementar a descrição de regiões, servindo de indicador para a remoção de regiões cujas dimensões tornam provável a associação a peças. Considera-se que uma região contém aspeto semelhante a uma peça no espaço de cor YUV, a média da sua cor, segundo cada um dos canais associados à cor, esteja à distância máxima de um desvio padrão, da cor média associada a uma peça. Quanto maior o desvio

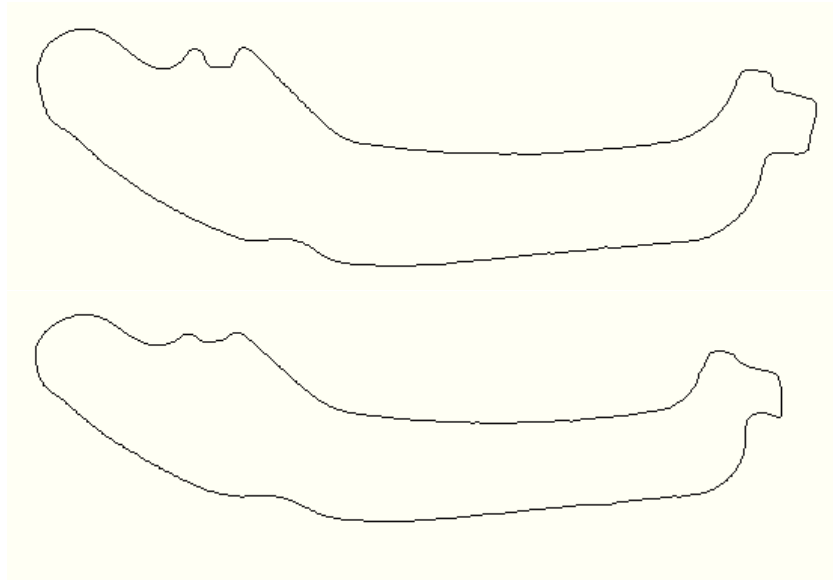


Figura 3.20: Influência da inclinação do vetor normal de uma superfícies na estimativa de métricas numa imagem bidimensional.

padrão da região maior a diferença permitida entre as médias. De modo a limitar a margem considerada, estipula-se um desvio padrão máximo associado a uma região, considerando que, se o desvio padrão de uma região for maior que este valor, a região não possui aspeto semelhante ao de uma peça. Para casos com desvio padrão muito baixo deve ser considerada uma diferença mínima entre as médias pois, no limite em que o desvio padrão seja zero, as médias associadas à peça e à região devem coincidir exatamente.

O modelo de classificação de regiões de interesse pretende ordenar as regiões detetadas de acordo com: (1) a potencialidade de correspondência a peças, (2) possibilidade de serem agarradas com a ferramenta considerada (ventosa), (3) facilidade de se extrair a possível peça da caixa, tendo em conta aspetos como sobreposição .

Para tal, a classificação engloba modelos de:(1) câmara, (2) montagem física do sistema e (3) peças. O modelo da câmara, neste contexto, engloba os parâmetros necessários para conversão entre coordenadas na imagem, depois de retificada, e coordenadas no mundo com origem de referencial na câmara, ver Equações (B.5) e (B.6). O modelo da montagem física do sistema, neste contexto engloba:

1. Zona de interesse⁶ com forma de paralelepípedo $Zona_{Interesse}$: (a) Profundidade mínima e máxima ao local de deteção de peças ou regiões de interesse z_{min} z_{max}
 (b) Zona retangular do mundo onde se espera encontrar peças x_{min} x_{max} y_{min}

⁶definida no referencial ortonormado com versores \hat{x} \hat{y} \hat{z} , onde se verifica que $\hat{z} = \hat{x} \times \hat{y}$

y_{max}

2. Raio da ventosa a utilizar: $r_{ventosa}$

O modelo das Peças, engloba:

1. Aspeto da imagem:

- (a) cor média esperada $\mu_{cor_{modelo}}$;
- (b) desvio padrão mínimo esperado $\sigma_{cor_{min}}$;
- (c) desvio padrão máximo esperado $\sigma_{cor_{max}}$;

2. Forma da peça:

- (a) Área da região detetada a_{max} ;
- (b) Valor máximo para maior distância à aresta mais próxima $d_{aresta_{max}}$ (ver Equação (3.2));
- (c) Comprimento máximo para o maior lado do retângulo de menor área que engloba a região detetada c_{max} ;
- (d) Comprimento máximo para o menor lado do retângulo de menor área que engloba a região detetada l_{max} .

Através dos sensores do sistema (Kinect) obtêm-se as medições:

1. Aspeto da imagem:

- (a) cor média de uma região μ_{cor} ;
- (b) desvio padrão σ_{cor} ;

2. Zona de Deteção da Imagem (em relação à câmara):

- (a) profundidade z ;
- (b) normal \vec{n} ;

3. Forma da imagem:

- (a) Área a ;
- (b) Máxima distância a aresta mais próxima d_{aresta} ;
- (c) Menor lado do retângulo de menor área que englobe a região l ;

(d) Maior lado do retângulo de menor área que englobe a região c .

4. ponto para captura da peça pela ventosa \vec{p} , correspondendo ao ponto para o qual a menor distância a aresta é máxima.

O processo de classificação começa com uma etapa de distinção entre regiões de interesse e regiões irrelevantes. Não existe nesta classificação nenhuma ambiguidade devido a apenas se considerar dois estados para classificação de uma região (Equação (3.6)). Uma região é considerada de interesse caso satisfaça as Equações (3.7), (3.8), (3.9), (3.10) e (3.10).

$$Região_{Interesse} = \neg Região_{Irrelevante} \quad (3.6)$$

$$\vec{p} \in Zona_{Interesse} \implies \begin{cases} x_{min} \leq \vec{p} \cdot \hat{x} \leq x_{max} \\ y_{min} \leq \vec{p} \cdot \hat{y} \leq y_{max} \\ z_{min} \leq \vec{p} \cdot \hat{z} \leq z_{max} \end{cases} \quad (3.7)$$

$$\pi r_{ventosa}^2 \leq a \leq a_{max} \quad (3.8)$$

$$r_{ventosa} \leq d_{aresta} \leq d_{aresta_{max}} \quad (3.9)$$

$$2r_{ventosa} \leq l \leq l_{max} \quad (3.10)$$

$$2r_{ventosa} \leq c \leq c_{max} \quad (3.11)$$

Apenas as regiões de interesse são consideradas nas etapas seguintes. Segue-se a normalização dos parâmetros associados à localização e à forma, em relação aos valores máximos e mínimos que podem tomar, através das Equações (3.12) , (3.14) ,(3.15), (3.16),(3.17) ou a valores convenientes, através da Equação (3.13), onde se consideram apenas peças com rotação inferior a 45 graus segundo uma direção perpendicular ao eixo z ($\cos(45^\circ) = \sqrt{2}/2$). Devido às considerações necessárias para uma região se afigurar de interesse, garante-se que estes valores se encontram entre 0 e 1. O valor 1 corresponde ao melhor caso, e o valor 0 ao pior caso. Quanto mais próximo de 1 melhor

a correspondência entre a grandeza medida e o valor correspondente ao modelo.

$$z_n = \frac{z_{max} - z}{z_{max} - z_{min}} \quad (3.12)$$

$$n_n = \begin{cases} \frac{\vec{n} \cdot \hat{z} - \sqrt{2}/2}{1 - \sqrt{2}/2} & \text{se } \frac{\sqrt{2}}{2} \leq \vec{n} \cdot \hat{z} \leq 1 \\ 0 & \text{caso contrário} \end{cases} \quad (3.13)$$

$$a_n = \frac{a - \pi r_{ventosa}^2}{a_{max} - \pi r_{ventosa}^2} \quad (3.14)$$

$$d_{aresta_n} = \frac{d_{aresta} - r_{ventosa}}{d_{aresta_{max}} - r_{ventosa}} \quad (3.15)$$

$$c_n = \frac{c - 2r_{ventosa}}{c_{max} - 2r_{ventosa}} \quad (3.16)$$

$$l_n = \frac{l - 2r_{ventosa}}{l_{max} - 2r_{ventosa}} \quad (3.17)$$

Com base nos valores normalizados, calcula-se um valor de referência para a localização e outro para a forma. Estes valores encontram-se normalizados entre 0 e 1, correspondendo 1 ao melhor caso e 0 ao pior. Para o valor de referência associado à localização, confere-se maior relevância à altura devido a este fator oferecer a possibilidade de evitar apanhar objetos com sobreposições. Considera-se que a inclinação de uma região é muito menos relevante que a sua altura, razão pela qual se atribui maior sensibilidade à altura no cálculo do *indicador*_{localização}, considerando $peso_1 \geq peso_2$ na Equação (3.18).

$$indicador_{localização} = \frac{peso_1 * z_n + peso_2 * n_n}{peso_1 + peso_2} \quad (3.18)$$

Para o cálculo do indicador associado à forma é atribuída mais relevância à métrica A_n , devido à maior sensibilidade desta. Das restantes, a mais relevante é a métrica D_n , devido ao seu valor configurar a margem de segurança para a correta captura da peça. Obtêm-se assim a relação de compromisso $peso_3 \geq peso_4 \geq peso_5 \wedge peso_4 \geq peso_6$ para o cálculo do *indicador*_{forma} através da Equação (3.19).

$$indicador_{forma} = \frac{peso_3 * a_n + peso_4 * d_{aresta_n} + peso_5 * l_n + peso_6 * c_n}{peso_3 + peso_4 + peso_5 + peso_6} \quad (3.19)$$

A classificação referente ao aspeto baseia-se na análise de imagens com os canais de cor: (1) u e (2) v . A classificação é, numa primeira etapa, calculada para cada um dos canais de cor em separado, sendo posteriormente aglomeradas num indicador único.

A classificação referente ao aspeto pretende garantir, em cada canal de cor, que a cor média associada ao modelo se encontra próxima da cor média da peça, medindo-se esta proximidade em quantidades de desvio padrão (ver Equação 3.20). Obtém-se com isto a situação de qualquer cor ser aceitável no caso σ_{cor} elevado ou nenhuma cor ser aceitável σ_{cor} próximo de 0. Para contornar estes inconvenientes, considera-se um valor mínimo para σ_{cor} e recorre-se à Equação (3.21), que é utilizada juntamente com a Equação (3.20), para produzir a métrica 3.22. Este valor encontra-se limitado entre 0 e 1, correspondendo 1 ao caso mais favorável. Obtendo-se esta classificação para cada um dos canais, procede-se ao cálculo do indicador baseado no aspeto 3.23, garantindo que o resultado não é melhor que o dobro da pior classificação. Obtém-se assim um valor adimensional e limitado entre 0 e 1, correspondendo 1 ao caso mais favorável.

$$\mu_{cor_{err}} = \begin{cases} \left| \frac{\mu_{cor} - \mu_{cor_{modelo}}}{\sigma_{cor}} \right| & \text{se } \left| \frac{\mu_{cor} - \mu_{cor_{modelo}}}{\sigma_{cor}} \right| \leq 1 \\ 1 & \text{caso contrário} \end{cases} \quad (3.20)$$

$$\sigma_{cor_{err}} = \begin{cases} 0 & \text{se } \sigma_{cor_{min}} \leq \sigma_{cor} \leq \sigma_{cor_{max}} \\ \frac{\frac{\sigma_{cor_{min}} + \sigma_{cor_{max}}}{2} - \sigma_{cor}}{\frac{\sigma_{cor_{min}} + \sigma_{cor_{max}}}{2}} & \text{caso contrário} \end{cases} \quad (3.21)$$

$$\mu\sigma_{cor} = |1 - cor_{\mu_{err}}| |1 - cor_{\sigma_{err}}| \quad (3.22)$$

$$indicador_{aspeto} = \begin{cases} 0 & \text{se } \mu\sigma_{cor_u} = 0 \vee \mu\sigma_{cor_v} = 0 \\ 2 \frac{\mu\sigma_{cor_u} * \mu\sigma_{cor_v}}{\mu\sigma_{cor_u} + \mu\sigma_{cor_v}} & \text{caso contrário} \end{cases} \quad (3.23)$$

Obtém-se, assim, o vetor de estado 3.24, a partir do qual se obtém o vetor de indicadores de estado 3.25.

$$E = [z_n, n_n, a_n, d_{aresta_n}, l_n, c_n, \mu_{cor_u}, \sigma_{cor_u}, \mu_{cor_v}, \sigma_{cor_v}, \vec{p}] \quad (3.24)$$

$$I_E = \begin{bmatrix} indicador_{localização} \\ indicador_{forma} \\ indicador_{aspeto} \end{bmatrix} \quad (3.25)$$

Com recurso aos vários indicadores pretende-se calcular um valor de referência $indicador_{ordem}$ cujo objetivo é a ordenação das regiões identificadas. Para tal é necessário estabelecer relações de compromisso entre as grandezas expostas no vetor de indicadores de estado 3.25. Devido à consideração de regiões de interesse já acautelar a possibilidade de uma região ser agarrada pela ventosa, a classificação deve procurar, em primeiro lugar, garantir que a região a apanhar pode ser levantada. Uma peça que não possa ser manipulada não possui interesse por parte do sistema, pelo que aferir a possibilidade de uma região pertencer a uma peça constitui uma segunda prioridade. O critério associado à possibilidade de se levantar uma peça é a ausência de sobreposição por parte de outras peças. Neste contexto é considerado mais relevante o indicador associado à localização. Considera-se que o indicador associado ao aspeto é muito menos relevante que os restantes por não conter informação diretamente associável a sobreposições. Com base nisto, estabelece-se a relação de compromisso $peso_{localização} > peso_{forma} > peso_{aspeto}$ para o cálculo do $indicador_{ordem}$ através da Equação 3.26. Após o cálculo do $indicador_{ordem}$, ordenam-se as deteções da melhor para a pior de acordo com o indicador $indicador_{ordem}$. A primeira deteção (considerada a melhor) deve ser a utilizada numa tentativa de captura.

$$indicador_{ordem} = \begin{bmatrix} \frac{peso_{localização}}{peso_{localização} + peso_{aspeto} + peso_{forma}} \\ \frac{peso_{forma}}{peso_{localização} + peso_{aspeto} + peso_{forma}} \\ \frac{peso_{aspeto}}{peso_{localização} + peso_{aspeto} + peso_{forma}} \end{bmatrix}^T \begin{bmatrix} indicador_{localização} \\ indicador_{forma} \\ indicador_{aspeto} \end{bmatrix} \quad (3.26)$$

Devido à presença de ruído associado às imagens lidas, esperam-se diferentes resultados para uma sequência de diferentes imagens de um ambiente. Exemplos incluem a intermitência de segmentos de aresta, o que se reflete em diferenças no indicador associado à forma de segmentos, podendo resultar na intermitência de deteções de segmentos.

Por se considerar que o ambiente se mantém inalterado entre tentativas de captura de

peças, criou-se o conceito de histórico de detecções de modo a minimizar as oscilações referentes à identificação de segmentos e indicadores correspondentes. A correspondência entre regiões é o componente base para a produção de histórico de detecções.

Considerando duas regiões a e b , a correspondência implementada, que se encontra ilustrada na Figura 3.21, consiste na verificação dos pontos de maior distância à aresta se encontrem dentro do retângulo que contém o segmento oposto e possui lados paralelos aos lados da imagem da câmara. Apesar de ser desejável que os pontos de maior distância a aresta das duas detecções se encontrem a uma distância inferior ao raio da ventosa, não se considera esta informação na determinação de correspondência devido à possibilidade de se encontrar em várias regiões correspondentes a círculos de raio igual ao raio da ventosa contidas na região associada a uma peça. No caso de um segmento pertencer ao histórico, os valores a utilizar na comparação correspondem aos valores médios.

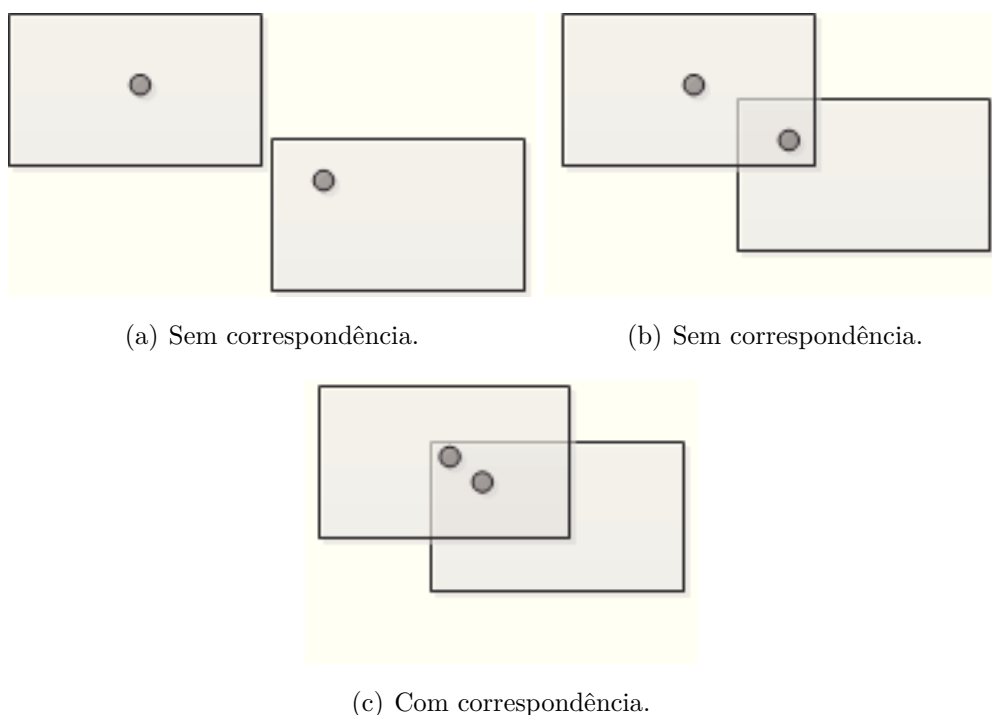


Figura 3.21: Correspondência entre regiões.

O histórico de detecções inclui, para cada região, a média e o desvio padrão de \vec{p} e dos indicadores referentes à localização, forma, aspeto e ordenação. Inclui ainda a quantidade de iterações, desde sua inclusão no histórico nas quais o segmento foi detetado, a quantidade de iterações nas quais o segmento não foi detetado e a quantidade de iterações desde a sua última detecção. O histórico de detecções possui as operações: (1) atualização, (2) união de regiões, (3) eliminação de regiões e (4) ordenação.

Estas operações são executadas sequencialmente.

A operação de atualização compreende a verificação de correspondências entre as regiões detetadas e as já contidas no histórico, a atualização dos valores de cada componente do histórico e a criação de novas regiões, no caso de não ser encontrada nenhuma correspondência entre deteções e histórico.

A união de regiões compreende a verificação de correspondência entre regiões pertencentes ao histórico, substituindo-se pares de regiões correspondentes por uma nova região pertencente ao histórico com informação correspondente às duas regiões.

A eliminação de regiões visa atribuir ao sistema a possibilidade de recuperar da introdução de falsos positivos, deteções ténues com valores próximos dos mínimos e deteções com indicadores instáveis face à sua magnitude média. A estas regiões associa-se muito baixa percentagem de deteção, com baixa média e elevado desvio padrão associados aos indicadores de forma e aspeto. O valor utilizado como referência para a ordenação de regiões é a diferença entre a média e o desvio padrão de *indicador_{ordem}*. Pretende-se valorizar a estabilidade dos indicadores, assumindo o desvio padrão o atributo de penalização.

A ordenação compreende duas fases, a primeira visa ordenar todos os valores, à exceção do primeiro, do maior para o menor. A segunda possui histerese e visa comparar o primeiro valor aos seguintes, apenas alterando a sua ordem caso a diferença entre valores seja maior que um valor limiar. Deste modo pretende-se estabilizar o resultado obtido, correspondente à primeira região do histórico de deteções, ao longo das sucessivas iterações.

3.5.2 Deteção de Pose de uma Peça

Após a captura de uma peça, pretende-se conhecer a sua pose com rigor suficiente para a sua correta colocação numa posição e orientação definidos. Esta determinação de pose apenas utiliza a informação referente à câmara de cor devido ao já referido erro associado ao mapa de profundidade. A determinação de pose inicia-se quando a peça se encontra presente num ambiente de fundo controlado, ilustrado na Figura 3.22, com uma posição conhecida face à câmara de cores e consistem em: (1) determinação de píxeis pertencentes à peça, (2) conversão dos píxeis associados à peça para o referencial do mundo, (3) cálculo de estimativa de transformação para alinhamento entre peça e

modelo⁷ e (4) cálculo de transformação entre peça e modelo.



Figura 3.22: Peça num ambiente de fundo controlado.

Devido à superfície controlada ser escura, para distinguir a peça do fundo recorre-se à intensidade luminosa de cada pixel. Para uma peça capturada, devido ao algoritmo implementado e ferramenta utilizada, determina-se a zona da imagem associada à ventosa contém uma região da peça constituída por um círculo de raio igual ao raio da ventosa. Esta informação é utilizada como ponto de partida para a determinação de píxeis pertencentes à peça, procurando-se o valor mínimo de intensidade luminosa de um pixel que corresponda a uma peça devido à sua localização na imagem. A distinção de píxeis correspondentes à peça consiste na comparação com 80% do valor mínimo considerado, e possui duas etapas:(1) identificação de píxeis de intensidade superior a um valor limite e(2) filtrar píxeis identificados de modo a obter um segmento de píxeis ligados, sendo pelo menos um destes pertencente à zona da imagem na qual é expectável que se encontre uma peça.

A transformação dos píxeis associados à peça para o referencial mundo é efetuado através das Equações (B.1) e (B.6), onde, para a altura da peça, se considera a distância conhecida entre a ventosa e a câmara. O cálculo da estimativa de transformação entre a peça e o modelo é constituído por :(1) estimativa de translação, seguida de (2) estimativa de rotação. Para obter a estimativa de translação calcula-se o centroide da região associada à peça $\overrightarrow{centroide}_{peça}$. A estimativa de translação entre a peça e o modelo consiste na diferença entre o centroide de ambas. Devido ao modelo ser construído de forma a que o seu centroide se encontre na origem do referencial mundo, a translação segundo cada um dos versores, consiste em $-\overrightarrow{centroide}_{peça}$. A estimativa de rotação é efetuada com base nos momentos centrais de segunda ordem (matriz de covariância). Calculam-se a partir da matriz de covariância os vetores próprios, ilustrados na Figura 3.23. \overrightarrow{auto}_1 , \overrightarrow{auto}_2 e \overrightarrow{auto}_3 e os valores próprios a eles associados $autoval_1$, $autoval_2$ e

⁷Obtido de forma semelhante à deteção de peças com vista à determinação de pose, sendo aplicada uma transformação que visa alinhar o centroide com a origem, \overrightarrow{auto}_1 com o eixo \vec{x} e \overrightarrow{auto}_2 com o eixo \vec{y} .

$autoval_3$, sendo $autoval_1 \geq autoval_2 \geq autoval_3$. A rotação consiste no alinhamento entre os versores \hat{x}, \hat{y} e \hat{z} e os vetores $\overrightarrow{auto_1}, \overrightarrow{auto_2}$ e $\overrightarrow{auto_1} \times \overrightarrow{auto_2}$. Devido ao modelo ser construído de forma a que esta rotação constitua a matriz identidade, para alinhar a peça ao modelo, a estimativa de rotação após a translação de alinhamento de centroides corresponde à equação 3.27.

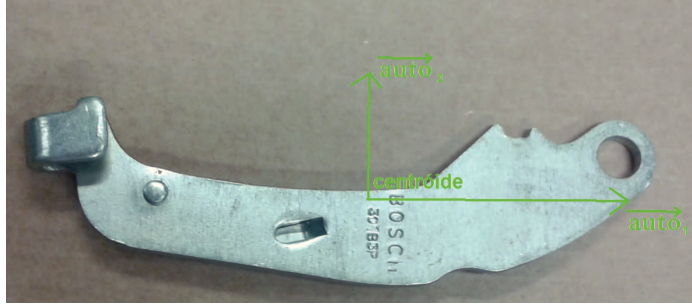


Figura 3.23: Vetores próprios e centroide numa peça.

$$matriz_{rotação} = \begin{bmatrix} \overrightarrow{auto_1} \cdot \hat{x} & \overrightarrow{auto_1} \cdot \hat{y} & \overrightarrow{auto_1} \cdot \hat{z} \\ \overrightarrow{auto_2} \cdot \hat{x} & \overrightarrow{auto_2} \cdot \hat{y} & \overrightarrow{auto_2} \cdot \hat{z} \\ \overrightarrow{auto_1} \times \overrightarrow{auto_2} \cdot \hat{x} & \overrightarrow{auto_1} \times \overrightarrow{auto_2} \cdot \hat{y} & \overrightarrow{auto_1} \times \overrightarrow{auto_2} \cdot \hat{z} \end{bmatrix} \quad (3.27)$$

O cálculo de transformação entre a peça e o modelo compreende uma simplificação da região associada a uma peça na qual se extraem apenas os seus contornos exteriores. De seguida é utilizado um algoritmo do tipo ICP[64] (podendo ser utilizadas variações propostas por [74],[66]), com quatro inicializações:(1) estimativa de transformação, (2) estimativa de transformação com rotação de 180 graus segundo $\overrightarrow{auto_1}$, (3) estimativa de transformação com rotação de 180 graus segundo $\overrightarrow{auto_2}$ e (4) estimativa de transformação com rotação de 180 graus segundo $\overrightarrow{auto_3}$. Com estas quatro inicializações pretende-se obter robustez face à reflexão perceptível pela câmara de cores, correspondente a uma rotação de 180 graus segundo um versor perpendicular à normal da câmara, assim como independência face ao sentido dos vetores próprios encontrados, ultrapassando-se possíveis mínimos do algoritmo tipo ICP. Utiliza-se a transformação (de entre as quatro possíveis) com menor erro quadrático médio. De seguida, por forma a minimizar o conjunto de situações em que a convergência consiste num mínimo local, verifica-se se, com uma pequena translação da transformação obtida segundo o eixo correspondente a $\overrightarrow{auto_1}$, se obtém um melhor resultado, utilizando-se o melhor valor com um novo teste segundo o eixo correspondente a $\overrightarrow{auto_2}$. Com base no erro quadrático médio da transformação obtida, afere-se se a peça corresponde ao modelo, ou não (podendo conter defeitos).

3.6 Controlo do Processo

O controlador funciona como servo do computador, executando ordens emitidas por este. A comunicação entre o controlador e computador foi implementada tendo em conta: (1) modularidade, (2) expansibilidade e (3) robustez. A implementação de funções associadas ao envio e receção de informação, as funções de conversão de tipos de dados e as funções relativas à interpretação dos dados são independentes entre si. Isto facilita a substituição de tecnologia de comunicação ou de controlador com diferentes definições de tipos de dados associado. O sistema resultante é robusto face a interrupções na comunicação quer no computador quer no controlador, assim como a perda ou erros de informação (ver anexo C). O processo de comunicação encontra-se esquematizado na Figura 3.24.

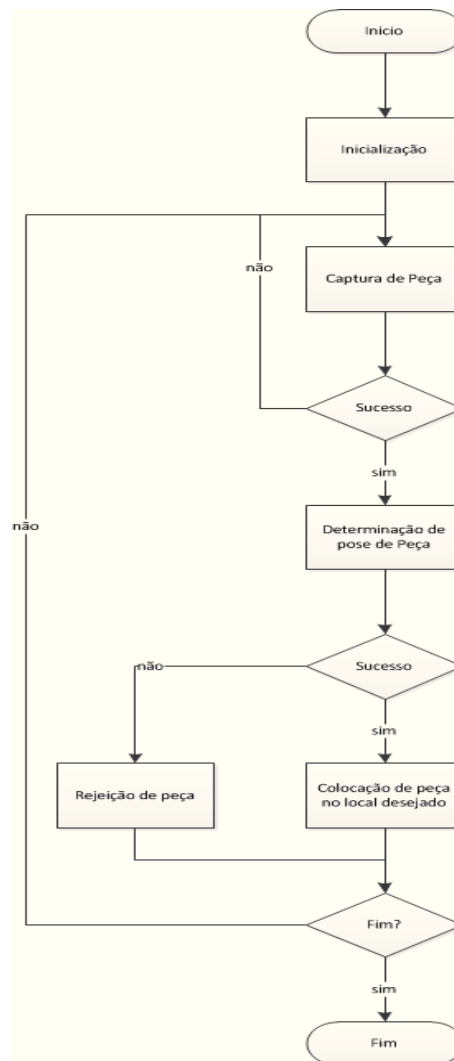


Figura 3.24: Fluxograma geral do processo implementado.

O processo de *bin-picking* começa com a inicialização do sistema. Esta inicialização contempla: (1) leitura da postura do manipulador robótico, (2) movimentação do manipulador robótico sem colisões para uma postura pré-determinada e (3) calibração entre os referenciais câmara e manipulador robótico (caso não exista calibração prévia).

A captura de peças, esquematizada na Figura 3.25, inicia-se com o movimento do manipulador robótico para uma posição que deixe a caixa com peças visível pois, para além de obstruir a caixa com peças, a presença do manipulador robótico ou da ferramenta numa imagem, constitui um elemento que se pode assemelhar ao modelo da peça, sendo contrária ao princípio de simplificação do modelo de peças (ver Secção 3.5). Segue-se a execução do algoritmo de identificação de peças a capturar (ver Secção 3.5), seguido da tentativa de captura especificada. A tentativa de captura começa com uma aproximação ao objeto. Esta aproximação tem como objetivo evitar colisões, como por exemplo atravessar um dos lados da caixa. Finda esta aproximação, a ferramenta encontra-se muito próxima do alvo. Neste momento, ativa-se a funcionalidade de captura de peças⁸ e move-se o manipulador até que o sensor responsável por verificar a captura de peças confirme o sucesso, ou até que a extensão do movimento ultrapasse a previsível. Segue-se o afastamento na mesma direção percorrida para a captura até chegar à zona de partida. De seguida move-se o manipulador para uma posição pré-determinada que não diminua a visibilidade da caixa de peças. Caso a captura não seja bem sucedida, repete-se o processo de determinação de peça a capturar e efetua-se a conseqüente tentativa de captura.

⁸na aplicação desenvolvida corresponde ao acionamento da ventosa.

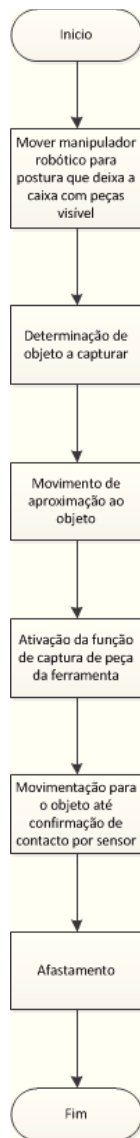


Figura 3.25: Fluxograma da captura de peças.

Após uma tentativa de captura bem sucedida, segue-se a determinação de pose da peça capturada (ver Secção 3.5). O processo de determinação de pose de peças encontra-se esquematizado na Figura 3.26.

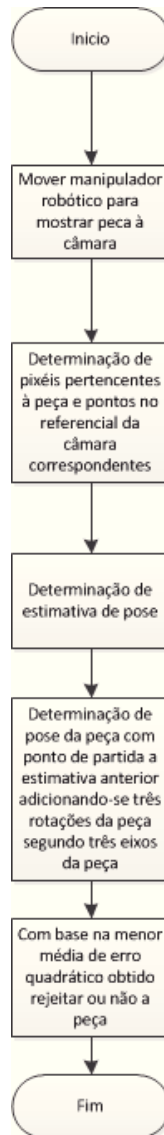


Figura 3.26: Fluxograma da determinação de pose de uma peça.

A determinação de pose de peças inclui uma movimentação para aproximar a peça da câmara a cores. Segue-se a determinação da pose da peça que inclui, caso seja necessária, uma rotação da peça com o objetivo desta se encontrar completamente visível na imagem. Após a determinação de pose, se a peça corresponder ao modelo, procede-se à colocação desta no local desejado, caso contrário rejeita-se a peça.

Capítulo 4

Testes e Resultados

Neste capítulo, apresentam-se resultados da aplicação dos algoritmos descritos nas Secções 3.5 e 3.6, comentando-se, quando oportuno, aspetos referentes às relações de compromisso estabelecidas. Os algoritmos introduzidos permitem a execução completa do processo de *bin-picking*. De modo a facilitar a leitura, apresentam-se os resultados dos módulos que constituem o sistema em separado:(1) estimativa de pose de peças com vista à sua captura, (2) captura de peças, (3) determinação de pose de peças.

4.1 Estimativa de pose com vista a captura de peças

Na situação atual não se possui luz controlada. Contudo, é possível obter-se diferentes situações de iluminação, nomeadamente:(1) luz solar lateral (ver Figura 4.1), (2) iluminação difusa, (ver Figura 4.1), (3) iluminação artificial por cima (ver Figura 4.2) e (4) iluminação artificial diagonal (ver Figura 4.2).

A iluminação disponível afeta significativamente o aspeto das imagens capturadas pela câmara de cores. É possível observar na Figura 4.3 a diferença na intensidade luminosa, a presença de sombras e ruído nas quatro configurações de iluminação apresentadas para uma caixa vazia.

Para avaliar a qualidade das deteções abordam-se os casos nos quais a caixa:(1) possui peças sem sobreposições, (2) possui peças com sobreposições e (3) não possui peças. Utilizaram-se os parâmetros da Tabela 4.1 para a execução dos doze testes propostos.



(a) Iluminação solar lateral (estores altos).



(b) Iluminação solar difusa (estores baixos).

Figura 4.1: Iluminação solar.



(a) Iluminação artificial de cima.



(b) Iluminação artificial diagonal.

Figura 4.2: Iluminação artificial.

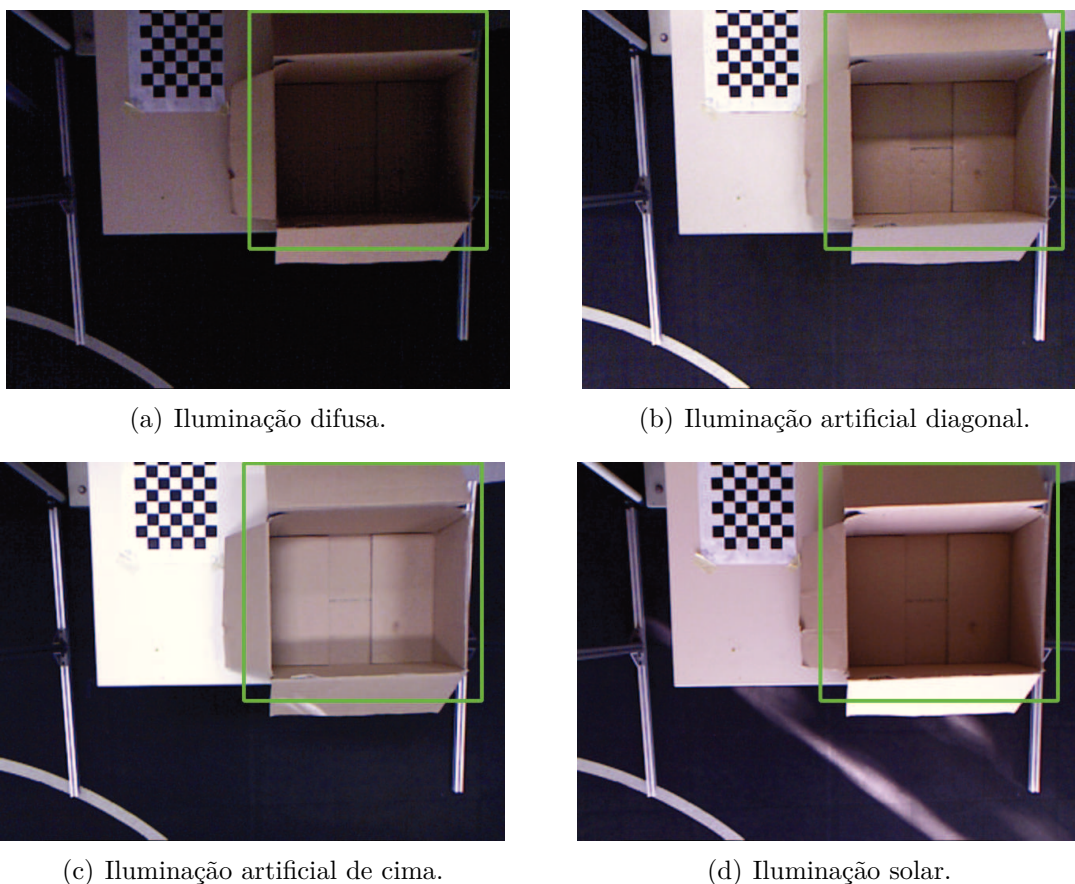


Figura 4.3: Aspetto de caixa vazia de acordo com a iluminação.

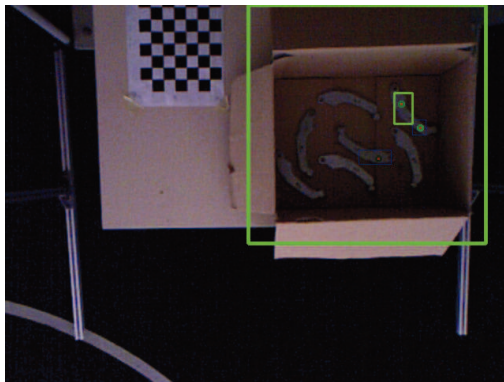
Os resultados para a situação na qual a caixa possui peças sem sobreposição encontram-se nas Figuras 4.4 e 4.5. Na Figura 4.5 vêem-se os contornos fechados encontrados com anotações a cores. Nestas anotações encontra-se o motivo pelo qual uma região não é considerada ou dois números correspondentes à classificação de forma (em baixo) e aspeto (em cima). Os motivos para não consideração de uma região são avaliados sequencialmente segundo a ordem: (1) Existência de estimativa de posição no espaço, assinalado por "3d"; (2) d_{aresta} pertence ao intervalo de valores do modelo, assinalado por "d"; (3) área pertence ao intervalo de valores do modelo, assinalado por "a"; (4) ponto de maior d_{aresta} se encontrar dentro da região de interesse, assinalado por "ri"; (5) existência de erro no cálculo da normal, assinalado por "n"; (6) c pertence ao intervalo do modelo, assinalado por "lrM"; (7) l pertence ao intervalo do modelo, assinalado por "lrm" e (8) aspeto pertence ao modelado, assinalado por "cor". Verifica-se que falhas no cálculo da normal constituem o principal motivo para regiões associadas a peças serem rejeitadas. Em alguns casos o menor lado do retângulo que engloba a região é o motivo de exclusão da região associada a uma peça. Por este motivo repetiu-se o teste ignorando falhas no cálculo da normal e utilizando para o lado maior do retângulo de menor área o limite máximo $c_{maxnovo} = \sqrt{l_{max}^2 + c_{max}^2}$ e o valor de lado menor para re-

tângulo de menor área medido $l_{max_{novo}} = \sqrt{2}l_{max}$, que corresponde à distância segundo a direção perpendicular ao segmento de linha de maior comprimento, para o caso de o retângulo de menor área ser um quadrado (situação na qual este comprimento é maior). Neste novo teste, ilustrado na figura 4.6, verifica-se a deteção de todas as peças nos casos de iluminação artificial de cima e iluminação difusa. No caso de iluminação diagonal não se deteta uma das peças devido à sua posição relativa à câmara e à fonte de luz. Verifica-se um gradiente de intensidade luminosa muito elevado num dos lados e muito fraco no outro, não se fechando os contornos dessa peça com sucesso. No ensaio com iluminação proveniente do sol não se detetaram quaisquer peças. O aspeto das peças foi o fator que impediu o seu reconhecimento, sendo os valores de média e desvio padrão 123,5 e 4,353 para o canal u e 139,9 e 5,950 para o canal v.

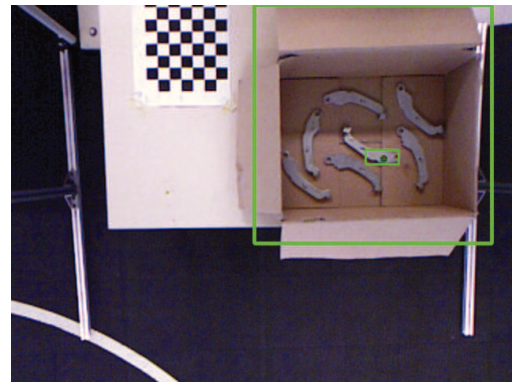
A cor detetada numa peça resulta da reflexão da iluminação, logo, é expectável que a presença de fontes de luz de diferentes espectros influencie a cor de uma região na câmara. Este efeito, nas peças consideradas, é agravado pela elevada refletividade. É possível resolver a falha na deteção, alterando os valores referentes ao aspeto no modelo. Com os parâmetros 123 e 139 para as médias nos canais U e V, e 10 para o desvio padrão em ambas as cores, obtiveram-se os resultados presentes na Figura 4.7, onde se verifica que todas as peças são detetadas. Durante os testes com iluminação difusa detetou-se um falso positivo, visível na Figura 4.8. Esta deteção, com valor 0,18 para o aspeto, está associada à combinação entre ruído e baixo nível de iluminação.

Segue-se o teste para a situação na qual existem sobreposições entre peças, encontrando-se nas Figuras 4.9 e 4.10 resultados obtidos para a situação na qual se ignoram falhas no cálculo das normais. O erro no cálculo das normais é o principal fator para a não deteção de maior quantidade de peças presentes. No ensaio com iluminação solar foram detetadas peças, contrariamente ao decorrido no teste com poucas peças. A grande maioria das regiões pertencentes a peças foi devidamente identificada. Na Figura 4.11 pode-se observar os falsos positivos detetados. No caso de iluminação difusa verificou-se a presença de falsos positivos esporádicos e um falso positivo relativamente persistente associado a um zona da imagem muito escura. Nos ensaios com os restantes tipos de iluminação foram detetados falsos positivos esporádicos, associados a regiões escuras e reflexões por parte das peças. Com isto a média da cor destas regiões aproximou-se da média da cor das peças. Observou-se uma deteção que engloba parte de uma peça e parte do fundo da caixa, no ensaio com a iluminação artificial de cima. Esta deteção deve-se a um contorno de baixo gradiente de intensidade luminosa.

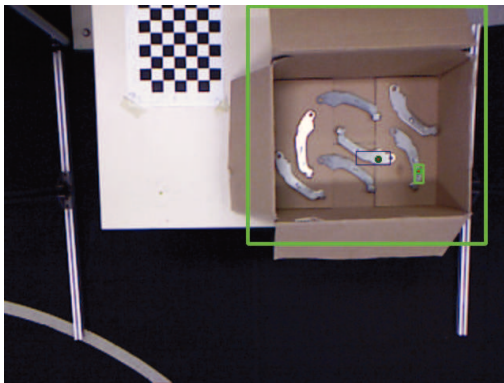
Para o teste em que a caixa se encontra vazia, o algoritmo não detetou nenhuma peça nos ensaios de iluminação difusa, solar ou artificial diagonal. No ensaio com iluminação



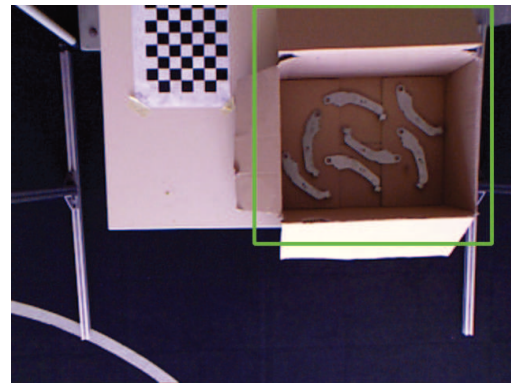
(a) Iluminação difusa.



(b) Iluminação artificial diagonal.

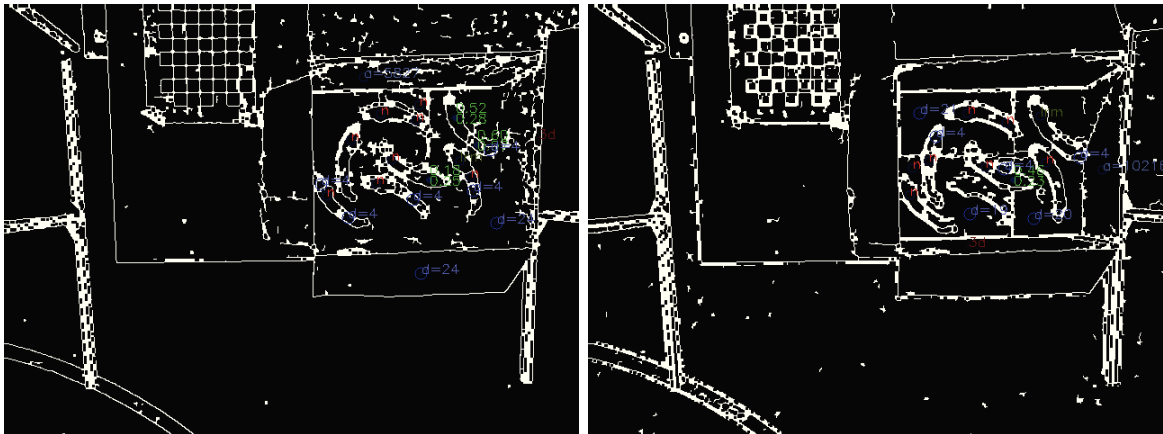


(c) Iluminação artificial de cima.



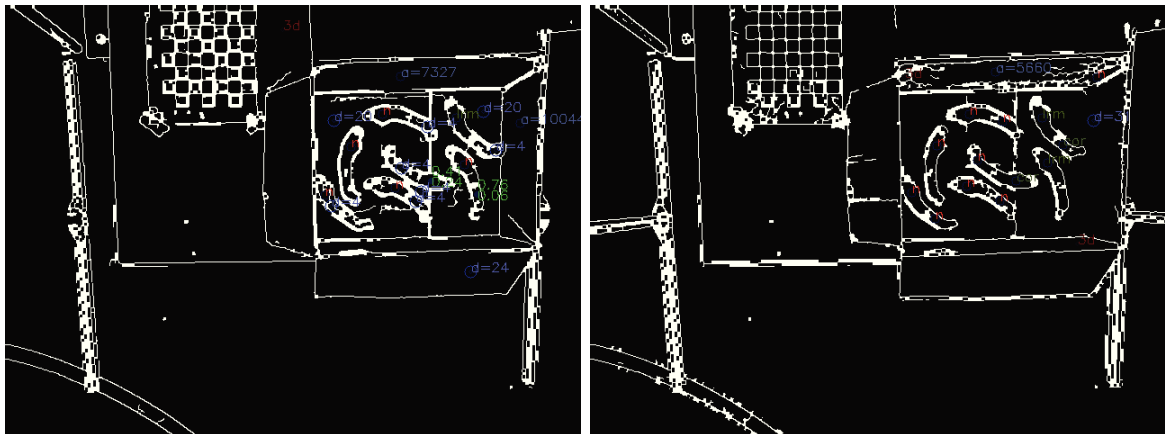
(d) Iluminação solar.

Figura 4.4: Aspeto de caixa com peças sem sobreposição de acordo com a iluminação.



(a) Iluminação difusa.

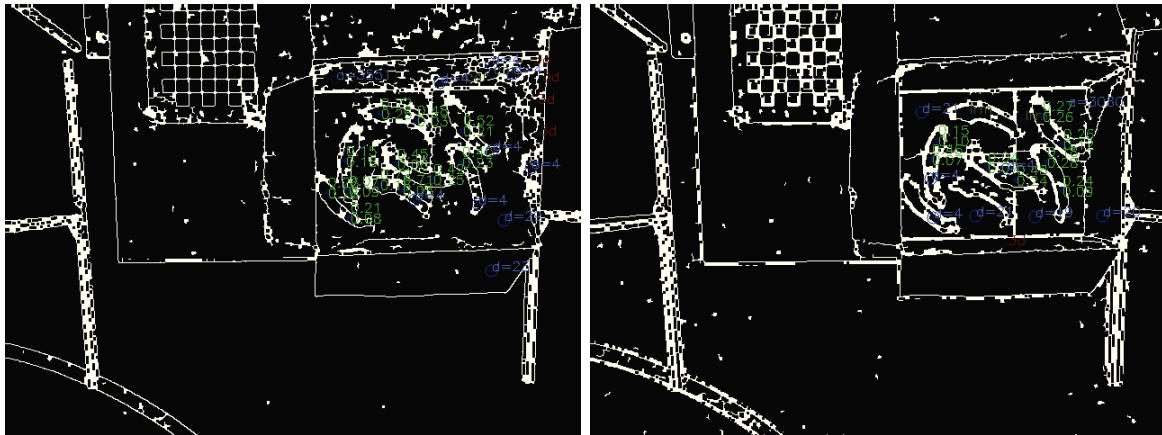
(b) Iluminação artificial diagonal.



(c) Iluminação artificial de cima.

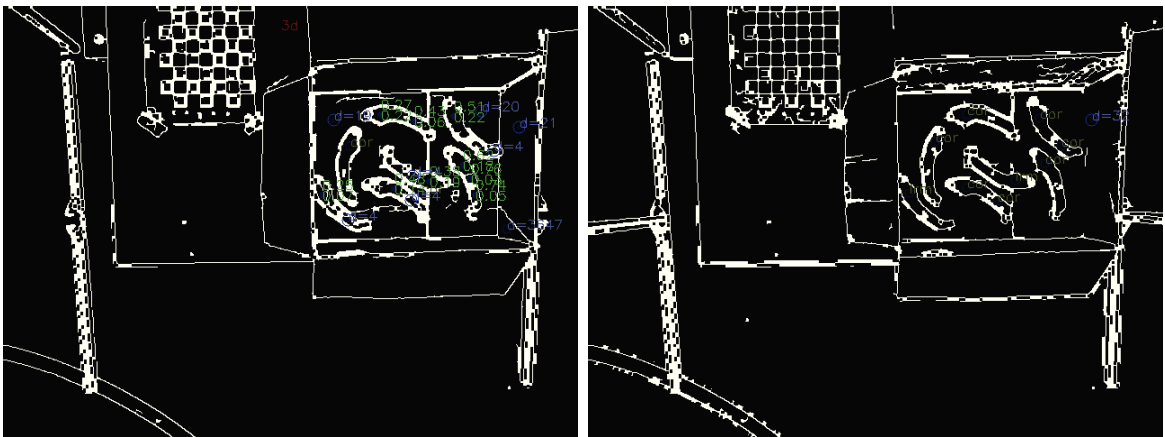
(d) Iluminação solar.

Figura 4.5: Deteções em caixa com peças sem sobreposição.



(a) Iluminação difusa.

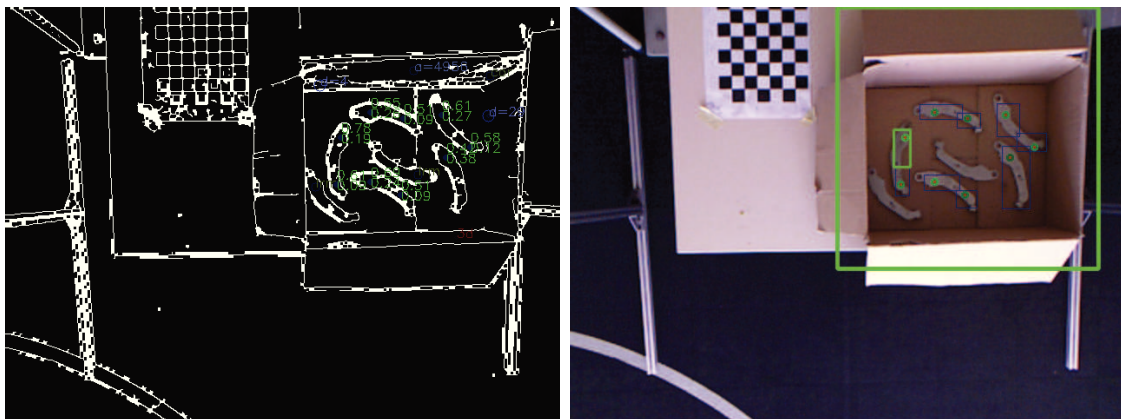
(b) Iluminação artificial diagonal.



(c) Iluminação artificial de cima.

(d) Iluminação solar.

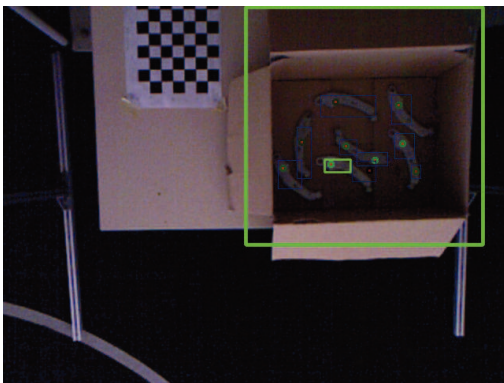
Figura 4.6: Deteções em caixa com peças sem sobreposição ignorando falhas na estimativa da normal e consideração de folga nas medidas do lado do retângulo.



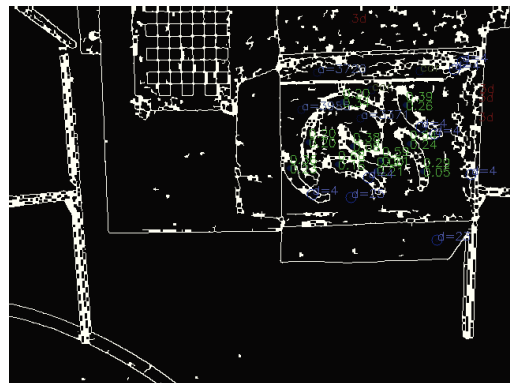
(a) Deteção efetuada.

(b) Anotação da imagem referente à deteção.

Figura 4.7: Deteção de peças com iluminação solar.

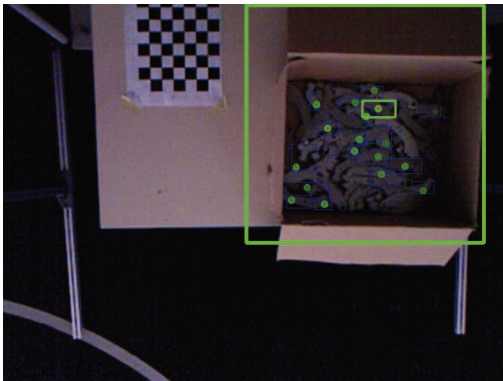


(a) Detecção efetuada.



(b) Anotação da imagem referente à detecção.

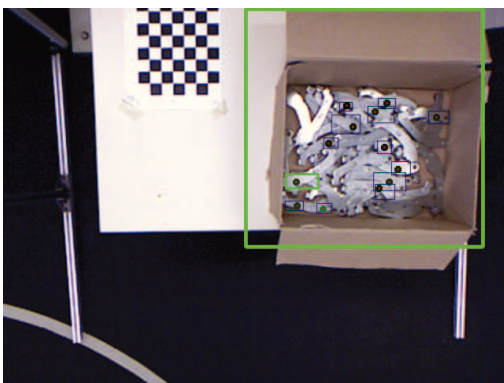
Figura 4.8: Detecção de falso positivo com iluminação difusa.



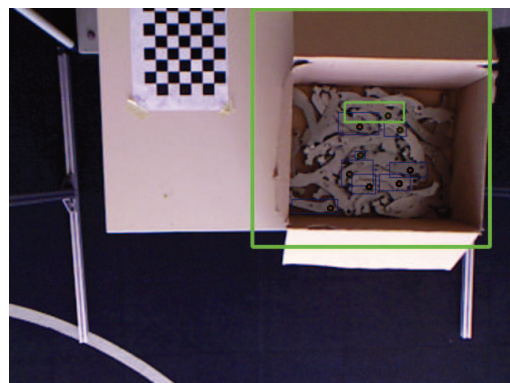
(a) Iluminação difusa.



(b) Iluminação artificial diagonal.

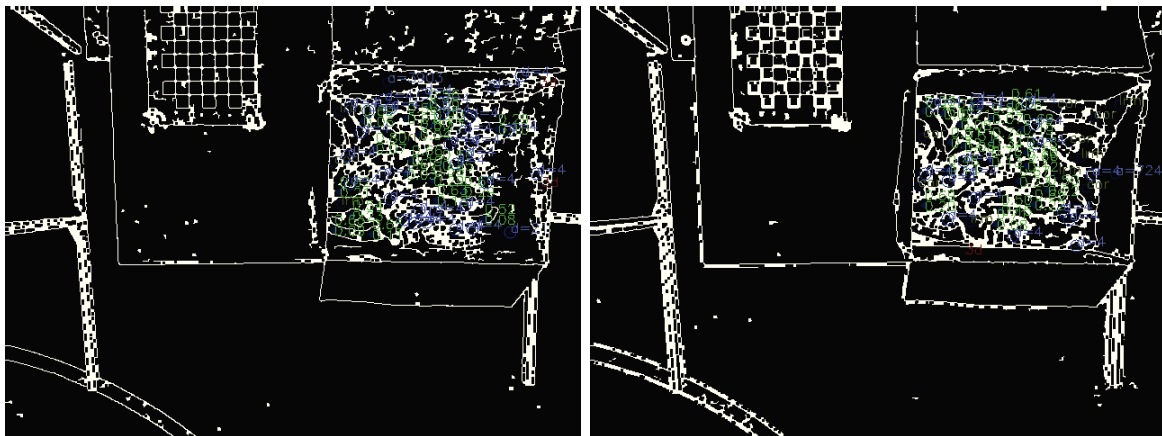


(c) Iluminação artificial de cima.



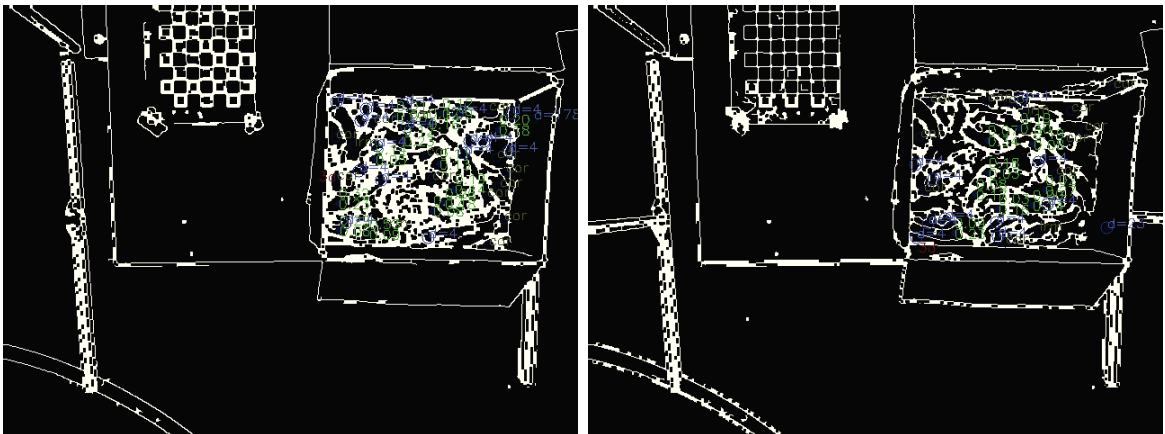
(d) Iluminação solar.

Figura 4.9: Aspetto de caixa com peças com sobreposição de acordo com a iluminação.



(a) Iluminação difusa.

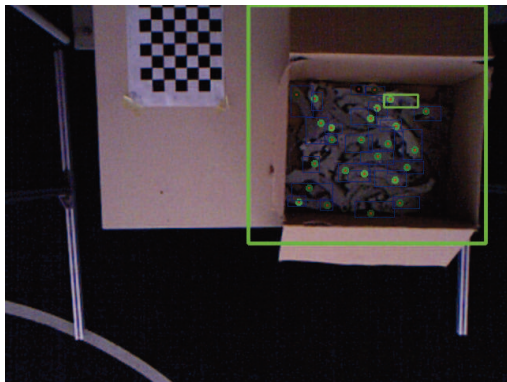
(b) Iluminação artificial diagonal.



(c) Iluminação artificial de cima.

(d) Iluminação solar.

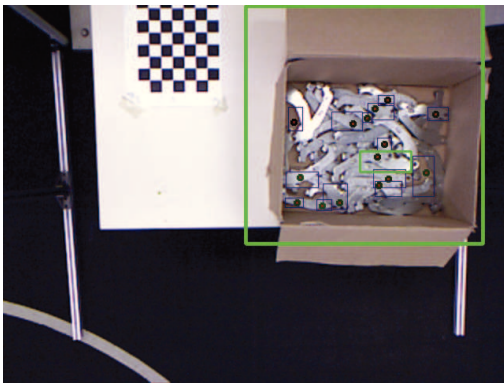
Figura 4.10: Deteções em caixa com peças com sobreposição.



(a) Iluminação difusa.



(b) Iluminação artificial diagonal.



(c) Iluminação artificial de cima.



(d) Iluminação solar.

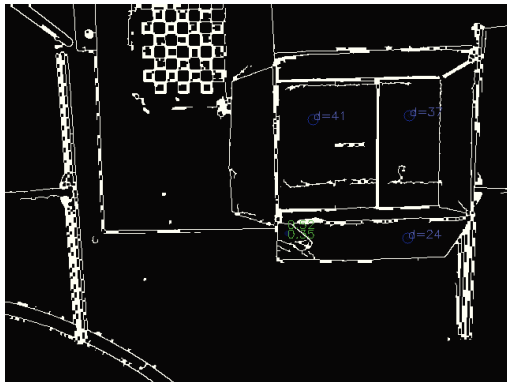
Figura 4.11: Deteções de falsos positivos em caixa com peças com sobreposição

Tabela 4.1: Parâmetros utilizados no algoritmo de procura de peças.

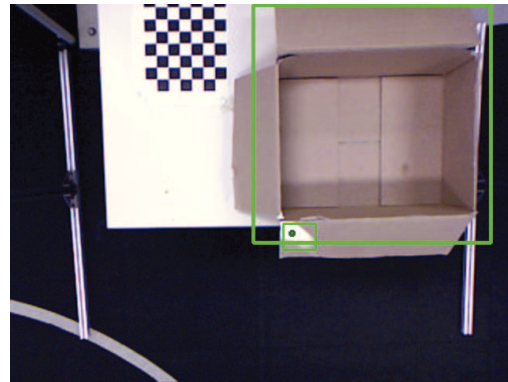
Parâmetro	Valor	Unidade
$x_{máx}$	0,45	metro
$x_{mín}$	0,00	metro
$y_{máx}$	0,05	metro
$y_{mín}$	-0,40	metro
$z_{máx}$	1,10	metro
$z_{mín}$	0,80	metro
$d_{aresta_{mín}}$	$7,5 \times 10^{-3}$	metro
$d_{aresta_{máx}}$	32×10^{-3}	metro
$área_{máx}$	6×10^{-3}	$metro^2$
$área_{mín}$	$1,767 \times 10^{-4}$	$metro^2$
$l_{máx}$	40×10^{-3}	metro
$l_{mín}$	15×10^{-3}	metro
$c_{máx}$	150×10^{-3}	metro
$c_{mín}$	15×10^{-3}	metro
<i>média cor canal u</i>	130	-
<i>desvio padrão máximo cor canal u</i>	12	-
<i>desvio padrão mínimo cor canal u</i>	0	-
<i>média cor canal v</i>	130	-
<i>desvio padrão máximo cor canal v</i>	12	-
<i>desvio padrão mínimo cor canal v</i>	0	-
$peso_{d_{aresta_n}}$	0,4	-
$peso_{a_n}$	0,5	-
$peso_{l_n}$	0,05	-
$peso_{c_n}$	0,05	-
Remoção de região do histórico caso classificação seja inferior a	0.001	-
Remoção de região do histórico caso a percentagem de deteções seja inferior a	0.1	-

artificial de cima foi detetada uma falsa região, apenas no caso em que se ignora a falha na deteção de normais. Esta região foi devidamente removida através do histórico de regiões. Esta deteção, ilustrada na Figura 4.12 a) e b), deve-se a reflexões por parte da caixa num dos seus bordos. A fonte de iluminação associada a esta reflexão tem origem na deficiente obstrução da luz solar pelos estores das janelas. A forma desta região atribui-se ao movimento dos ramos de árvores do lado de fora da janela. A cor média e o desvio padrão associados a esta deteção, 130.2 e 4.496 para o canal u e 131.9 e 4.369 para o canal v. A classificação do aspeto desta região é 0,52. Esta região foi detetada quatro vezes com o formato enquadrado no modelo, mas em três delas, a cor associada não propiciou a classificação como peça. Os conjuntos de parâmetros nestas situações foram:(1) 126,4 , 1,742 , 139,7 1,381 (2) 122,3 2,775 , 141,1 , 1,910 (3) 122,9 , 3,020 140,7 , 1,727 , correspondendo, respetivamente aos valores para a média no

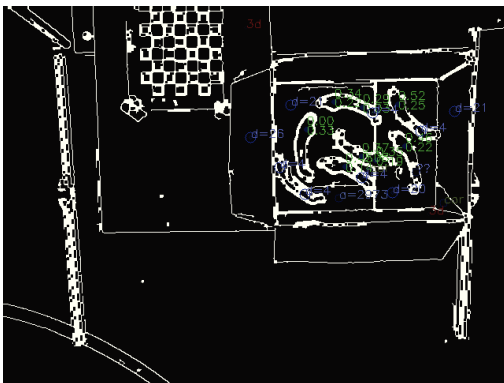
canal u , desvio padrão no canal u , média para o canal v e desvio padrão o canal v . Depreende-se assim que a excessiva intensidade luminosa refletida pela caixa alterou significativamente o aspeto desta.



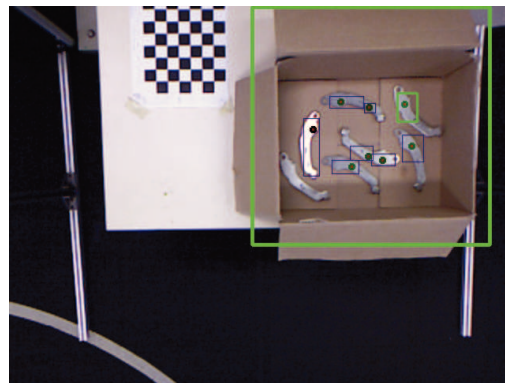
(a) Detecção efetuada.



(b) Anotação da imagem referente à deteção.



(c) Detecção correta efetuada.



(d) Anotação da imagem referente à deteção correta.

Figura 4.12: Detecção de falso positivo devido a reflexões de iluminação e correta deteção de zona com aspeto similar.

O algoritmo detetou peças com sucesso nos doze casos anteriormente mencionados. Devido à ausência de condições de iluminação controlada e ruído as deteções não são constantes em forma, aspeto e localização ao longo do tempo. O número de falsos positivos é muito baixo face ao número de deteções. Para os doze casos considerados, verifica-se que a sujidade nas peças não contribuiu significativamente para falhas na sua deteção face ao efeito da iluminação na sua aparência. A influência da iluminação notou-se mais nos testes em que a caixa possuía poucas peças. Aconselha-se que os valores médios e de desvio padrão do modelo a utilizar sejam adaptados à iluminação a utilizar.

A utilização do algoritmo na ausência de condições de iluminação controlada simplifica

o conjunto de estruturas necessárias à integração do sistema. A opção por utilização de técnicas de iluminação controlada deve ser pensada no contexto de uma relação de compromisso face aos resultados obtidos na sua ausência.

Para a escolha de uma caixa devem-se considerar os seguintes fatores:(1) a reflexão de iluminação e (2) diferença entre a cor da caixa e das peças a utilizar. É preferível escolher uma caixa cuja reflexão não influencie significativamente a percepção da cor (dependendo da posição da câmara face às fontes de luz). É conveniente maximizar a diferença entre a cor média da peça e da caixa. A escolha cuidadosa do fundo pode eliminar todos os falsos positivos detetados. Há ainda a hipótese de utilizar um sensor com maior precisão.

4.2 Captura de peças

A altura associada ao ponto de captura mais seguro (ponto de uma região onde se obtém d_{aresta}), possui ruído associado, variando ao longo do tempo como exemplificado na Figura 4.13, onde se verifica a variação de altura associada a um peça e na figura 4.14, onde se encontra a altura de uma entre seis peças detetadas no plano da mesa, sem qualquer sobreposição. A diferença entre o limite máximo e mínimo pode chegar a 1cm. Devido à distância entre as peças e a câmara ser superior a 500cm, o erro associado, em percentagem é, neste caso, inferior a 0,2%. Sendo o campo de visão da câmara de infravermelhos de 57 por 45 graus (ver Anexo A.1), considerando que os pontos se encontram na direção de um pixel no canto da imagem, o desvio é de $\frac{0,01}{2} \sqrt{\text{seno}(\frac{57^\circ}{2})^2 + \text{seno}(\frac{45^\circ}{2})^2}$, o que dá menos de 3,1mm (menos de metade do raio da ventosa). A ferramenta e o manipulador robótico são rígidos, não possuem elasticidade. Se a altura associada a uma deteção for menor que a real, a captura de peças falha. No caso de a distância medida ser maior que a real, um manipulador robótico "cego" não consegue finalizar o seu movimento com sucesso, efetuando demasiada pressão sobre a superfície das peças. Neste contexto o sensor de contacto entre a ventosa e uma peça é essencial a uma correta captura.

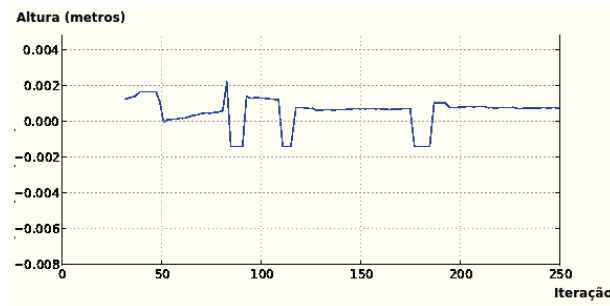


Figura 4.13: Variação da altura associada a uma possível captura de uma peça.

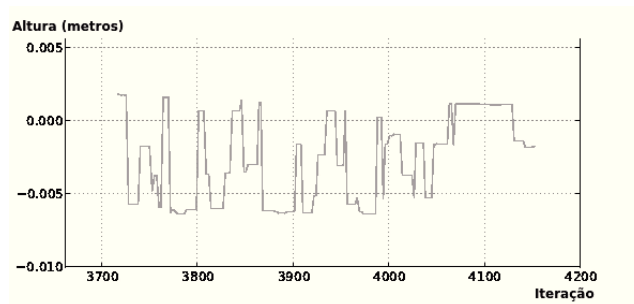


Figura 4.14: Variação da altura associada a uma possível captura de peças num mesmo plano.

4.3 Determinação de pose com vista a recolocação de peças

Na Figura 4.15 encontram-se resultados de deteção de peças, encontrando-se marcado a verde os contornos estimados, um retângulo e um ponto correspondentes à zona na qual a intensidade luminosa foi amostrada e o ponto de início para a região em crescimento. No processo de deteção de peça não se deve utilizar fontes de luz com a mesma direção da normal da câmara pela possibilidade de gerar zonas muito brilhantes na peça. Na Figura 4.16 estão presentes as inicializações do processo de determinação de pose. Os contornos detetados encontram-se a vermelho, com uma cruz a assinalar a centroide. O modelo e o resultado obtido para alinhamento da peça sem considerar a etapa de translações segundo os vetores próprios encontram-se representados a verde e azul com pontos "grandes". As inicializações resultantes do alinhamento dos momentos centrais de segunda ordem e centroide encontram-se representados com pontos "pequenos" a roxo, vermelho, verde e cinzento, na origem do referencial. Observa-se que a inicialização do processo de determinação de pose não é significativamente afetada por:

(1) pequenas zonas no interior da peça associadas a buracos ou sujidade não consideradas parte integrante da peça, ou (2) pequenos erros na determinação dos contornos das peças. Na Figura 4.17 encontram-se os resultados da determinação de pose para peças vistas de diferentes lados e com presença de erro na determinação dos contornos. A verde encontra-se o modelo, a azul encontra-se o resultado sem considerar a etapa de translações segundo os vetores próprios e a cinzento o alinhamento obtido considerando a etapa de translações segundo os vetores próprios. Observa-se a presença de erro de translação em algumas das poses que diminui drasticamente quando se consideram novas inicializações, compostas por translações do resultado obtido. O erro final encontra-se limitado pela discretização efetuada no reconhecimento da peça. À distância utilizada (14,4cm) um pixel corresponde a cerca de 0,25mm. A média do quadrado do erro obtido entre as correspondências de pontos ($1,0754 \times 10^{-08}$, $4,4327 \times 10^{-08}$, $4,1565 \times 10^{-08} m^2$) indica que se obteve correspondência praticamente perfeita (com base nas discretizações efetuadas) entre a peça e o modelo. Deve-se notar que, da elevada quantidade de pontos associados a uma peça, a maioria se encontra ao longo de linhas aproximadamente retas e paralelas ao eixo de maior variância. Por este motivo, translações segundo este eixo possuem relativamente pouca influência na média do erro quadrático. Para todos os pontos nestas linhas aos quais é aplicada uma pequena translação segundo a direção da linha, é possível encontrar um novo ponto na mesma linha muito próximo do anterior. Uma translação de quase meio centímetro visível a azul na Figura 4.17, possui associado um erro quadrático médio de $1,2573 \times 10^{-06} m^2$.

Utilizaram-se outras peças de dimensões semelhantes, encontrando-se na figura 4.18 uma comparação entre as peças sobre um mesmo retângulo. Na Figura 4.19 (que possui as mesmas anotações com código de cores igual às Figuras 4.15, 4.16 e 4.17) encontram-se os resultados da tentativa de determinação de pose de peças com modelo diferente. Obtiveram-se os erros médios quadráticos de $7,4585 \times 10^{-06}$, $7,2621 \times 10^{-06}$, $6,3562 \times 10^{-06}$, $6,3423 \times 10^{-06}$ e $6,3729 \times 10^{-06} m^2$.



Figura 4.15: Deteções de peças com vista ao cálculo de pose.

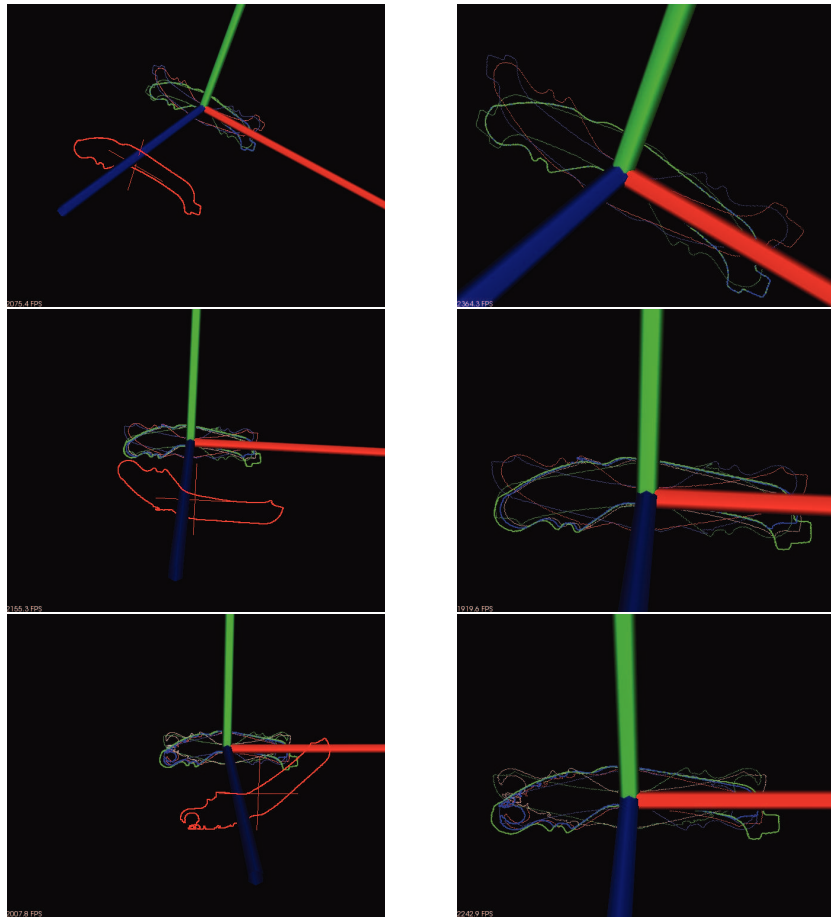


Figura 4.16: Cálculo de pose inicial.

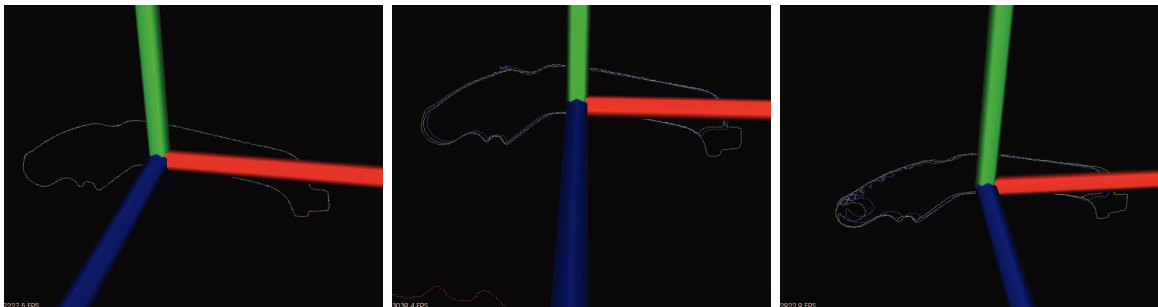


Figura 4.17: Pose calculada.



Figura 4.18: Comparação entre dois tipos de peças diferentes.

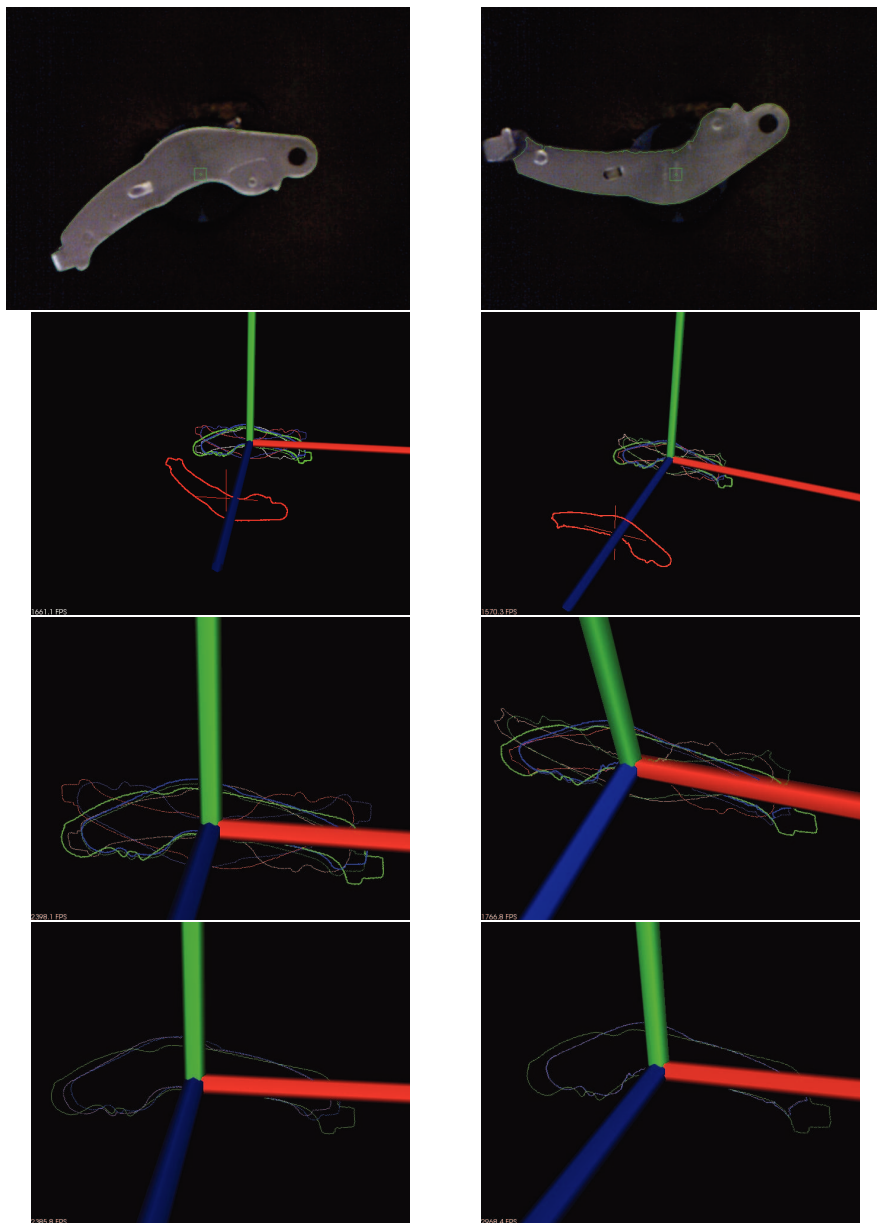


Figura 4.19: Cálculo de pose para peça de dimensões similares.

Capítulo 5

Conclusões e trabalho futuro

No âmbito desta dissertação foi desenvolvido e implementado um sistema automático de reposicionamento de peças, partindo do caso de estudo de peças metálicas aproximadamente planas, de baixa espessura e com aspeto brilhante. O sistema de deteção de peças utilizado foi desenvolvido tendo em vista os desafios inerentes ao caso de estudo, recorrendo a um sensor de baixo custo. Este sistema, aplicável em tempo real, foi testado sem condições de iluminação controlada. Para todas as condições testadas reconheceu com sucesso partes pertencentes a peças. A deteção de falsos positivos foi baixa, devendo-se a sua ocorrência sobretudo à conjugação de baixo nível de iluminação e ruído (uma das condições de teste) ou à conjugação entre a proximidade no aspeto da caixa e das peças (em termos de cor) e reflexões de iluminação entre as peças e a caixa. A utilização do algoritmo em condições de iluminação não controlada permite simplificar o conjunto de estruturas necessária à implementação do sistema (como apontado na introdução o aspeto que atualmente determina o custo de um sistema de *bin-picking*). O sensor de contacto no manipulador robótico para complementar a informação relativa ao mapa de profundidade revelou-se essencial à correta execução de captura de peças. A aquisição de informação relativa à profundidade, mesmo com falhas, foi vantajosa ao processo de deteção de peças, permitindo uma estimativa das dimensões da peça no mundo real e a execução do movimento de captura de peças, numa única sequência de movimentos, sem necessidade de interrupção para aquisição de informação adicional. O sistema de determinação de pose obteve sucesso para vistas de ambos os lados de uma peça correspondente ao modelo, assim como para pequenos erros na determinação de contornos. O sistema desenvolvido é aplicável a outros objetos com superfícies aproximadamente planas. A ausência de necessidade de treino de métodos de *machine learning*, em conjunto com a simples parametrização do sistema permite a adaptação a diferentes tipos de objetos de forma bastante célere, sendo apli-

cável a linhas de produção nas quais os objetos a introduzir se encontram em constante mudança.

Direções para trabalho futuro

Para desenvolvimento de trabalho futuro pode-se estudar a substituição ou desenvolvimento de *hardware*. A utilização da versão 2 do sensor Kinect, que foi lançada recentemente, é um dos caminhos a explorar.

Para desenvolvimento e teste de aplicações de *bin-picking* diferentes grupos de investigadores utilizam diferentes tipos de objetos. O desenvolvimento de uma base de dados, devidamente anotada e disponível *on-line* vai de encontro aos interesses desta área de pesquisa. Para esta base de dados, seria útil que se utilizassem diferentes tipos de peças, observadas por diferentes sensores de pontos de vista distintos, e que figurem aspetos como o ruído associado a medições no mundo real. A necessidade de observações de diferentes pontos de vista referidos anteriormente, para além de possibilitar o treino de algoritmos baseados em métodos de *machine learning*, é útil para o teste de algoritmos de perceção ativa. Para além de imagens a cores e nuvens de pontos, nesta base de dados podem ser incluídos aspetos como a massa, o centróide e os momentos angulares detetáveis na manipulação de um objeto. Em conjunto com a base de dados o desenvolvimento de métricas para comparação do sucesso entre os algoritmos apresenta especial interesse, por permitir estabelecer relações de compromisso na escolha entre a aplicação de diferentes métodos ou na sua fusão.

Bibliografia

- [1] K. Ikeuchi, “Generating an interpretation tree from a cad model for 3d-object recognition in bin-picking tasks,” *International Journal of Computer Vision*, vol. 1, no. 2, pp. 145–165, 1987. [Online]. Disponível em: <http://dx.doi.org/10.1007/BF00123163>
- [2] S. Hutchinson, G. Hager, e P. Corke, “A tutorial on visual servo control,” *Robotics and Automation, IEEE Transactions on*, vol. 12, no. 5, pp. 651–670, Oct 1996.
- [3] D. Kragic e H. I. Christensen, “Survey on visual servoing for manipulation,” Computational Vision and Active Perception Laboratory, Relatório Tec., 2002.
- [4] T. Moeslund e J. Kirkegaard, “Pose estimation of randomly organized stator housings,” in *Image Analysis*, ser. Lecture Notes in Computer Science, H. Kalviainen, J. Parkkinen, e A. Kaarna, Eds. Springer Berlin Heidelberg, 2005, vol. 3540, pp. 679–688. [Online]. Disponível em: http://dx.doi.org/10.1007/11499145_69
- [5] J. Kirkegaard e T. Moeslund, “Bin-picking based on harmonic shape contexts and graph-based matching,” in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 2, 2006, pp. 581–584.
- [6] A. Pochyly, T. Kubela, V. Singule, e P. Cihak, “3d vision systems for industrial bin-picking applications,” in *MECHATRONIKA, 2012 15th International Symposium*, Dec 2012, pp. 1–6.
- [7] A. Pochyly, T. Kubela, M. Kozak, e P. Cihak, “Robotic vision for bin-picking applications of various objects,” in *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, June 2010, pp. 1–5.

- [8] M. Palzkill e A. Verl, “Object pose detection in industrial environment,” in *Robotics; Proceedings of ROBOTIK 2012; 7th German Conference on*, May 2012, pp. 1–5.
- [9] S. Fuchs, S. Haddadin, M. Keller, S. Parusel, A. Kolb, e M. Suppa, “Cooperative bin-picking with time-of-flight camera and impedance controlled dlr lightweight robot iii,” in *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, Oct 2010, pp. 4862–4867.
- [10] R. A. Tauro, B. Kaiser, e H. Woern, “Path planning process optimization for a bin picking system,” in *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, June 2010, pp. 1–7.
- [11] J. Xu, S. Pu, G. Zeng, e H. Zha, “3d pose estimation for bin-picking task using convex hull,” in *Mechatronics and Automation (ICMA), 2012 International Conference on*, Aug 2012, pp. 1381–1385.
- [12] J. A. Marvel, K. Saidi, R. Eastman, T. Hong, G. Cheok, e E. Messina, “Technology readiness levels for randomized bin picking,” in *Proceedings of the Workshop on Performance Metrics for Intelligent Systems*, ser. PerMIS ’12. New York, NY, USA: ACM, 2012, pp. 109–113. [Online]. Disponível em: <http://doi.acm.org/10.1145/2393091.2393114>
- [13] J. Rodrigues, J.-S. Kim, M. Furukawa, J. Xavier, P. Aguiar, e T. Kanade, “6d pose estimation of textureless shiny objects using random ferns for bin-picking,” in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, Oct 2012, pp. 3334–3341.
- [14] M. Bennamoun e G. J. Mamic, *Object recognition: fundamentals and case studies*. Springer, 2002.
- [15] M. Bicego, U. Castellani, e V. Murino, “A hidden markov model approach for appearance-based 3d object recognition,” *Pattern Recognition Letters*, vol. 26, no. 16, pp. 2588–2599, 2005.
- [16] A. Jain e C. Dorai, “3d object recognition: Representation and matching,” *Statistics and Computing*, vol. 10, no. 2, pp. 167–182, 2000. [Online]. Disponível em: <http://dx.doi.org/10.1023/A%3A1008998410728>
- [17] E. Trucco e A. Verri, *Introductory Techniques for 3D Computer Vision*. Prentice Hall, 1998.

- [18] J. Tsotsos, *The Encyclopedia of Artificial Intelligence*. John Wiley and Sons, 1992.
- [19] S. Dickinson, *What is Cognitive Science?* Basil Blackwell Publishers, 1999.
- [20] A. Andreopoulos e J. K. Tsotsos, “50 years of object recognition: Directions forward,” *Computer Vision and Image Understanding*, vol. 117, no. 8, pp. 827 – 891, 2013. [Online]. Disponível em: <http://www.sciencedirect.com/science/article/pii/S107731421300091X>
- [21] H. Ali e N. Figueroa, “Segmentation and pose estimation of planar metallic objects,” in *Computer and Robot Vision (CRV), 2012 Ninth Conference on*, May 2012, pp. 376–382.
- [22] R. Rusu, N. Blodow, e M. Beetz, “Fast point feature histograms (fpfh) for 3d registration,” in *Robotics and Automation, 2009. ICRA '09. IEEE International Conference on*, May 2009, pp. 3212–3217.
- [23] R. Rusu, “Semantic 3d object maps for everyday manipulation in human living environments,” *KI - Künstliche Intelligenz*, vol. 24, no. 4, pp. 345–348, 2010. [Online]. Disponível em: <http://dx.doi.org/10.1007/s13218-010-0059-6>
- [24] K. Liu, Z. Sun, e M. Fujii, “Ellipse detection based bin-picking visual servoing system,” in *Pattern Recognition (CCPR), 2010 Chinese Conference on*, Oct 2010, pp. 1–5.
- [25] J.-K. Oh, K. Baek, D. Kim, e S. Lee, “Development of structured light based bin picking system using primitive models,” in *Assembly and Manufacturing, 2009. ISAM 2009. IEEE International Symposium on*, Nov 2009, pp. 46–52.
- [26] M. Hao, Z. Sun, e M. Fujii, “Ellipse detection based long range robust visual servoing,” in *Journal of Central South University*. Springer, 2007, vol. 38, pp. 432 –439.
- [27] J. Canny, “A computational approach to edge detection,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. PAMI-8, no. 6, pp. 679–698, Nov 1986.
- [28] A. Fitzgibbon, M. Pilu, e R. Fisher, “Direct least square fitting of ellipses,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 21, no. 5, pp. 476–480, May 1999.

- [29] R. Safaee-Rad, I. Tchoukanov, K. Smith, e B. Benhabib, “Three-dimensional location estimation of circular features for machine vision,” *Robotics and Automation, IEEE Transactions on*, vol. 8, no. 5, pp. 624–640, Oct 1992.
- [30] M. A. Wani e B. Batchelor, “Edge-region-based segmentation of range images,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 16, no. 3, pp. 314–319, Mar 1994.
- [31] S. Lee, J. Kim, M. Lee, K. Yoo, L. Barajas, e R. Menassa, “3d visual perception system for bin picking in automotive sub-assembly automation,” in *Automation Science and Engineering (CASE), 2012 IEEE International Conference on*, Aug 2012, pp. 706–713.
- [32] M. Nieuwenhuisen, D. Droschel, D. Holz, J. Stuckler, A. Berner, J. Li, R. Klein, e S. Behnke, “Mobile bin picking with an anthropomorphic service robot,” in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, May 2013, pp. 2327–2334.
- [33] N. J. Mitra, N. Gelfand, H. Pottmann, e L. Guibas, “Registration of point cloud data from a geometric optimization perspective,” in *Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, ser. SGP '04. New York, NY, USA: ACM, 2004, pp. 22–31. [Online]. Disponível em: <http://doi.acm.org/10.1145/1057432.1057435>
- [34] D. Buchholz, S. Winkelbach, e F. M. Wahl, “Ransam for industrial bin-picking,” in *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, June 2010, pp. 1–6.
- [35] G. Stockman, “Object recognition and localization via pose clustering,” *Computer Vision, Graphics, and Image Processing*, vol. 40, no. 3, pp. 361 – 387, 1987. [Online]. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0734189X87801470>
- [36] J.-K. Oh, C.-H. Lee, S.-H. Lee, S.-H. Jung, D. Kim, e S. Lee, “Development of a structured-light sensor based bin-picking system using icp algorithm,” in *Control Automation and Systems (ICCAS), 2010 International Conference on*, Oct 2010, pp. 1673–1677.
- [37] J.-K. Oh, S. Lee, e C.-H. Lee, “Stereo vision based automation for a bin-picking solution,” *International Journal of Control, Automation and Systems*, vol. 10, no. 2, pp. 362–373, 2012. [Online]. Disponível em: <http://dx.doi.org/10.1007/s12555-012-0216-9>

- [38] Cognex Corporation, “Patmax and patquick,” 2005.
- [39] “Cognex mvs-8000 series,” in *EVG 620 User Manuals*, 2013.
- [40] K. Boehnke, “Object localization in range data for robotic bin picking,” in *Automation Science and Engineering, 2007. CASE 2007. IEEE International Conference on*, Sept 2007, pp. 572–577.
- [41] I. Park, M. Germann, M. Breitenstein, e H. Pfister, “Fast and automatic object pose estimation for range images on the gpu,” *Machine Vision and Applications*, vol. 21, no. 5, pp. 749–766, 2010. [Online]. Disponível em: <http://dx.doi.org/10.1007/s00138-009-0209-8>
- [42] M. Palzkill, T. Ledermann, e A. Verl, “Anticipation-preprocessing for object pose detection,” in *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, June 2010, pp. 1–6.
- [43] M. Ulrich, C. Wiedemann, e C. Steger, “Combining scale-space and similarity-based aspect graphs for fast 3d object recognition,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 10, pp. 1902–1914, Oct 2012.
- [44] C. Hema, M. Paulraj, R. Nagarajan, e S. Yaacob, “Object localization using stereo sensors for adept scara robot,” in *Robotics, Automation and Mechatronics, 2006 IEEE Conference on*, June 2006, pp. 1–5.
- [45] T. Drummond e R. Cipolla, “Real-time visual tracking of complex structures,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, no. 7, pp. 932–946, Jul 2002.
- [46] D. Buchholz, D. Kubus, I. Weidauer, A. Scholz, e F. M. Wahl, “Combining visual and inertial features for efficient grasping and bin-picking,” in *Proceedings of the International Conference on Robotics and Automation (ICRA). IEEE*, 2014.
- [47] D. Kubus, T. Kroger, e F. Wahl, “On-line estimation of inertial parameters using a recursive total least-squares approach,” in *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, Sept 2008, pp. 3845–3852.
- [48] P. Arbelaez, M. Maire, C. Fowlkes, e J. Malik, “Contour detection and hierarchical image segmentation,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 5, pp. 898–916, May 2011.
- [49] R. Xiaofeng e L. Bo, “Discriminatively trained sparse code gradients for contour detection,” in *Advances in Neural Information Processing Systems 25*,

- F. Pereira, C. Burges, L. Bottou, e K. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 584–592. [Online]. Disponível em: <http://papers.nips.cc/paper/4787-discriminatively-trained-sparse-code-gradients-for-contour-detection.pdf>
- [50] Y. Pati, R. Rezaifar, e P. S. Krishnaprasad, “Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition,” in *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*, Nov 1993, pp. 40–44 vol.1.
- [51] M. Aharon, M. Elad, e A. Bruckstein, “k -svd: An algorithm for designing over-complete dictionaries for sparse representation,” *Signal Processing, IEEE Transactions on*, vol. 54, no. 11, pp. 4311–4322, Nov 2006.
- [52] P. Dollar e C. Zitnick, “Structured forests for fast edge detection,” in *Computer Vision (ICCV), 2013 IEEE International Conference on*, Dec 2013, pp. 1841–1848.
- [53] Y. Zhao, W. Gui, e Z. Chen, “Edge detection based on multi-structure elements morphology,” in *Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on*, vol. 2, 2006, pp. 9795–9798.
- [54] C. Lopez-Molina, B. D. Baets, H. Bustince, J. Sanz, e E. Barrenechea, “Multiscale edge detection based on gaussian smoothing and edge tracking,” *Knowledge-Based Systems*, vol. 44, no. 0, pp. 101 – 111, 2013. [Online]. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0950705113000464>
- [55] S. feng Ma, G. feng Zheng, L. xu Jin, S. li Han, e R. feng Zhang, “Directional multiscale edge detection using the contourlet transform,” in *Advanced Computer Control (ICACC), 2010 2nd International Conference on*, vol. 2, March 2010, pp. 58–62.
- [56] M. Fang, G. Yue, e Q. Yu, “The study on an application of otsu method in canny operator,” in *International Symposium on Information Processing(IROS), 2012 IEEE/RSJ International Conference on*, Aug 2009, pp. 109–112.
- [57] R. Medina-Carnicer, R. Muñoz-Salinas, E. Yeguas-Bolivar, e L. Diaz-Mas, “A novel method to look for the hysteresis thresholds for the canny edge detector,” *Pattern Recognition*, vol. 44, no. 6, pp. 1201 – 1211, 2011. [Online]. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0031320310005741>
- [58] O. Nobuyuki, “A threshold selection method from gray-level histograms,” *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 9, no. 1, pp. 62–66, Jan 1979.

- [59] H. Cai, Z. Yang, X. Cao, W. Xia, e X. Xu, “A new iterative triclass thresholding technique in image segmentation,” *Image Processing, IEEE Transactions on*, vol. 23, no. 3, pp. 1038–1046, March 2014.
- [60] B. Enjarini e A. Graser, “Planar segmentation from depth images using gradient of depth feature,” in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, Oct 2012, pp. 4668–4674.
- [61] M. A. Fischler e R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981. [Online]. Disponível em: <http://doi.acm.org/10.1145/358669.358692>
- [62] R. P. d. Figueiredo, P. Moreno, e A. Bernardino, “Fast 3d object recognition of rotationally symmetric objects,” in *Pattern Recognition and Image Analysis*, ser. Lecture Notes in Computer Science, J. M. Sanches, L. Micó, e J. Cardoso, Eds. Springer Berlin Heidelberg, 2013, vol. 7887, pp. 125–132. [Online]. Disponível em: http://dx.doi.org/10.1007/978-3-642-38628-2_14
- [63] D. Bryner, A. Srivastava, e E. Klassen, “Affine-invariant, elastic shape analysis of planar contours,” in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, June 2012, pp. 390–397.
- [64] P. Besl e N. D. McKay, “A method for registration of 3-d shapes,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 14, no. 2, pp. 239–256, Feb 1992.
- [65] H. Pottmann, Q.-X. Huang, Y.-L. Yang, e S.-M. Hu, “Geometry and convergence analysis of algorithms for registration of 3d shapes,” *International Journal of Computer Vision*, vol. 67, no. 3, pp. 277–296, 2006. [Online]. Disponível em: <http://dx.doi.org/10.1007/s11263-006-5167-2>
- [66] A. Segal, D. Haehnel, e S. Thrun, “Generalized-icp.” in *Robotics: Science and Systems*, vol. 2, no. 4, 2009.
- [67] J. Flusser, “On the independence of rotation moment invariants,” *Pattern Recognition*, vol. 33, no. 9, pp. 1405 – 1410, 2000. [Online]. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0031320399001272>
- [68] P. Felzenszwalb, D. McAllester, e D. Ramanan, “A discriminatively trained, multiscale, deformable part model,” in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, June 2008, pp. 1–8.

- [69] S. Lucey, R. Navarathna, A. B. Ashraf, e S. Sridharan, “Fourier lucas-kanade algorithm,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 6, pp. 1383–1396, June 2013.
- [70] H. J. Seo e P. Milanfar, “Training-free, generic object detection using locally adaptive regression kernels,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 32, no. 9, pp. 1688–1704, Sept 2010.
- [71] L.-P. Ellekilde, J. Jorgensen, D. Kraft, N. Kruger, J. Piater, e H. Petersen, “Applying a learning framework for improving success rates in industrial bin picking,” in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, Oct 2012, pp. 1637–1643.
- [72] M. R. Andersen, T. Jensen, P. Lisouski, A. K. Mortensen, M. K. Hansen, T. Gregersen, e P. Ahrendt, “Kinect depth sensor evaluation for computer vision applications,” Arhus Universitet, Relatório Tec., Feb 2012. [Online]. Disponível em: <http://www.eng.au.dk>
- [73] “Data streams returned by the kinect,” Feb 2014. [Online]. Disponível em: <http://www.mathworks.com/help/imaq/data-streams-returned-by-the-kinect.html>
- [74] H. Pottmann, Q.-X. Huang, Y.-L. Yang, e S.-M. Hu, “Geometry and convergence analysis of algorithms for registration of 3d shapes,” *International Journal of Computer Vision*, vol. 67, no. 3, pp. 277–296, 2006. [Online]. Disponível em: <http://dx.doi.org/10.1007/s11263-006-5167-2>
- [75] *Kinect Unleashed: Getting Control over High Resolution Depth Maps*, May 2013. [Online]. Disponível em: <https://cvhci.anthropomatik.kit.edu/~manel/publications/mva2013RGBD.pdf>
- [76] J. Smisek, M. Jancosek, e T. Pajdla, “3d with kinect,” in *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, Nov 2011, pp. 1154–1160.
- [77] “Mt9m112 datasheet,” Feb. [Online]. Disponível em: <http://pdf1.alldatasheet.com/datasheet-pdf/view/111283/MICRON/MT9M112.html>
- [78] “Mt9m112 datasheet,” Feb. [Online]. Disponível em: http://white.stanford.edu/teach/images/c/c8/MT9M112_DS_E.pdf
- [79] “Mt9m001 - 1/2-inch megapixel digital image sensor features,” Feb. [Online]. Disponível em: http://pdf.datasheetcatalog.com/datasheets2/58/588740_1.pdf

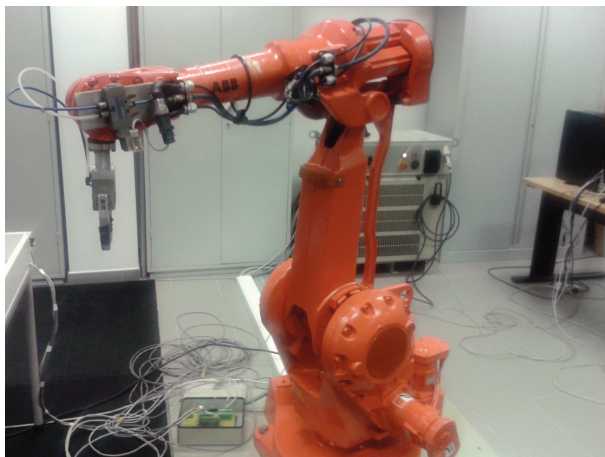
- [80] “Kinect accuracy.” [Online]. Disponível em: http://wiki.ros.org/openni_kinect/kinect_accuracy
- [81] J. Han, L. Shao, D. Xu, e J. Shotton, “Enhanced computer vision with microsoft kinect sensor: A review,” *Cybernetics, IEEE Transactions on*, vol. 43, no. 5, pp. 1318–1334, Oct 2013.
- [82] C. B. Duane, “Close-range camera calibration,” *Photogramm. Eng*, vol. 37, pp. 855–866, 1971.
- [83] D. Marquardt, “An algorithm for least-squares estimation of nonlinear parameters,” *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963. [Online]. Disponível em: <http://dx.doi.org/10.1137/0111030>
- [84] A. Ramey, “Kinect calibration,” Feb 2014. [Online]. Disponível em: <https://sites.google.com/site/rameyarnaud/research/ros/kinect-calibration>

Esta página foi intencionalmente deixada em branco.

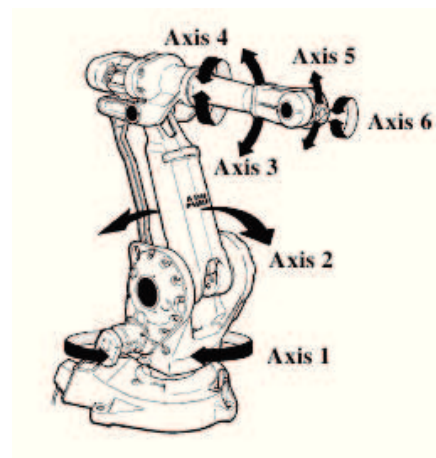
Apêndice A

Especificações dos Componentes do Sistema Físico

O manipulador industrial utilizado é um IRB2400 M98A de 1,5m da ABB, com capacidade de cargas de 16kg e seis graus de liberdade, ver FiguraA.1.



(a) IRB2400.



(b) Graus de Liberdade ,retirado de Product Manual IRB 2400 M98A, On-line Manual.

Figura A.1: Manipulador Industrial.

O controlador do manipulador industrial é um IRC S4C Plus da ABB, ver Figura A.2. As ligações entre o controlador, a ferramenta o o sensor encontram-se na Figura C.



Figura A.2: Controlador robótico utilizado.

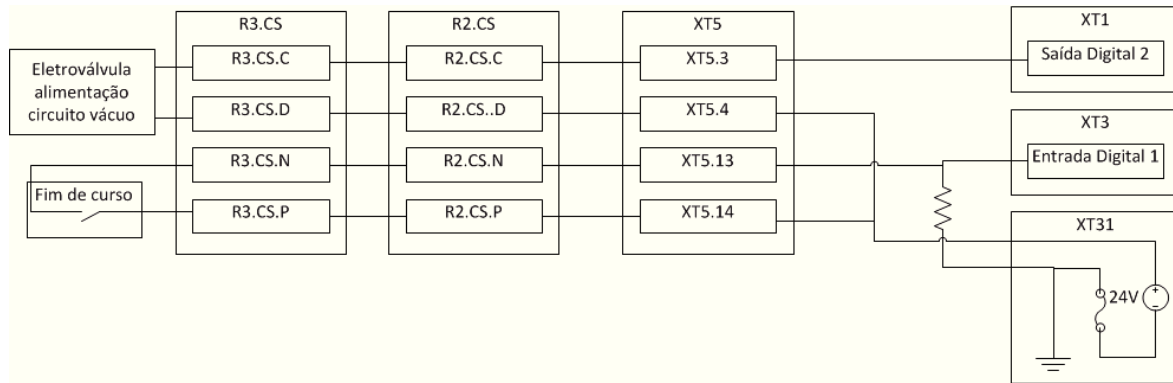


Figura A.3: Ligações entre controlador robótico, sensor fim de curso e eletroválvula.

A ferramenta utilizada, ilustrada na Figura A.4, consiste numa ventosa ligada a um circuito de produção vácuo. Este circuito é alimentado através de uma canalização de ar comprimido já presente no manipulador industrial utilizado. A admissão de ar comprimido é controlada com uma eletroválvula acionada através do controlador robótico (ver Figura A.3).

O componente utilizado para produzir o vácuo é um M10A6-NB da PIAB, com pressão de entrada máxima de $6 \times 10^5 Pa$ e pressão máxima de $-8,4 \times 10^4 Pa$. O vácuo produzido é ligado a uma ventosa de raio de 7,5mm. Com base nos valores atribuídos à ferramenta utilizada, com base nas equações A.1,A.2,A.3 e A.4, calcularam-se os valores de referência: (1) área da ventosa de $1,767 \times 10^{-4} m^2$, (2) força máxima associada à ferramenta de $14,844 N$, (3) peso máximo suportável por uma peça e $1,5 Kg$, considerando a aceleração gravítica (g) igual a $9,81 ms^{-2}$ e (4) aceleração máxima na qual um objeto de $0,1 Kg$ (peso das peças modelo) é suportado como $148,4 ms^{-2}$.

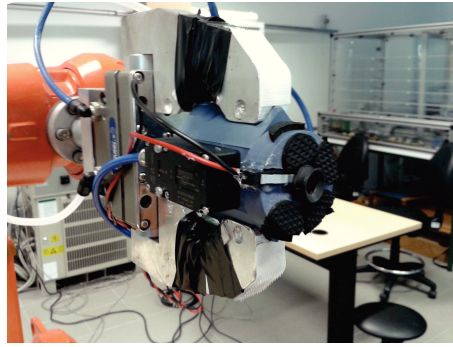


Figura A.4: Ferramenta utilizada.

$$\acute{a}rea_{ventosa} = \pi * raio_{ventosa}^2 \quad (A.1)$$

$$for\c{c}a_{m\acute{a}xima} = \acute{a}rea_{ventosa} * press\c{o}m\acute{a}xima \quad (A.2)$$

$$peso_{m\acute{a}ximo} = \frac{for\c{c}a_{m\acute{a}xima}}{g} \quad (A.3)$$

$$acelera\c{c}\tilde{a}o_{m\acute{a}xima} = \frac{for\c{c}a_{m\acute{a}xima}}{massa_{objeto}} \quad (A.4)$$

O sensor com o objetivo de detetar pe\c{c}as capturadas pela ferramenta consiste num fim de curso instalado ao lado da ventosa de modo a que quando uma pe\c{c}a seja capturada por ela pressione a patilha do sensor (ver Figura A.5).



Figura A.5: Sensor de fim de curso.

A.1 Sensor Microsoft Kinect para a Xbox 360

Numa aplicação de *bin-picking*, a principal responsabilidade do sistema sensorial é a identificação da postura dos objetos de interesse com rigor suficiente para que a consequente recolha e recolocação dos mesmos seja considerada bem sucedida. Neste contexto, a visão é um tipo importante de sensor na medida em que é um sensor sem contacto, podendo recolher informação sem perturbar o ambiente, para além de conseguir obter informação global sobre o mesmo [1]. O sensor Kinect para Xbox360 da Microsoft é um sensor de baixo custo, que possui capacidade de gerar mapas de profundidade, para além de permitir capturar imagens a cor.

A.1.1 Características do dispositivo Microsoft Kinect

Constituição

A Kinect é um dispositivo de fonte fechada, contudo pode-se encontrar informação sobre os seus componentes e o seu modo de funcionamento em diversos artigos. Nesta secção está-se interessado nos componentes diretamente responsáveis pela produção do mapa de profundidade e pelas imagens. Estes dispositivos são: (1) um projetor infravermelhos,(2) uma câmara infravermelho,(3) uma câmara RGB e (4) uma placa PS1080 [75] .

Câmara RGB

A câmara RGB, uma MT9M112 da Micron¹, possui resolução de 1280 por 1024 pixels, campo de visão de 63 graus na horizontal por 50 graus na vertical [76, 77, 78].

¹<http://www.chipworks.com/en/technical-competitive-analysis/resources/blog/teardown-of-the-microsoft-kinect-focused-on-motion-capture/>, consultado em fevereiro de 2014

Câmara Infravermelho

A câmara infravermelho, uma MT9M001C12STM da Micron², possui resolução de 1280 por 1024 pixels, campo de visão de 57 graus na horizontal por 45 graus na vertical [76, 79].

Projektor Infravermelho

O projetor de infravermelhos é composto por um laser de comprimento de onda 830 nanómetros, com estabilização de temperatura, cuja radiação atravessa uma rede de difração onde se divide num feixe de 3 por 3, atravessando de seguida uma segunda rede de difração que divide cada um destes feixes de forma a criar um padrão des-correlacionado quase periódico 211 por 165 pontos [75]. Este padrão é o mesmo para todas as Kinect. A utilização de um padrão descorrelacionado ajuda com o problema da correspondência e aumenta a robustez face a interferência [75].

Placa PS1080

A placa PS1080 processa o sinal de saída da câmara de infravermelhos com vista a calcular o mapa de profundidade. Esta placa suporta, opcionalmente, uma câmara RGB possuindo capacidade de computar a correspondência entre os pixels do mapa de profundidade e da imagem a cores [75]. O chip PS1080 na Kinect possui três canais de transmissão [75]:

1. Canal de controlo para receber e enviar mensagens de configuração e estado;
2. Canal de imagem, usado para enviar dados das câmaras RGB e infravermelhos mas não em simultâneo (para ambas as câmaras existem vários formatos disponíveis);
3. Canal de profundidade, utilizado para enviar imagens de disparidade no formato VGA à taxa de 30 imagens por segundo. O formato é de ponto fixo com 8 bits para representação de parte inteira e 3 para a parte fracionaria. Este modo apenas se encontra disponível quando a câmara infravermelhos se encontra configurada a 640x488 à taxa de 30 imagens por segundo.

²<http://www.chipworks.com/en/technical-competitive-analysis/resources/blog/teardown-of-the-microsoft-kinect-focused-on-motion-capture/>, consultado em fevereiro de 2014

Características mapa profundidade

A medição de distância é baseada num método de luz estruturada, recorrendo à triangulação dos pontos projetados e dos lidos pela câmara de infravermelhos [75], pelo que o erro expectável em medições de profundidade varia proporcionalmente ao quadrado da distância [80]. Devido à utilização do projetor e câmara infravermelho, iluminação infravermelha proveniente de fontes como Sol, lâmpadas incandescentes ou de halogéneo, pode afetar negativamente a medição de distâncias ao ponto de a inviabilizar.

A imagem de profundidade representa as distâncias entre mundo e o plano câmara infravermelhos de forma não linear, com maior resolução para distâncias mais baixas [72, 75]. Segundo [76] a resolução da distância (diferença entre dois valores consecutivos), é aproximadamente $2,73z^2 + 0,74z - 0,58[mm]$, com z em metros.

Mesmo com a Kinect fixa num ambiente estático os valores das distâncias em cada pixel variam com o tempo. Verifica-se maior amplitude de variação nas bordas dos objetos, observando-se que variam entre o valor da distância do fundo e o valor da distância do objeto [72].

Segundo [72], desde que a Kinect é ligada, é possível observar um período de cerca de 30 segundos até o valor da profundidade estabilizar no valor final, acontecendo um processo semelhante quando o sensor é rodado e apontado para um ambiente diferente.

Segundo a Microsoft³ estão acessíveis os seguintes fluxos de dados, com a restrição de não aceder em simultâneo aos fluxos de dados de imagens a cores e infravermelho:

1. Imagem da câmara a cores: (1) formato rgb com resolução de 640x480 pixels, à taxa de 30 imagens por segundo, (2) formato rgb com resolução de 1280x960 pixels, à taxa de 12 imagens por segundo, (3) formato yuv com resolução 640x480 pixels à taxa de 15 por segundo, (4) formato raw yuv com resolução 640x480 pixels à taxa de 15 por segundo, (5) formato raw bayer com resolução 1280x960 à taxa de 12 por segundo, (6) formato raw bayer com resolução 640x480 à taxa de 30 por segundo.
2. Imagem da câmara a infravermelho: (1) imagem de 16 bits (dos quais os últimos 6 não são utilizados, correspondendo a zeros) com resolução 640x480 à taxa de 30 por segundo.

³<http://www.microsoft.com/en-us/kinectforwindows/faq.aspx>; <http://msdn.microsoft.com/en-us/library/jj131027.aspx>; <http://msdn.microsoft.com/en-us/library/jj663793.aspx>; <http://msdn.microsoft.com/en-us/library/jj131028.aspx>; <http://msdn.microsoft.com/en-us/library/microsoft.kinect.depthimageformat.aspx> consultados em fevereiro de 2014.

3. Imagem de profundidade: (1) com resolução de 640x480 à taxa de 30 por segundo, (2) com resolução de 320x240 à taxa de 30 por segundo, (3) com resolução de 80x60 à taxa de 30 por segundo.

Esta página foi intencionalmente deixada em branco.

Apêndice B

Calibração da Câmera *Microsoft Kinect*

As Kinect são calibradas durante a produção com um algoritmo proprietário [75]. A calibração de fábrica não corrige distorções no mapa de profundidade [75][81] mas, como as lentes da câmaras de infravermelhos e cores da Kinect são de baixa distorção, o erro, mesmo nos cantos das imagens, não chega a ser superior a alguns píxeis¹. A calibração, segundo a Microsoft², é uma das razões apontadas para a disponibilização de imagens da câmara infravermelhos.

A exatidão dos mapas de profundidade da Kinect pode ser melhorada com a calibração dos parâmetros intrínsecos e extrínsecos das câmaras, podendo um mapa de profundidade de uma Kinect não calibrada apresentar um desvio de mais de 1cm². Uma câmara Kinect calibrada pode alcançar exatidão² de ± 1 mm.

¹http://wiki.ros.org/openni_kinect/kinect_accuracy, consultado em fevereiro de 2014.

²<http://msdn.microsoft.com/en-us/library/jj663793.aspx>, consultado em fevereiro de 2014.

Para calibrar a Kinect pode-se recorrer a bibliotecas ou aplicações já desenvolvidas³. Optou-se pelas aplicações disponibilizadas pela plataforma ROS visto possuir uma vasta comunidade de utilizadores e investigadores. Esta ferramenta⁴ utiliza o modelo de distorção de *Brown* [82], também conhecido como *plum_bob*. A ferramenta utilizada recorre à otimização não linear de Levenberg-Marquardt[83], de modo a encontrar os parâmetros do modelo⁵.

Considere-se o referencial formado pelos vetores $\vec{x}, \vec{y}, \vec{z}$ no qual $x \perp y, z = x \times y$, e os valores $X Y Z$, as distâncias segundo os versores $\hat{x}, \hat{y}, \hat{z}$. Considere ainda que o plano da câmara corresponde ao plano XoY. O modelo de distorção de Brown[82] é baseado no modelo de câmara *pinhole*, no qual se verifica a Equação (B.1), onde P_{pm} representa um ponto no plano XoY no qual os pontos do mundo são projetados, ver Figura B.1.

$$P_{pm} = \begin{bmatrix} P_{pm_x} \\ P_{pm_y} \end{bmatrix} = \frac{1}{Z} \begin{bmatrix} X \\ Y \end{bmatrix} \quad (\text{B.1})$$

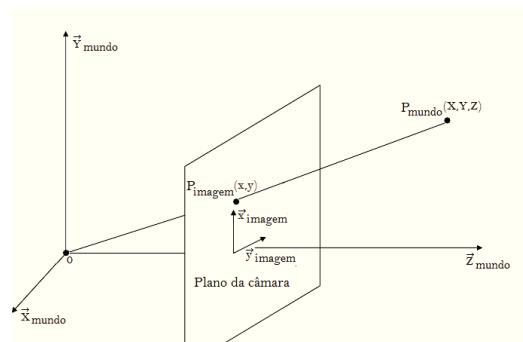


Figura B.1: Exemplo de projeção em câmaras *pinhole*, adaptado de <http://www.ics.uci.edu/~majumder/vispercep/cameracalib.pdf>

³<http://research.microsoft.com/en-us/um/people/zhang/Calib/>; http://www.vision.caltech.edu/bouguetj/calib_doc/; <http://code.google.com/p/procamcalib/>; <http://arthronav.isr.uc.pt/easycamcalib/>; <http://portal.survey.ntua.gr/main/labs/photo/staff/gkarras/faucal.html/>; <http://free-dee.org/software/vpc/>; http://www.dlr.de/rm/desktopdefault.aspx/tabid-3925/6084_read-9201/; <http://www.ee.oulu.fi/~dherrera/kinect/>; "http://wiki.ros.org/openni_launch/Tutorials"; http://docs.opencv.org/doc/tutorials/calib3d/camera_calibration/camera_calibration.html consultados em fevereiro de 2014.

⁴http://wiki.ros.org/image_pipeline/CameraInfo; http://wiki.ros.org/camera_calibration/Tutorials/MonocularCalibration; http://docs.ros.org/api/sensor_msgs/html/msg/CameraInfo.html consultados em fevereiro de 2014.

⁵http://wiki.ros.org/camera_calibration/Tutorials/MonocularCalibration; http://wiki.ros.org/camera_calibration; http://docs.opencv.org/modules/calib3d/doc/camera_calibration_and_3d_reconstruction.html consultados em fevereiro de 2014.

Segundo o modelo de distorção de Brown[82] a correspondência entre um ponto na imagem e um ponto no mundo é obtida através de:

1. aplicação de uma função de distorção às coordenadas do mundo segundo a Equação (B.2), constituída pela soma das coordenadas originais às funções de distorção radial, Equação (B.3), e tangencial, Equação (B.4);
2. multiplicação matricial entre a *MatrizCâmara* (B.5) e as coordenadas obtidas com a aplicação da distorção (B.6).

$$P_{pmd} = \begin{bmatrix} P_{pmd_x} \\ P_{pmd_y} \end{bmatrix} = P_{pm} + P_{pmd_{radial}} + P_{pmd_{tangencial}} \quad (B.2)$$

$$P_{pmd_{radial}} = \begin{bmatrix} P_{pmd_x} \\ P_{pmd_y} \end{bmatrix} = (k_1 r^2 + k_2 r^4) P_{pm}, \text{ onde } r^2 = P_x^2 + P_y^2 \quad (B.3)$$

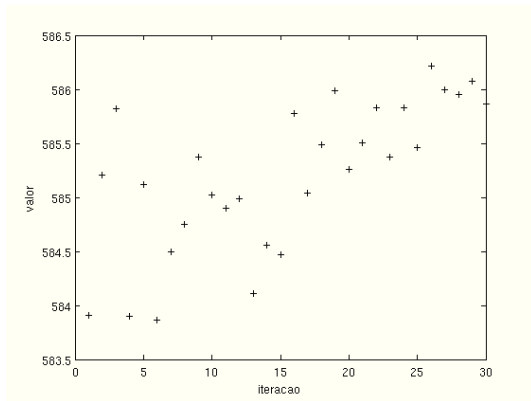
$$P_{pmd_{tangencial}} = \begin{bmatrix} 2k_3(P_x P_y) + k_4(r^2 + 2P_x^2) \\ 2k_4(P_x P_y) + k_3(r^2 + 2P_y^2) \end{bmatrix}, \text{ onde } r^2 = P_x^2 + P_y^2 \quad (B.4)$$

$$MatrizCâmara = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (B.5)$$

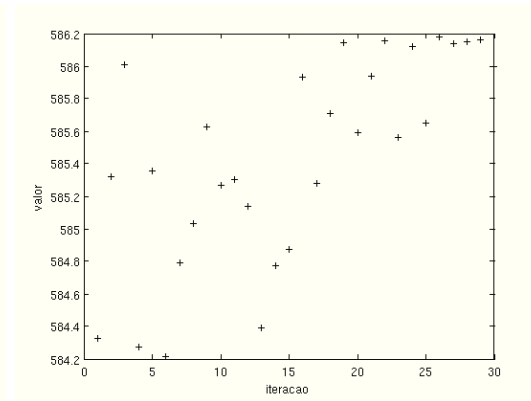
$$P_{imagem} = \begin{bmatrix} P_{imagem_x} \\ P_{imagem_y} \end{bmatrix} = Matriz\ Camera \begin{bmatrix} P_{pmd_x} \\ P_{pmd_y} \\ 1 \end{bmatrix} \quad (B.6)$$

Foram efetuadas 30 calibrações para ambas as câmaras da Kinect. Na Figura B.2 encontram-se os parâmetros referentes à *Matriz_Câmara* e na Figura B.3 os parâmetros referentes à função de distorção obtidos com as várias calibrações da câmara de infravermelhos.

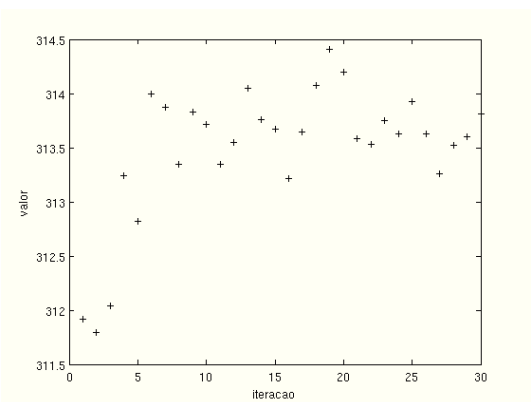
Através da comparação individual dos parâmetros não se consegue afirmar que estes converjam, contudo observa-se elevada correlação entre diversos parâmetros da câmara (ver Tabela B.1) como a correlação entre k_1 e k_2 ou entre f_x e f_y , evidenciada na Figura B.4.



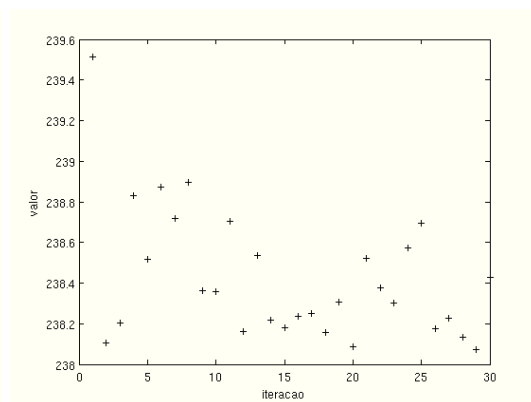
(a) Gráfico com parâmetros f_x .



(b) Gráfico com parâmetros f_y .

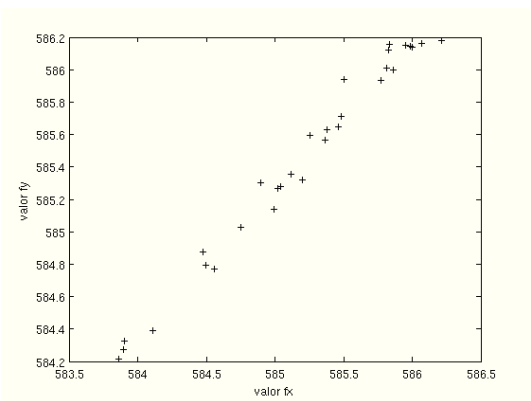


(c) Gráfico com parâmetros c_x .

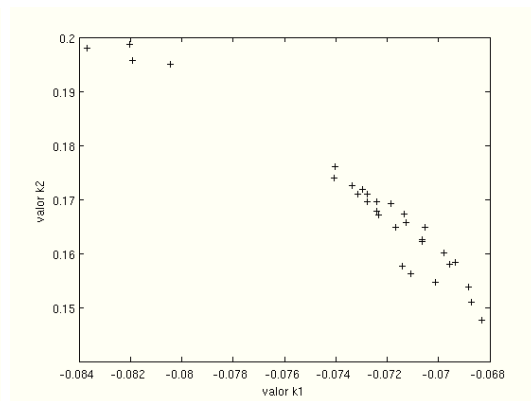


(d) Gráfico com parâmetros c_y .

Figura B.2: Gráficos de parâmetros referentes à $Matriz_{Câmara}$ das calibrações da câmara de infravermelhos.



(a) Gráfico relação entre f_x e f_y .



(b) Gráfico relação entre k_1 e k_2 .

Figura B.4: Gráficos que evidenciam a estreita relação entre parâmetros obtidos nas calibrações da câmara de infravermelhos.

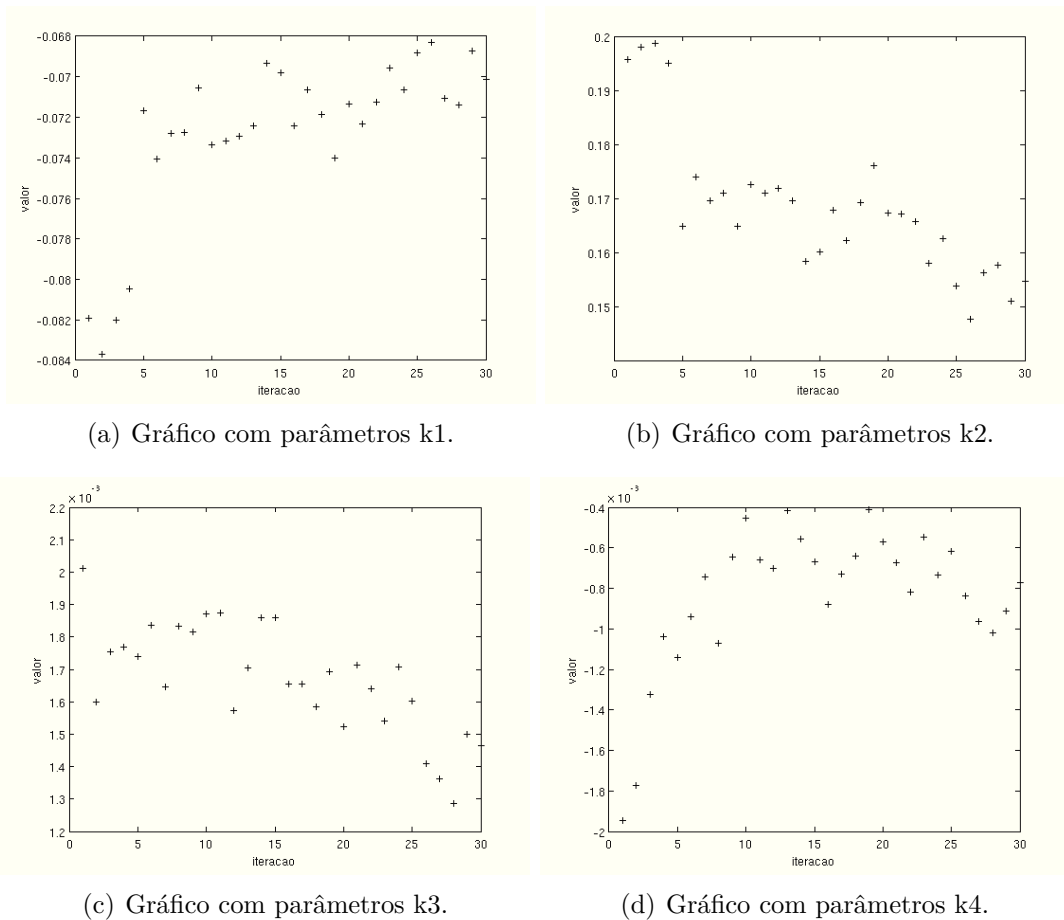


Figura B.3: Gráficos de parâmetros referentes aos parâmetros de distorção das calibrações da câmara de infravermelhos.

Tabela B.1: Correlação entre parâmetros de distorção das calibrações da câmara de infravermelhos.

Matriz Correlações	k1	k2	k3	k4	fx	fy	cx	cy
k1	1							
k2	-0,97	1						
k3	-0,36	0,49	1					
k4	0,72	-0,59	-0,12	1				
fx	0,36	-0,44	-0,67	0,15	1			
fy	0,36	-0,43	-0,63	0,16	0,99	1		
cx	0,75	-0,64	-0,21	0,9	0,09	0,1	1	
cy	-0,35	0,37	0,59	-0,35	-0,62	-0,58	-0,22	1

De modo a comparar os resultados obtidos com cada uma das calibrações utilizaram-se as métricas:

1. Equação (B.7), média do integral em toda a área da imagem do quadrado da distância (euclidiana) entre os dois pontos $d_{i,2}(x, y)$ no referencial da imagem, resultantes da aplicação de duas transformações de coordenadas imagem para coordenadas mundo, utilizando dois conjuntos de parâmetros intrínsecos da câmara diferentes. O resultado encontra-se na unidade $[metro^2/metro^2]$.

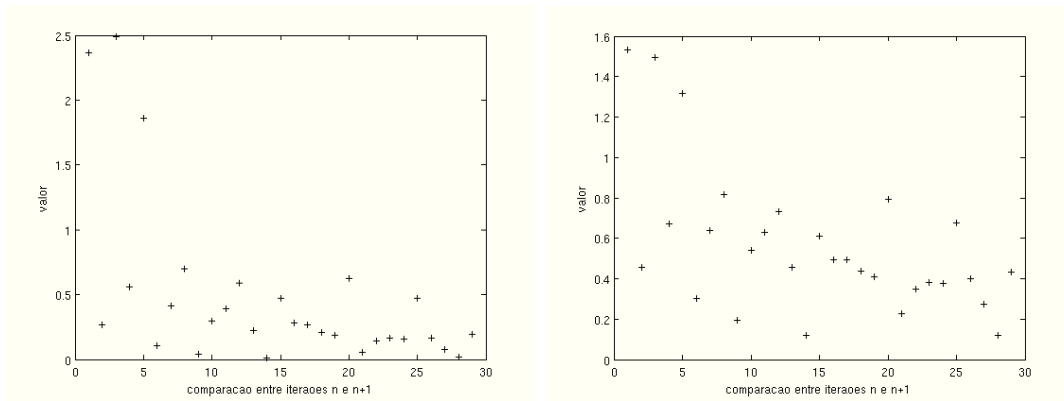
2. Equação (B.8), média da distância euclidiana entre dois pontos na imagem, resultantes de duas transformações de coordenadas mundo para coordenadas imagem, utilizando dois conjuntos diferentes de parâmetros intrínsecos da câmara, de uma grelha de pontos igualmente espaçados ($\Delta y = (y_{max} - y_{min})/(numero_linhas)$ e $\Delta x = (x_{max} - x_{min})/(numero_colunas)$) segundo os versores \hat{x} e \hat{y} referencial do mundo, o resultado encontra-se na unidade $[pixel]$.

3. Média da distância euclidiana entre dois pontos no mundo $d_{m,2}(x, y)$, resultantes de duas transformações de coordenadas mundo para coordenadas imagem, utilizando dois conjuntos de parâmetros intrínsecos da câmara diferentes, de uma grelha de pontos igualmente espaçados ($\Delta y = (y_{max} - y_{min})/(numero_linhas)$ e $\Delta x = (x_{max} - x_{min})/(numero_colunas)$) segundo os versores \hat{x} e \hat{y} referencial do mundo B.9, o resultado encontra-se na unidade $[metro]$.

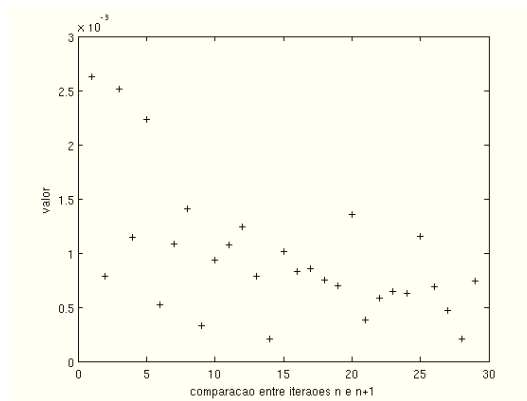
$$\frac{1}{(y_{max} - y_{min}) * (x_{max} - x_{min})} \int_{y_{min}}^{y_{max}} \int_{x_{min}}^{x_{max}} d_{i,2}(x_1, y_1, x_2, y_2)^2 dx dy \quad (B.7)$$

$$\frac{1}{numero_colunas * numero_linhas} \sum_{y=y_{min}+a\Delta y}^{y_{max}} \sum_{x=x_{min}+a\Delta x}^{x_{max}} d_{i,2}(x_1, y_1, x_2, y_2), \text{ onde } a \in \mathbb{N} \quad (B.8)$$

$$\frac{1}{numero_colunas * numero_linhas} \sum_{y=y_{min}+a\Delta y}^{y_{max}} \sum_{x=x_{min}+a\Delta x}^{x_{max}} d_{m,2}(x_1, y_1, x_2, y_2), \text{ onde } a \in \mathbb{N} \quad (B.9)$$



(a) Gráfico distância em píxeis, calculada através de integral. (b) Gráfico distancia média em píxeis.



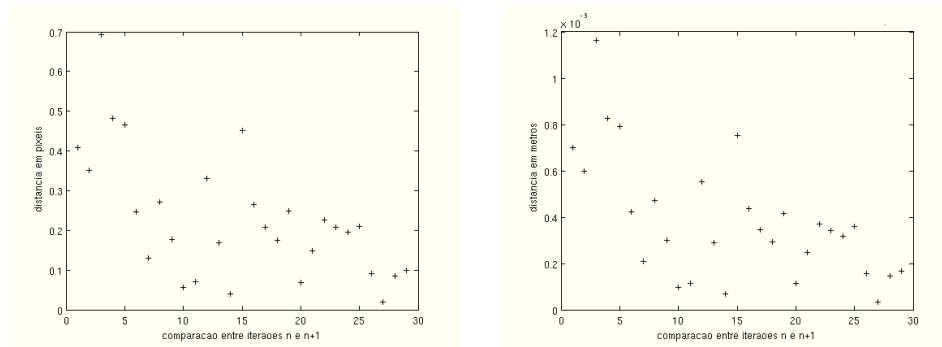
(c) Gráfico distância média em metros.

Figura B.5: Valores associados às métricas de distância entre duas conversões com parâmetros distintos de um mesmo ponto.

Pode-se ainda substituir o cálculo da distância entre dois pontos por outro no qual se faça o deslocamento de todos os pontos pela diferença média segundo os eixos \hat{x} e \hat{y} , considerando-se neste caso que um desvio fixo pode ser compensado através de uma etapa posterior de calibração entre câmaras, não sendo necessário pesar a sua contribuição para aferir a convergência da compensação da distorção, estes valores encontram-se expostos na Figura B.6.

Observa-se que, após as primeiras cinco calibrações (que contêm vistas do padrão xadrez longe, perto e com gama de inclinações +/- completa) as calibrações convergem. Este método foi repetido para os resultados obtidos com a câmara a cores da qual se obtiveram resultados semelhantes. Os parâmetros correspondentes podem ser consultados nas Tabelas B.

De modo a escolher os parâmetros intrínsecos a utilizar, foram efetuadas medições,



(a) Gráfico distância média em pixels sem considerar deslocamento médio.

(b) Gráfico distância média em metros sem considerar deslocamento médio.

Figura B.6: Valores associados às métricas de distância entre duas conversões com parâmetros distintos de um mesmo ponto, sem considerar o desvio médio.

Tabela B.2: Correlação entre parâmetros de distorção das calibrações da câmara de cores.

Matriz Correlações	k1	k2	k3	k4	fx	fy	cx	cy
k1	1							
k2	-0,95	1						
k3	-0,43	0,49	1					
k4	-0,03	-0,02	0,58	1				
fx	0,2	-0,11	0,31	0,35	1			
fy	0,21	-0,11	0,34	0,37	1	1		
cx	0,33	-0,47	-0,35	-0,05	-0,71	-0,69	1	
cy	-0,69	0,68	0,82	0,52	0,08	0,09	-0,33	1

para cada conjunto de parâmetros obtidos. A primeira medição consiste no lado de quadrados (25mm) do tabuleiro com padrão de xadrez, com o plano deste paralelo ao plano da câmara, a uma distância de 82cm. A métrica utilizada para comparar resultados foi o desvio padrão. Para a Câmara de cores foi escolhida a calibração 9 devido a conter o menor valor (0,234 mm). A escolha da calibração da câmara de infravermelhos possui outra métrica associada, que mede a qualidade da profundidade num mapa de profundidade. Para avaliar a qualidade do mapa de profundidade com a câmara fixa, retiraram-se imagens de uma porção de uma mesa de 40cm de lado, a 112cm da câmara, calculando o plano que melhor descreve os pontos obtidos e a distância dos pontos em relação ao plano e o desvio padrão desta medida. Devido ao desvio na métrica associada ao quadrado do padrão xadrez ser muito menor que o associado à distância ao plano, escolheu-se a calibração que possui menor desvio-padrão da distância entre pontos e plano, a calibração 22. Para a calibração 22 foram medidos os valores (3,156 mm) e (0,2455 mm) de desvio padrão para lado do quadrado e distância a plano, respectivamente. Na Figura B.7 pode ser visto o histograma da distância a plano. Os valores para cada calibração podem ser vistos na Tabela B.7.

Tabela B.3: Câmara de Cores - Parâmetros Distorção.

Nº	k1	k2	k3	k4
1	0,1633409506925	-0,2926560485495	-0,0012211321453842	-0,003403255055948
2	0,1675374745533	-0,3010097884487	-0,0014717529997787	-0,003597476810313
3	0,1663664908340	-0,2903022004988	-0,0007636713610541	-0,002672395869643
4	0,1684564555991	-0,2949217947823	-0,0007206334572256	-0,003169418608146
5	0,1678955029539	-0,2985884802435	-0,0008969304082416	-0,002998584841660
6	0,1686164435171	-0,3001600456822	-0,0006823585572700	-0,002506743970161
7	0,1673246478877	-0,2947755441404	-0,0007825446191577	-0,002860064353947
8	0,1696933580821	-0,3030203457584	-0,0008619985385607	-0,002295057089033
9	0,1704996714759	-0,3087243582770	-0,0009192640538337	-0,002003394361875
10	0,1636419123614	-0,2898196206916	-0,0009788472828155	-0,002732039172152
11	0,1633857941309	-0,2851111777077	-0,0009547351851237	-0,003130015343920
12	0,1697348223806	-0,3028149122149	-0,0009516164789581	-0,002644106552255
13	0,1668272250046	-0,2931616564558	-0,0006969301326082	-0,002530562677553
14	0,1684138800527	-0,2994246718890	-0,0008847571759091	-0,002701552843169
15	0,1662666749838	-0,2921107092726	-0,0007271168898176	-0,002267465948158
16	0,1643845353227	-0,2905020167896	-0,0010803087073467	-0,002521998074311
17	0,1610842844677	-0,2854988852069	-0,0005165713281091	-0,002327348768199
18	0,1605996180563	-0,2836670727539	-0,0004844834697512	-0,002549992078906
19	0,1602560070203	-0,2834967873657	-0,0005674674414087	-0,002626532460747
20	0,1609461871212	-0,2844507968408	-0,0006098502573133	-0,002614407988838
21	0,1607506773471	-0,2850707965228	-0,0007141350564879	-0,002448862687006
22	0,1617345666626	-0,2870439898849	-0,0005975675388592	-0,002116065515181
23	0,1613758022017	-0,2854627132713	-0,0008169172081530	-0,002534727325761
24	0,1623850141327	-0,2868737688432	-0,0006754555226427	-0,002622547867689
25	0,1617751701317	-0,2880156041388	-0,0006331013000737	-0,002399658943613
26	0,1617751701317	-0,2880156041388	-0,0006331013000737	-0,002399658943613
27	0,1611218246926	-0,2854392678712	-0,0006334872380043	-0,002385143290175
28	0,1594545949489	-0,2806352234551	-0,0007311976317256	-0,002870753662988
29	0,1608597225697	-0,2819975093498	-0,0006453752381851	-0,002857467375787
30	0,1594006071174	-0,2789389040761	-0,0009493274458246	-0,002907527417943

Tabela B.4: Câmara de Cores - Parâmetros da Matriz Câmara.

N	fx	fy	cx	cy
1	521,923265078651	522,157209373000	312,730896586358	266,372052301797
2	526,732854259833	526,862521235395	309,398739893563	264,941671428148
3	526,176218332563	526,556568396382	309,329285483973	266,264202480228
4	527,252761116923	527,474131960464	309,071126521341	266,159381293289
5	526,794964753310	527,064203283571	309,175296291816	266,157767201046
6	526,958435687568	527,402032881058	310,351018111361	266,451667441669
7	526,353884074222	526,719948262657	309,088960231995	266,877245604016
8	526,456912291836	526,928759144544	310,180084899347	266,471800646592
9	526,630592731850	526,859342092662	310,479777178652	266,434506986859
10	525,740814592154	525,968459951937	308,795715445337	266,664428056718
11	527,017518298592	527,356228533124	308,517383601476	266,700151401545
12	527,299447390834	527,428111497989	309,117342608218	266,684152333922
13	526,841958939679	527,212660307704	309,282245420069	266,930018659056
14	527,501385335420	527,828758193174	309,034146334343	266,818378518514
15	527,071068567402	527,313068550519	309,341991303825	266,930240508121
16	526,616592781103	526,795345225871	309,039609813659	266,660881050943
17	526,704505609655	526,981151322915	309,185346786017	267,582147910652
18	526,430048956147	526,720451574654	308,774995275938	267,556073063195
19	526,553240268686	526,880069024446	308,853048510210	267,471059228738
20	526,653076080522	526,896662521904	308,952445380634	267,097032776619
21	526,660449215004	526,936342544978	309,123343905228	267,258991466675
22	526,605513815928	526,800261255896	309,494852257805	267,245742632694
23	526,819871566845	527,058967418240	308,914726165348	267,071788609433
24	526,524208911080	526,849634182951	309,035204373411	267,377161582359
25	526,665119247063	527,011183771279	309,520648302107	267,196864247699
26	526,191801936682	526,503678385572	308,737001671252	267,658274030509
27	526,924485571784	527,158873546786	309,272050498311	267,317501836375
28	526,457181371563	526,750450575588	308,879277169594	267,345335208900
29	526,305998258346	526,518908952653	308,652235938397	267,493566855384
30	526,332303563661	526,605969289252	308,501664228327	266,691110340446

Tabela B.5: Câmara de cores - métricas convergência.

		Referencial Mundo [m]		Referencial Imagem [pxl]		
Nº	Nº	desvio máximo	desvio médio	média do integral do desvio quadrado	desvio máximo	desvio médio
1	2	$1,815 \times 10^{-02}$	$7,578 \times 10^{-03}$	18,590	7,353	4,012
2	3	$6,248 \times 10^{-03}$	$2,756 \times 10^{-03}$	2,182	2,867	1,485
3	4	$2,948 \times 10^{-03}$	$9,877 \times 10^{-04}$	0,405	1,313	0,546
4	5	$2,752 \times 10^{-03}$	$5,663 \times 10^{-04}$	0,125	1,262	0,318
5	6	$4,180 \times 10^{-03}$	$2,529 \times 10^{-03}$	1,797	1,777	1,348
6	7	$3,938 \times 10^{-03}$	$2,700 \times 10^{-03}$	2,043	1,811	1,423
7	8	$4,231 \times 10^{-03}$	$2,445 \times 10^{-03}$	1,689	2,067	1,307
8	9	$2,527 \times 10^{-03}$	$7,080 \times 10^{-04}$	0,148	1,020	0,385
9	10	$5,968 \times 10^{-03}$	$3,604 \times 10^{-03}$	3,707	2,653	1,904
10	11	$5,075 \times 10^{-03}$	$1,284 \times 10^{-03}$	0,601	2,126	0,708
11	12	$4,114 \times 10^{-03}$	$1,360 \times 10^{-03}$	0,548	1,622	0,731
12	13	$2,222 \times 10^{-03}$	$7,166 \times 10^{-04}$	0,153	0,869	0,380
13	14	$1,428 \times 10^{-03}$	$7,102 \times 10^{-04}$	0,176	0,733	0,376
14	15	$2,826 \times 10^{-03}$	$8,811 \times 10^{-04}$	0,235	0,866	0,466
15	16	$3,112 \times 10^{-03}$	$9,935 \times 10^{-04}$	0,336	1,367	0,545
16	17	$3,117 \times 10^{-03}$	$1,967 \times 10^{-03}$	1,088	1,367	1,051
17	18	$1,589 \times 10^{-03}$	$8,968 \times 10^{-04}$	0,229	0,688	0,472
18	19	$4,568 \times 10^{-04}$	$2,402 \times 10^{-04}$	0,016	0,192	0,122
19	20	$1,274 \times 10^{-03}$	$7,580 \times 10^{-04}$	0,159	0,476	0,400
20	21	$1,005 \times 10^{-03}$	$4,808 \times 10^{-04}$	0,066	0,414	0,258
21	22	$1,794 \times 10^{-03}$	$8,588 \times 10^{-04}$	0,207	0,735	0,458
22	23	$3,244 \times 10^{-03}$	$1,355 \times 10^{-03}$	0,525	1,316	0,725
23	24	$1,480 \times 10^{-03}$	$6,796 \times 10^{-04}$	0,127	0,489	0,356
24	25	$1,825 \times 10^{-03}$	$1,064 \times 10^{-03}$	0,321	0,842	0,570
25	26	$2,570 \times 10^{-03}$	$1,882 \times 10^{-03}$	0,995	1,334	0,991
26	27	$2,684 \times 10^{-03}$	$1,436 \times 10^{-03}$	0,592	1,223	0,737
27	28	$2,032 \times 10^{-03}$	$1,010 \times 10^{-03}$	0,308	0,957	0,531
28	29	$9,430 \times 10^{-04}$	$5,280 \times 10^{-04}$	0,078	0,324	0,279
29	30	$2,782 \times 10^{-03}$	$1,645 \times 10^{-03}$	0,761	1,114	0,877

Tabela B.6: Câmera de cores - métricas convergência sem considerar o desvio médio.

		Referencial Mundo [m]		Referencial Imagem [pxl]	
Nº	Nº	desvio máximo	desvio médio	desvio máximo	desvio médio
1	2	$1,123 \times 10^{-02}$	$4,029 \times 10^{-03}$	3,701	2,169
2	3	$4,098 \times 10^{-03}$	$4,880 \times 10^{-04}$	1,604	0,249
3	4	$2,236 \times 10^{-03}$	$8,629 \times 10^{-04}$	0,944	0,485
4	5	$2,918 \times 10^{-03}$	$5,369 \times 10^{-04}$	1,121	0,302
5	6	$1,754 \times 10^{-03}$	$3,156 \times 10^{-04}$	0,484	0,148
6	7	$1,299 \times 10^{-03}$	$5,096 \times 10^{-04}$	0,427	0,255
7	8	$2,343 \times 10^{-03}$	$2,732 \times 10^{-04}$	0,778	0,137
8	9	$2,014 \times 10^{-03}$	$1,672 \times 10^{-04}$	0,693	0,092
9	10	$2,972 \times 10^{-03}$	$8,921 \times 10^{-04}$	0,776	0,450
10	11	$4,524 \times 10^{-03}$	$1,204 \times 10^{-03}$	1,791	0,659
11	12	$3,097 \times 10^{-03}$	$3,729 \times 10^{-04}$	0,998	0,192
12	13	$2,203 \times 10^{-03}$	$3,221 \times 10^{-04}$	0,518	0,168
13	14	$1,067 \times 10^{-03}$	$4,765 \times 10^{-04}$	0,406	0,255
14	15	$2,264 \times 10^{-03}$	$3,962 \times 10^{-04}$	0,430	0,208
15	16	$2,203 \times 10^{-03}$	$5,389 \times 10^{-04}$	0,869	0,296
16	17	$1,322 \times 10^{-03}$	$2,064 \times 10^{-04}$	0,355	0,100
17	18	$7,778 \times 10^{-04}$	$2,360 \times 10^{-04}$	0,230	0,116
18	19	$3,242 \times 10^{-04}$	$9,001 \times 10^{-05}$	0,125	0,046
19	20	$6,018 \times 10^{-04}$	$1,088 \times 10^{-04}$	0,116	0,054
20	21	$5,277 \times 10^{-04}$	$6,656 \times 10^{-05}$	0,246	0,039
21	22	$1,061 \times 10^{-03}$	$1,416 \times 10^{-04}$	0,327	0,070
22	23	$1,959 \times 10^{-03}$	$2,451 \times 10^{-04}$	0,619	0,131
23	24	$1,099 \times 10^{-03}$	$1,759 \times 10^{-04}$	0,263	0,086
24	25	$1,005 \times 10^{-03}$	$1,026 \times 10^{-04}$	0,316	0,053
25	26	$9,235 \times 10^{-04}$	$4,080 \times 10^{-04}$	0,355	0,204
26	27	$1,304 \times 10^{-03}$	$5,962 \times 10^{-04}$	0,520	0,304
27	28	$1,148 \times 10^{-03}$	$4,237 \times 10^{-04}$	0,455	0,215
28	29	$4,232 \times 10^{-04}$	$6,848 \times 10^{-05}$	0,064	0,034
29	30	$1,489 \times 10^{-03}$	$1,079 \times 10^{-04}$	0,270	0,047

Nas Figuras B.7 e B.9 pode-se observar, para a conversão de pontos do mundo para a imagem com os parâmetros iniciais e com os parâmetros referentes à calibração, a diferença entre a posição de pontos e o divergente da distância entre os pontos.

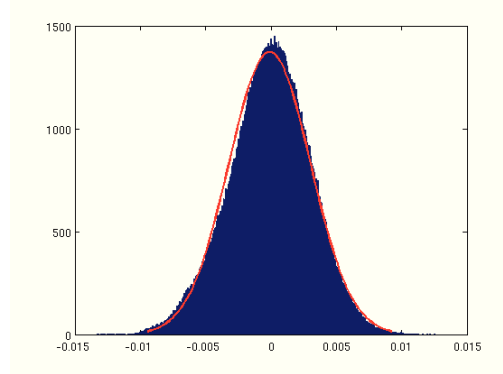
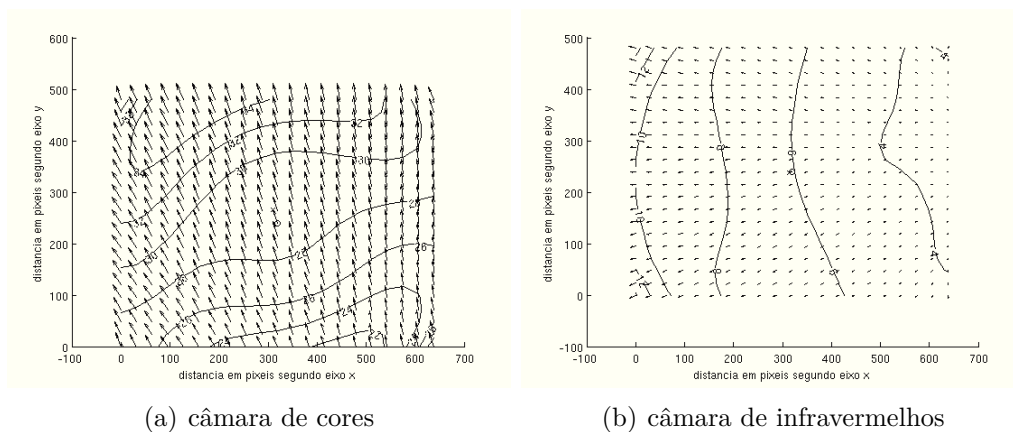


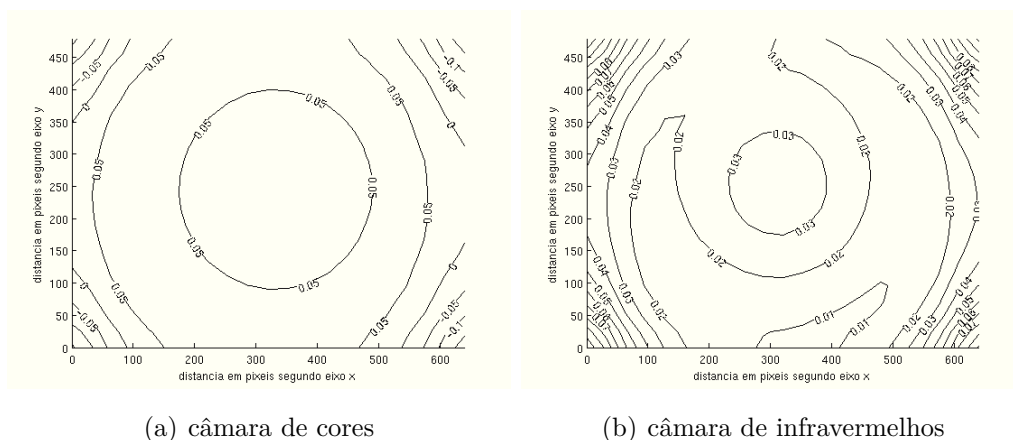
Figura B.7: Histograma de distância de pontos a plano para a calibração 22.



(a) câmara de cores

(b) câmara de infravermelhos

Figura B.8: Diferença entre conversão de pontos com os parâmetros iniciais e parâmetros relativos à calibração.



(a) câmara de cores

(b) câmara de infravermelhos

Figura B.9: Gradiente da diferença entre conversão de pontos com os parâmetros iniciais e parâmetros relativos à calibração.

Tabela B.7: Medições utilizadas para comparar calibrações

Nº	Desvio padrão distância a plano	Desvio padrão lado quadrado do padrão xadrez	
		câmara infravermelho	câmara cores
0	0,003346534963453550	0,0002247250	0,0003444470
1	0,003209899089173980	0,0002206050	0,0002495430
2	0,003464394118855880	0,0002258840	0,0002411780
3	0,003265334714408390	0,0002233920	0,0002460900
4	0,003283921886612250	0,0002323460	0,0002492890
5	0,003241768261951180	0,0002403060	0,0002462670
6	0,003270314725694560	0,0002382570	0,0002491010
7	0,003176916338798250	0,0002292170	0,0002455230
8	0,003196109493898920	0,0002431530	0,0002388500
9	0,003248757065588030	0,0002318060	0,0002339490
10	0,003251994211146480	0,0002460910	0,0002407900
11	0,003188688862688590	0,0002393500	0,0002462310
12	0,003210132392222800	0,0002300570	0,0002384020
13	0,003193178588392190	0,0002486470	0,0002464650
14	0,003161263622971320	0,0002476470	0,0002412660
15	0,003188207802268280	0,0002356390	0,0002377000
16	0,003221907951155690	0,0002394960	0,0002420660
17	0,003176928624968710	0,0002337850	0,0002420190
18	0,003182088023248810	0,0002242860	0,0002437010
19	0,003188979188674970	0,0002270920	0,0002427590
20	0,003181057171964800	0,0002430530	0,0002476950
21	0,003161830660097490	0,0002450630	0,0002386040
22	0,003156596126319290	0,0002454950	0,0002371640
23	0,003158567168254090	0,0002380290	0,0002402940
24	0,003173253356859880	0,0002544480	0,0002467700
25	0,003206535120314350	0,0002508150	0,0002413990
26	0,003169405413353890	0,0002479760	0,0002494350
27	0,003156786229799450	0,0002604690	0,0002367030
28	0,003162654161518430	0,0002611110	0,0002441550
29	0,003164173465323830	0,0003106210	0,0002480120
30	0,003176555639733950	0,0004127400	0,0002458100

Parâmetros Extrínsecos

A calibração de parâmetros extrínsecos consiste na matriz que transforma, através de rotação, translação e alteração de magnitude de escala, os pontos relativos ao mapa de profundidade nos pontos correspondentes visto pela câmara a cores.

A calibração de parâmetros extrínsecos foi feita com as ferramentas do ROS, seguindo-se a metodologia presente em [84]. Os valores obtidos encontram-se na Tabela B.8. Os valores foram testados através da visualização de objetos em cima de uma mesa. Neste teste a mesa e os objetos possuem cor distinta. Ao detetar com altura correspondente à mesa pontos com a cor do objeto, é efetuado um ajuste no plano XoY, utilizando a Equação (B.10), obtendo-se os parâmetros expostos na Tabela B.9.

$$Distância_{metros} = Distância_{pixeis} \frac{Distância\ à\ câmara}{Distancia\ Focal} \quad (B.10)$$

Tabela B.8: Parâmetros Extrínsecos.

Parâmetro	Valor associado a imagem de cores	Valor associado a mapa de profundidade
coordenada x	$1.20269999661 \times 10^{-08}$	-0.0249369309833
coordenada y	$1.03136427271 \times 10^{-08}$	-0.00253082093999
coordenada z	$-1.60632140015 \times 10^{-08}$	0.00117353132618
quaternião w	1	0.999995058006
quaternião x	$1.04820968153 \times 10^{-08}$	0.00138451700316
quaternião y	$-4.70035400635 \times 10^{-09}$	-0.00130496463158
quaternião z	$5.04776029778 \times 10^{-09}$	-0.00250282723795

Tabela B.9: Parâmetros Extrínsecos.

Parâmetro	Valor associado a imagem de cores	Valor associado a mapa de profundidade
coordenada x	0	-0.03393
coordenada y	0	-0.00153
coordenada z	0	0.00117
quaternião w	1	1
quaternião x	0	0
quaternião y	0	0
quaternião z	0	0

Esta página foi intencionalmente deixada em branco.

Apêndice C

Comunicação entre Controlador e Computador

A comunicação entre o controlador e computador foi idealizada tendo em conta modularidade, expansibilidade e robustez. Na Figuras C.1 e C.2 pode-se observar a troca de informação entre computador e controlador. Para cada envio de dados existe um eco. Este eco visa informar o dispositivo emissor que a informação foi recebida e convertida corretamente. Na Tabela C.1 encontram-se os valores definidos para eco em caso de sucesso e insucesso. Estes valores são diferentes para palavras chave ou informação, deste modo identificam-se os problemas:(1) interpretação de um envio de informação como vários envios (exemplo: múltiplos caracteres terminadores numa sequência de texto que só deveria possuir um no final) e (2) perda de um envio de informação.

Tabela C.1: Valores de Eco.

Valor	Descrição
"BOM"	informação recebida foi convertida com sucesso
"MAU"	informação recebida não foi convertida com sucesso
"CHVBOM"	palavra chave recebida foi convertida com sucesso
"CHVMAU"	palavra chave recebida não foi convertida com sucesso

A comunicação encontra-se dividida nos módulos: (1) envio e receção de dados, (2) conversão de dados e (3) interpretação.

O módulo de envio e receção de dados encontra-se separado dos restantes de modo a obter-se independência, por parte da aplicação, da tecnologia de comunicação. Operações de substituição de tecnologia de comunicação, por exemplo, para tecnologias já implementadas num ambiente fabril, encontram-se facilitadas. Com o objetivo de aumentar a expansibilidade, optou-se por enviar dados na forma de texto, sendo utili-

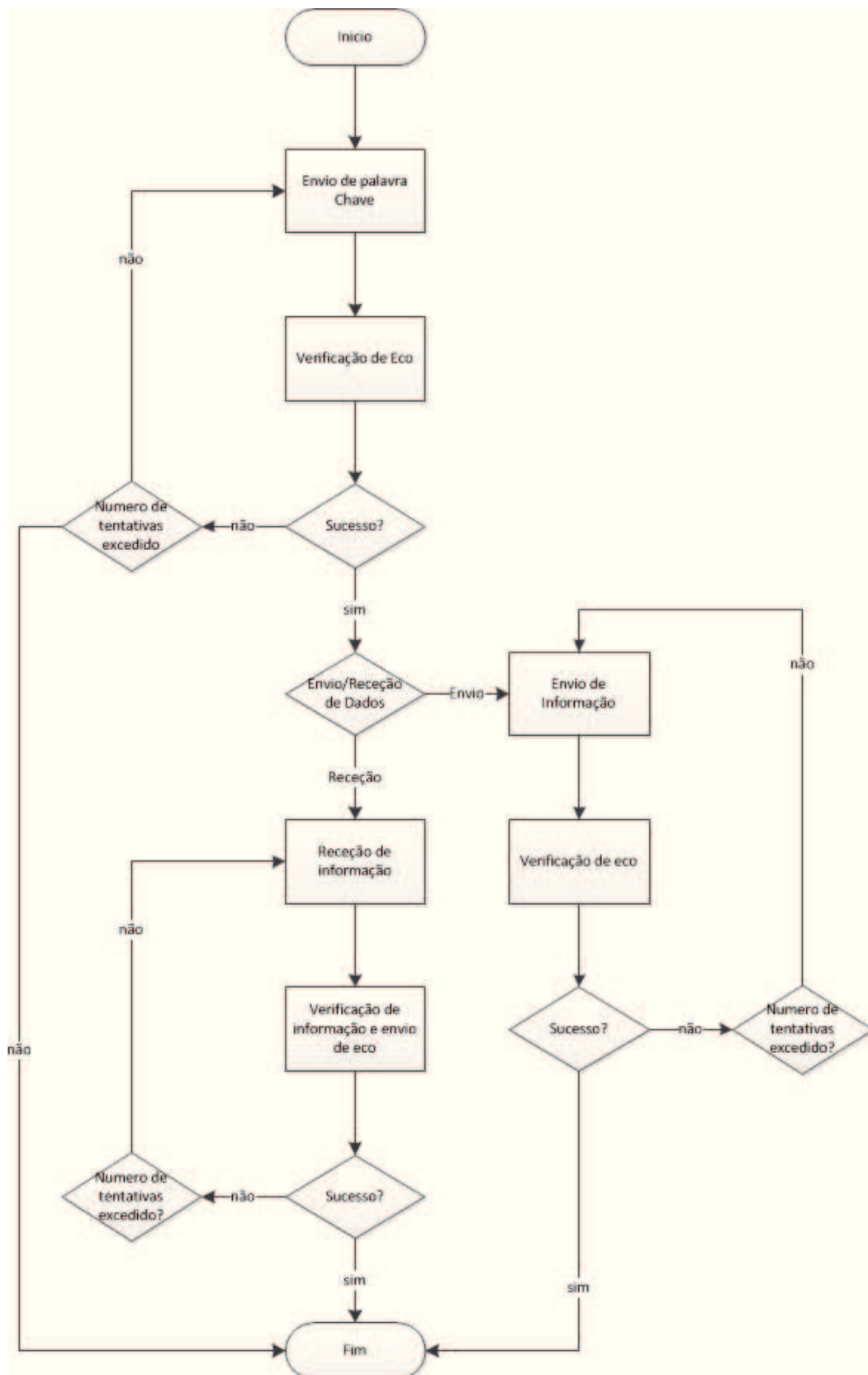


Figura C.1: Fluxograma descritivo do processo de comunicação por parte do computador.

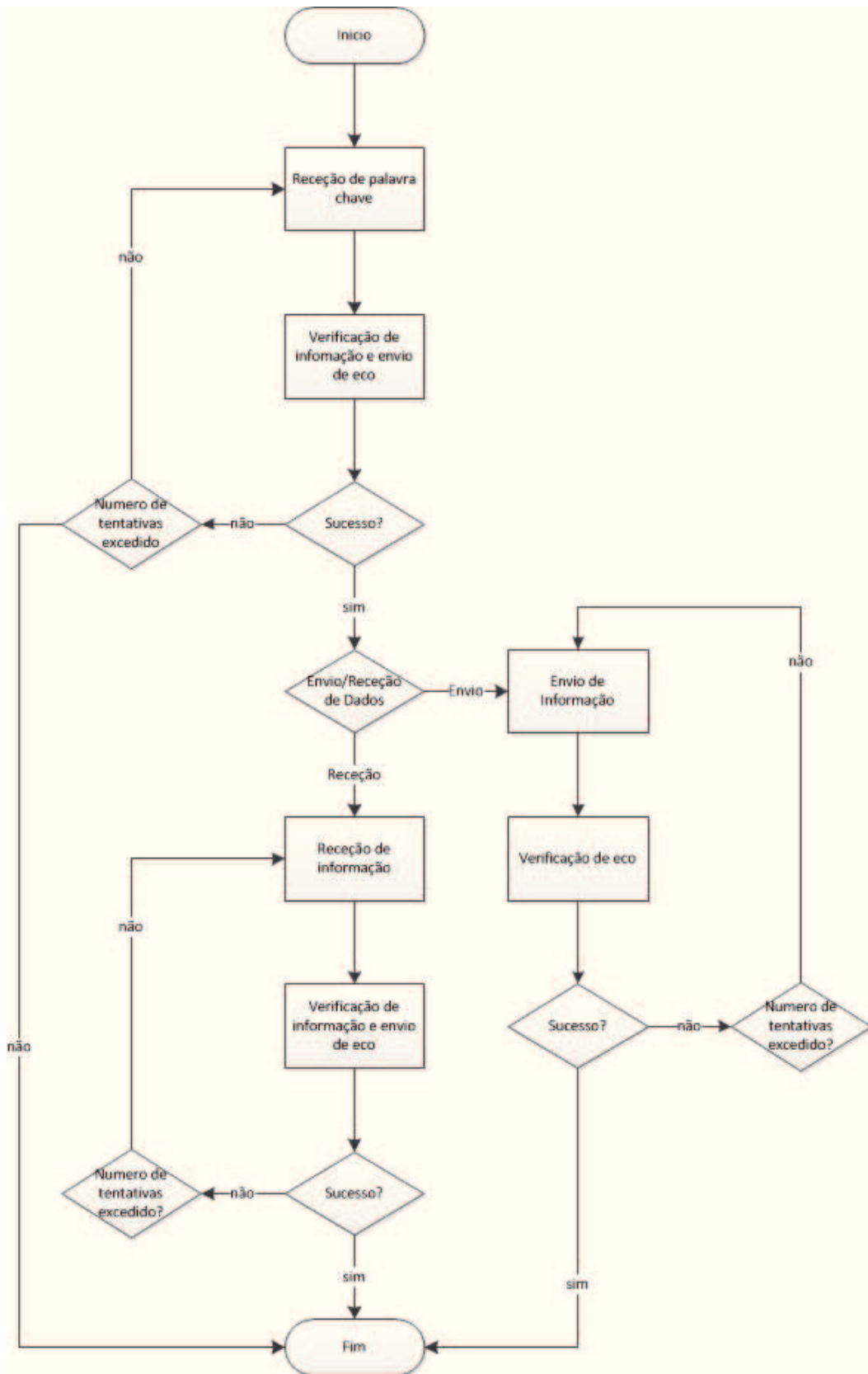


Figura C.2: Fluxograma descritivo do processo de comunicação por parte do controlador.

zado um terminador nulo para indicar o seu fim. Isto permite descrever tipos de dados genéricos, de comprimento genérico, desde que se respeite os limites físicos associados ao sistema no qual está implementado (com o controlador utilizado o comprimento máximo para uma sequência de texto é 80 caracteres). O módulo de envio e recepção de dados possui as funções de leitura e escrita de texto. Estas funções contêm a opção de tempo de espera máximo após o qual se assume que não chegou informação. Esta funcionalidade permite evitar situações nas quais a comunicação cesse por o controlador e o computador estarem ambos à espera de dados. O módulo de envio e recepção de dados deve possuir, quando aplicável, funções de configuração, abertura para leitura de dados, abertura para escrita de dados e fecho. As funções de abertura e fecho, quando necessárias são automaticamente chamadas pelas funções de leitura e escrita de texto. A implementação atual utiliza comunicação série, sendo que a componente associada ao computador, sem a interface ROS, utilizável em sistemas operativos Windows e Linux.

O módulo de conversão de dados recebe informação sobre o tipo de dados através do módulo de interpretação. A implementação atual contém os tipos de dados (1) texto, (2) número inteiro, (3) número decimal, (4) variável booleana, (5) *confdata*¹, (6) *pos*¹, (7) *orient*¹, (8) *pose*¹, (9) *extjoint*¹, (10) *robjoint*¹, (11) *wobjdata*¹. O módulo de conversão possui identificação de erros no caso dos dados recebidos não serem convertíveis para os desejados.

A interpretação consiste na utilização de palavras chave associadas ao sentido da informação, ao tipo de dados enviado e à interpretação referente a esses dados. Na Tabela C.2 encontram-se as palavras chave utilizadas, o sentido de informação, tipos de dados e a interpretação associadas.

Tabela C.2: Palavras Chave.

Texto	Sentido de Informação	Descrição
ENVPOS	Computador→ Controlador	Envio de Posição X,Y,Z para o controlador robótico.
ENVORIENT	Computador→ Controlador	Envio de orientação qw qx qy qz para o controlador robótico.

¹consultar documento da ABB "Technical reference manual RAPID Instructions, Functions and Data types"

ENVCONFDATA	Computador→ Controlador	Envio de informação sobre configuração do manipulador robótico para o controlador robótico.
ENVEXTJOINT	Computador→ Controlador	Envio de informação referente às juntas do manipulador robótico para o controlador robótico.
ENVROBJOINT	Computador→ Controlador	Envio de informação referente eixos externos ao manipulador robótico para o controlador robótico.
ENVPOSE	Computador→ Controlador	Envio de Posição X,Y,Z e Orientação qw,qx,qy,qz para o manipulador robótico.
ENVWOBJDATA	Controlador→ Computador	Envio de definição de objeto de trabalho para o controlador robótico.
RECPOS	Controlador→ Computador	Envio de Posição X,Y,Z para o computador.
RECORIENT	Controlador→ Computador	Envio de orientação qw qx qy qz para o computador.
RECCONFDATA	Controlador→ Computador	Envio de informação sobre configuração do manipulador robótico para o computador.
RECEXTJOINT	Controlador→ Computador	Envio de informação referente às juntas do manipulador robótico para o computador.
RECROBJOINT	Controlador→ Computador	Envio de informação referente a eixos externos do manipulador robótico para o computador.

RECPOSE	Controlador→ Computador	Envio de posição X,Y,Z e orientação qw,qx,qy,qz para o computador.
RECWOBJDATA	Controlador→ Computador	Envio da definição de objeto de trabalho para o computador.
OFFSET	Controlador→ Computador	Envio de posição X,Y,Z desde o ponto onde se encontra a ferramenta do robô para o controlador robótico.
MOVEL	Controlador→ Computador	Envio de ordem ao controlador robótico, para movimento linear até à última posição enviada para o controlador robótico ou sensor de contacto detetar colisão.
MOVEABSJ	Controlador→ Computador	Envio de ordem ao controlador robótico para movimento para último conjunto de posições de juntas enviado para o controlador robótico ou sensor de contacto detetar colisão.
ENVDIGITALOUT	Computador→ Controlador	Envio de sinal referente à ativação da funcionalidade de captura de peças por parte da ferramenta para o controlador robótico.
RECDIGITALOUT	Controlador→ Computador	Envio de sinal referente à ativação da funcionalidade de captura de peças por parte da ferramenta para o computador.
RECDIGITALIN	Controlador→ Computador	Envio de sinal correspondente ao estado do sensor de contacto para o computador.

ENVVELOCIDADE	Computador → Controlador	Envio de velocidade desejada para o controlador robótico.
PING	Computador → Controlador	Teste de conexão entre o computador e o controlador robótico.

Para a adição de um novo tipo de dados ou uma nova instrução é necessário adicionar uma palavra chave (que não exista previamente), definindo-se o sentido de informação, o tipo de dados e interpretação associada (pode ser preciso definir um novo tipo de dados). Não é necessário efetuar qualquer alteração ao módulo de envio e recepção de dados, encontrando-se a funcionalidade adicionada, à semelhança das restantes, independente da tecnologia de comunicação implementada.

Para o computador foi criada uma aplicação de comando do manipulador robótico, utilizando o sistema de comunicação implementado. Esta aplicação utiliza ROS, pelo que é executável em sistemas operativos do tipo Linux, nomeadamente distribuições Ubuntu. Na Tabela C.3 encontram-se definidas as funcionalidades e os parâmetros associados. Todas contém um campo cujo valor se encontra associado à ocorrência de erros durante a execução. Este campo pode ser utilizado por um programa que execute um comando na aplicação de comunicação para invalidar os eventuais dados recolhidos ou repetir um comando.

Tabela C.3: Serviços ROS.

Serviços Ros	Descrição	Campos Entrada	Campos Saída
serv_ler_confdata	Leitura de informação sobre configuração do manipulador robótico.	tempo	Sucesso cf1 cf2 cf4 cfx
serv_ler_extax	Leitura de informação sobre eixos externos ao manipulador robótico.	tempo	Sucesso eixo_exterior_1 eixo_exterior_2 eixo_exterior_3 eixo_exterior_4 eixo_exterior_5 eixo_exterior_6

serv_ler_- juntas	Leitura de informação referente às juntas do manipulador robótico.	tempo	sucesso junta_1 junta_2 junta_3 junta_4 junta_5 junta_6
serv_ler_- pose	Leitura de informação referente à pose associada à ferramenta.	tempo	sucesso x y z qw qx qy qz
serv_ler_rot	Leitura de informação sobre a orientação associada à ferramenta.	tempo	sucesso qw qx qy qz
serv_ler_- trans	Leitura de informação referente à posição da ferramenta no espaço.	tempo	sucesso x y z

serv_ler_wobj	Leitura de informação referente ao objeto de trabalho.	tempo	sucesso robot_hold user_frame_programmed user_frame_mecanical_unit_x user_frame_mecanical_unit_y user_frame_mecanical_unit_z user_frame_mecanical_unit_qw user_frame_mecanical_unit_qx user_frame_mecanical_unit_qy user_frame_mecanical_unit_qz user_frame_x user_frame_y user_frame_z user_frame_qw user_frame_qx user_frame_qy user_frame_qz
serv_ler_ventosa	Leitura de informação referente ao estado da ventosa.	tempo	sucesso estado_ventosa
serv_ler_fimcurso	Leitura de informação referente ao sensor de contacto.	tempo	sucesso estado_fimcurso
serv_esc_confdata	Envio de informação sobre configuração do manipulador robótico.	tempo cf1 cf2 cf4 cfx	sucesso

serv_esc_extax	Envio de informação sobre eixos externos ao manipulador robótico.	tempo eixo_exterior_1 eixo_exterior_2 eixo_exterior_3 eixo_exterior_4 eixo_exterior_5 eixo_exterior_6	sucesso
serv_esc_wobj	Envio de informação referente ao objeto de trabalho.	tempo robot_hold user_frame_programmed user_frame_mechanical_unit_x user_frame_mechanical_unit_y user_frame_mechanical_unit_z user_frame_mechanical_unit_qw user_frame_mechanical_unit_qx user_frame_mechanical_unit_qy user_frame_mechanical_unit_qz user_frame_x user_frame_y user_frame_z user_frame_qw user_frame_qx user_frame_qy user_frame_qz	sucesso
serv_esc_ventosa	Envio de comando para estado da ventosa.	tempo estado_ventosa	sucesso

serv_esc_velocidade	Envio de velocidade desejada.	tempo velocidade	sucesso
serv_mover_juntas	Envio de ordem de movimento de juntas.	tempo junta_1 junta_2 junta_3 junta_4 junta_5 junta_6	Sucesso estado_fimcurso
serv_mover_offset_trans	Envio de ordem de movimento incremental linear.	tempo offs_x offs_y offs_z	sucesso estado_fimcurso
serv_mover_pose	Envio de ordem de movimento para pose desejada.	tempo x y z qw qx qy qz	sucesso estado_fimcurso
serv_mover_rot	Envio de ordem de movimento para orinetação de ferramenta desejada.	tempo qw qx qy qz	sucesso estado_fimcurso
serv_mover_trans	Envio de ordem de movimento para posição desejada.	tempo x y z	sucesso estado_fimcurso

Robustez

O sistema é robusto face a erros no envio de dados. Estes erros são detetados na incapacidade de conversão entre texto e dados, e transmitidos para a fonte que os enviou, constituindo uma validação da informação enviada. A informação só é considerada

caso ambas as validações associadas à palavra chave e de informação sejam positivas. O sistema também é robusto perante interrupções de comunicação e erros no trânsito de informação tais como: perda de envio ou falha na receção de dados; interpretação de um envio de dados como múltiplos envios de dados (exemplo: múltiplos caracteres terminadores numa sequência de texto que só deveria possuir um no final), podendo recuperar com sucesso destas perturbações. A uma interrupção de comunicação pode associar-se a ausência de envio de informação, um exemplo é a tentativa de execução de um movimento perto de regiões singulares². Esta ação constitui um erro que não se encontra no conjunto de erros recuperáveis e impõe uma interrupção na execução do programa. Neste caso, deve-se reiniciar o ponteiro de execução do programa³, implicando que a informação a enviar do controlador para o computador no final do movimento se perde. Reiniciar a comunicação em ambos os lados pode tornar-se problemático no caso de as funcionalidades de comunicação não se encontrem separadas da execução do programa, podendo implicar a perda da informação de parte de um processo. De modo a evitar estes constrangimentos utiliza-se um tempo limite para esperar a chegada de dados em cada um dos lados, voltando a tentar um número de vezes pré-determinado e disponibilizando informação sobre a existência de erro na comunicação. Isto, associado à repetição de tentativas de comunicação por parte da aplicação (devido ao reconhecimento de existência de erros) permite voltar a estabelecer a comunicação nestes casos críticos, reiniciando apenas a comunicação no dispositivo onde ocorreu um erro fatal. Um fator determinante para a recuperação de erros é não procurar receber informação sem a validação de uma palavra chave. Com isto o elemento (computador ou controlador) que ficar bloqueado à espera de informação desiste e passa a procurar uma palavra chave. Com o decorrer de duas execuções dos processos enunciados nos fluxogramas presentes nas Figuras C.1 e C.2, o computador e o controlador encontram-se, em simultâneo, a tentar validar uma palavra chave. Para acelerar este processo, o número de repetições no controlador é diferente do número de repetições no computador. Deste modo durante uma execução dos fluxogramas C.1 e C.2, ambos se encontram na fase de informação ou na fase de palavra chave.

O sistema implementado não possui encriptação, sendo vulnerável a ataques do tipo repetição e a interceção com vista a leitura de dados. A informação transmitida corresponde essencialmente a comandos para a movimentação do manipulador robótico, não constituindo informação sensível. Devido à aplicação não necessitar de comunicação com dispositivos exteriores⁴ para processamento, o sistema a implementar pode ser isolado. Devido à comunicação entre computador e controlador ser ponto a ponto,

²tradução livre do termo em inglês *singularity*

³tradução livre do termo em inglês *program pointer*

⁴Serviços de computação "na nuvem" como o projeto *RoboEarth*

apenas é intercetável com acesso direto ao controlador, ao computador ou ao cabo, pelo que para a execução de um ataque é necessário o acesso ao local onde o sistema se encontra instalado. Por este motivo não se afigura necessário o uso de encriptação. Para introduzir encriptação basta encriptar os dados na forma de texto, o que pode ser implementado através de um novo módulo, independente dos restantes já implementados. Caso se opte por encriptar dados, aconselha-se que a introdução caracteres aleatórios no final da mensagem de modo a que a análise da quantidade de caracteres não permita determinar o tipo de mensagem, facilitando a descodificação. Aconselha-se igualmente a que a escolha do método de encriptação tenha em conta fatores que podem acelerar o processo de descriptação tais como:(1) valores repetidos como zeros e separadores decimais, (2) a presença de caracteres abertura e fecho de parêntesis retos em posições conhecidas (início e fim da mensagem) e (3) a presença de caracteres numa ordem conhecida (exemplo: virgulas e parêntesis retos) .

Esta página foi intencionalmente deixada em branco.

Apêndice D

Estimação de Pose

Nesta secção encontra-se exposta a argumentação que suporta a Equação 3.5 da Secção 3.5.

De modo a uma região conter outra (Equação 3.3), a sua área deve ser igual ou superior à da outra região visto conter pelo menos todos os elementos de área da outra. Por este motivo, entende-se que $Região(s_1) \subseteq Região(s_2) \implies Área(s_1) \geq Área(s_2)$. Uma ilustração deste princípio encontra-se na Figura D.1, na qual os elementos de uma região se encontram marcados a branco ou cinzento. Nesta Figura, a região da direita contém a da esquerda.

Sendo d_{aresta} um operador que visa identificar o valor máximo, de entre todos os elementos de uma região, do valor mínimo para a distância entre cada um dos pontos pertencentes à região e cada um dos pontos não pertencentes à região, a adição de um novo elemento a uma região constitui, para cada elemento presente anteriormente, a eliminação de um potencial mínimo de distância a aresta. Por este motivo, para todos os pontos anteriores o novo valor de distância a aresta é maior ou igual ao seu valor antigo. Isto implica que o máximo da distância dos pontos antigos se mantém inalterada ou aumenta com a adição de um novo ponto, motivo pelo qual mesmo que o valor de distância associado ao novo ponto não seja superior ao valor máximo de distância a aresta mais próxima anterior à sua adição, o novo valor de d_{aresta} é necessariamente maior ou igual ao anterior. Conclui-se assim $Região(s_1) \subseteq Região(s_2) \implies Máxima\ distância\ a\ aresta\ mais\ próxima(s_1) \geq Máxima\ distância\ a\ aresta\ mais\ próxima(s_2)$. Isto encontra-se exemplificado nas Figuras D.1 onde se encontram destacada a cinzento a atualização de menor distância a aresta para cada um dos elementos afetados.

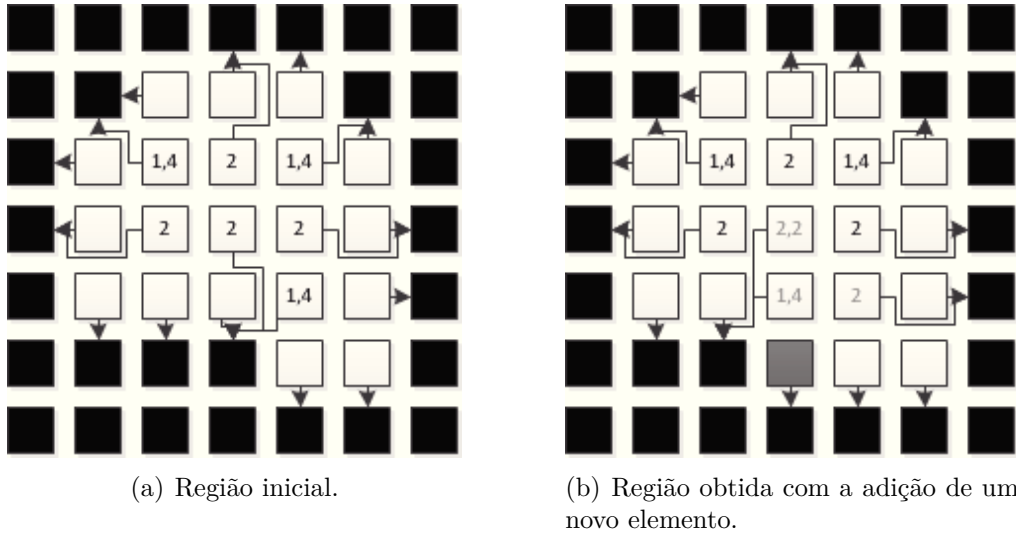


Figura D.1: Exemplo de atualização de máxima distância a aresta mais próximo numa região.

Considere duas regiões associadas a dois segmentos s_1 e s_2 , tais que $Região(s_1) \subseteq Região(s_2)$. A área do retângulo de menor área que contém o segmento s_1 é necessariamente maior ou igual à área associada ao retângulo de menor área que contém s_2 . Caso contrário existe um retângulo que contém todos os pontos do segmento s_1 , com área menor que o considerado. Assim obtêm-se que $Região(s_1) \subseteq Região(s_2) \implies l_{menor\ Área}(s_1) * c_{menor\ área}(s_1) \geq l_{menor\ Área}(s_2) * c_{menor\ área}(s_2)$.

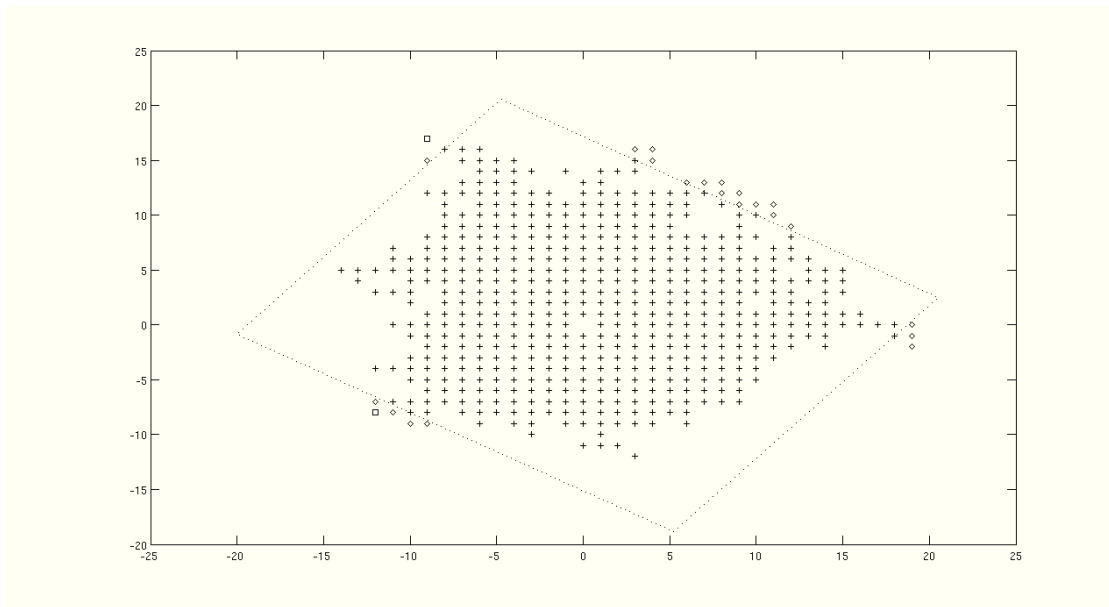
Com a adição de um novo ponto na vizinhança dos restantes, entenda-se dois pontos são vizinhos se a sua distância for inferior ao módulo de um pequeno incremento segundo os versores \hat{x} e \hat{y} ($|\delta(\vec{x}) + \delta(\vec{y})|$), é possível considerar um retângulo com dimensões $l_{menor\ área} + ((\delta(\vec{x}) + \delta(\vec{y})) \cdot \hat{l})$ e $c_{menor\ Área} + ((\delta(\vec{x}) + \delta(\vec{y})) \cdot \hat{c})$ que contenha a região obtida através da adição de um novo ponto, pois basta alargar cada um dos lados o valor associado à distância entre pontos vizinhos para acomodar o novo ponto, obtendo-se assim um limite superior para o aumento de área devido à adição de um ponto, com o valor $(l_{menor\ área} + ((\delta(\vec{x}) + \delta(\vec{y})) \cdot \hat{l})) * (c_{menor\ área} + ((\delta(\vec{x}) + \delta(\vec{y})) \cdot \hat{c})) - l_{menor\ área} * c_{menor\ área}$, ou seja, $\|\delta(\vec{x})\| \|\delta(\vec{y})\| + ((\delta(\vec{x}) + \delta(\vec{y})) \cdot \hat{l}) * c_{menor\ área} + ((\delta(\vec{x}) + \delta(\vec{y})) \cdot \hat{c}) * l_{menor\ área}$.

Através de simulação, gerando regiões aleatórias às quais se adiciona um ponto pode-se provar que existe a possibilidade de a adição de um ponto a uma região diminuir o valor associado a uma das medidas do lado do retângulo de menor área que engloba a região em crescimento, isto porque a adição de um novo ponto pode rodar o retângulo de menor área, o que ocorre no exemplo da Figura D.2.

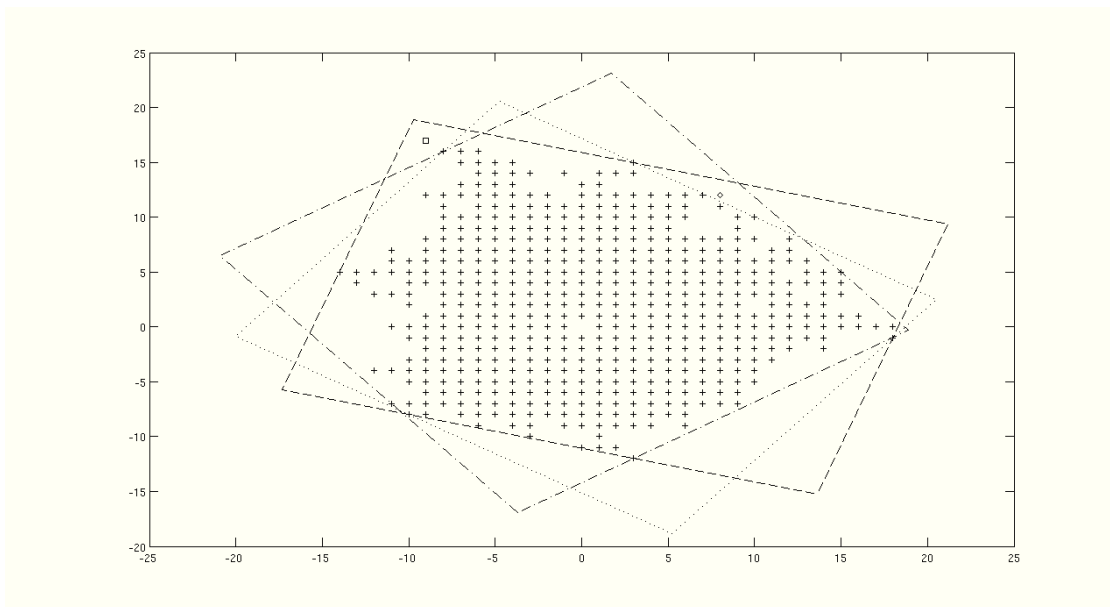
Pode provar-se que apenas um dos lados pode diminuir porque considerando $l_{\text{menor área}}(s_1) = a * l_{\text{menor área}}(s_2)$, e $c_{\text{menor área}}(s_1) = b * l_{\text{menor área}}(s_2)$, obtêm-se $l_{\text{menor área}}(s_1) * c_{\text{menor área}}(s_1) \geq l_{\text{menor área}}(s_2) * c_{\text{menor área}}(s_2) \implies a * b \geq 1 \implies a \geq 1 \vee b \geq 1$. Conclui-se assim que $Região(s_1) \subseteq Região(s_2) \implies l(s_1) \geq l(s_2) \vee c(s_1) \geq c(s_2)$.

Existe um limite mínimo para os valores do lado do retângulo de menor área, para uma região arbitrária igual a duas vezes a máxima distância à aresta mais próxima de uma região visto que a região associada deve conter um círculo inscrito de diâmetro igual ao valor desta métrica e um retângulo que contenha um círculo inscrito deve possuir lados superiores ao diâmetro desse círculo. Assim obtêm-se que $Região(s_1) \subseteq Região(s_2) \implies \exists \{ \text{limite mínimo}(l_{\text{menor área}}(s_1)), \text{limite mínimo}(l_{\text{menor área}}(s_2)) \} : \text{limite mínimo}(l_{\text{menor área}}(s_1)) \geq \text{limite mínimo}(l_{\text{menor área}}(s_2))$

É possível encontrar um limite máximo para ambos os lados, considerando que o retângulo de menor área que engloba um segmento não pode possuir lados com dimensões superiores ao diâmetro do círculo de menor área que engloba toda a região. Considere duas regiões associadas a dois segmentos s_1 e s_2 , tais que $Região(s_1) \subseteq Região(s_2)$, a área associada ao círculo de menor área que contém o segmento s_1 é necessariamente maior ou igual à área associada ao círculo de menor área que contém s_2 . Caso contrário existe um círculo que contém todos os pontos do segmento, com área menor que o considerado. O retângulo que compreende o círculo de menor área que engloba a região compreende necessariamente todos os pontos associados à região. Considerando um retângulo que contém o círculo considerado e que um dos seus lados é superior ao diâmetro do círculo, é possível encontrar um segundo retângulo que contenha o círculo com a dimensão desse lado igual ao diâmetro do círculo, mantendo as dimensões do outro lado, o que implica que este novo retângulo possui menor área que o anterior. Neste retângulo os novos vértices são obtidos através da interseção entre as duas retas do retângulo paralelas ao lado considerado e os pontos pertencentes a duas retas tangentes à circunferência e perpendiculares ao lado considerado. Assim obtêm-se que $Região(s_1) \subseteq Região(s_2) \implies \exists \{ \text{limite mínimo}(c_{\text{menor área}}(s_1)), \text{limite mínimo}(c_{\text{menor área}}(s_2)) \} : \text{limite mínimo}(c_{\text{menor área}}(s_1)) \geq \text{limite mínimo}(c_{\text{menor área}}(s_2))$. Este processo encontra-se esquematizado na Figura D.3.

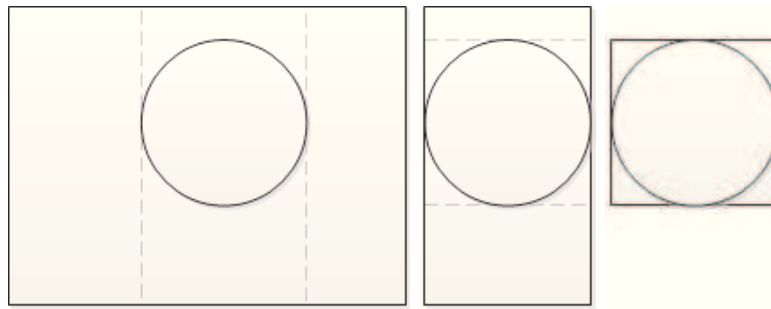


(a) Retângulo de menor área para associado a uma região, com marcadores associados à adição de pontos, que diminuem a dimensão do lado menor (idêntificado com \diamond) e pontos que diminuem a dimensão do lado maior (idêntificado com \square) do novo retângulo de menor área.



(b) Retângulos de menor área obtidos com a adição dos pontos (idêntificados com \diamond e \square) a uma região.

Figura D.2: Exemplo de diminuição de um dos lados do retângulo de menor área que engloba uma região com a adição de um ponto a uma região.



(a) Retângulo que contém a região definida por uma circunferência. (b) Iteração do processo de obtenção de limites máximos para um dos lados. (c) Iteração do processo de obtenção de limites máximos.

Figura D.3: Exemplo de obtenção de um retângulo com dimensões iguais aos limites máximos para um retângulo de menor área que contém uma região.