



**Rodrigo António
Nunes Salgueiro**

**Correspondência entre Mapeamentos 3D e
Modelos CAD para Operações de Bin-Picking**



**Rodrigo António
Nunes Salgueiro**

**Correspondência entre Mapeamentos 3D e
Modelos CAD para Operações de Bin-Picking**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Mecânica, realizada sob a orientação científica do Doutor Vitor Manuel Ferreira dos Santos, Professor Associado do Departamento de Engenharia Mecânica da Universidade de Aveiro.

o júri / the jury

presidente / president

Professor Doutor José Paulo Oliveira Santos

Professor Auxiliar da Universidade de Aveiro

vogais / examiners committee

Doutor Luís André Freitas da Rocha

Investigador do Instituto de Engenharia de Sistemas e Computadores do Porto
(arguente)

Professor Doutor Vítor Manuel Ferreira dos Santos

Professor Associado da Universidade de Aveiro (orientador)

**agradecimentos /
acknowledgements**

O trabalho desenvolvido nesta dissertação não teria sido alcançado sem a ajuda indispensável de algumas pessoas. Como tal, as seguintes palavras vêm expressar o que nunca lhes conseguirei agradecer. Aos Professores Vitor Santos e Miguel Oliveira, todos os ensinamentos passados e ajuda prestada, sem esquecer os incentivos nos momentos de maior desalento. Aos meus Amigos e companheiros de estudo, que me ensinaram a camaradagem e amizade, tanto nos momentos de trabalho como fora deles, ao longo de todo o percurso académico. Aos meus Pais e Irmã, pelo apoio que me deram sempre, pelas oportunidades que me possibilitaram, por aquilo que sou hoje... A Família que pela sua proximidade sempre me ajudou. A Andreia que sempre esteve a meu lado, tornou os desalentos em ânimos e sempre me fez acreditar que ia conseguir.

Palavras-Chave

Bin-Picking; CAD; Percepção 3D; Kinect; PCL; ROS;

Resumo

Este trabalho propõe uma solução para Bin-Picking, baseada em mapeamentos 3D e referências do modelo CAD 3D do objeto. Em primeiro lugar foi convertido o modelo STL, da peça, em nuvem de pontos por diferentes métodos. De seguida foi detetado o contentor onde os objetos se encontram no espaço de trabalho, isolando o espaço de busca e definindo as restrições á abordagem do manipulador. Deste modo, as duas nuvens foram conseguidas possibilitando a comparação do modelo sintetizado com a nuvem recolhida pelo sensor. A comparação foi executada a custa de descritores locais, calculados em ambas as nuvens, tais como o SHOT e FPFH. Algumas propriedades do modelo foram analisadas , como a parcialidade e inclusão de ruído.

KeyWords

Bin-Picking; CAD; 3D Perception; Kinect; PCL; ROS;

Abstract

This thesis proposes a solution for bin-picking based on 3D mapping and 3D CAD model reference, of the object. First it was converted STL model of the part, in point cloud by different methods. Then the container was detected where objects are in the workspace, isolating the search space and defining the approach restrictions of the robot. Thus the two clouds were obtained allowing comparison between the synthesized model and the cloud gathered by the sensor. The comparison was performed at the expense of local descriptors calculated in both clouds, such as SHOT and FPFH. Some properties of the model were analyzed, such as partiality and inclusion of noise.

Conteúdo

Conteúdo	i
Lista de Figuras	iii
Lista de Tabelas	vii
Lista de Acrónimos e siglas	ix
1 Introdução	1
1.1 Descrição da Problemática	2
1.2 Objetivos	3
1.3 Estrutura da Dissertação	4
2 Estado de Arte	5
2.1 Trabalhos Académicos	5
2.1.1 Object Recognition: Bin-Picking For Industrial Use	5
2.1.2 Position and Pose Recognition of Randomly Stacked Objects Using Highly Observable 3D Vector Pairs	6
2.2 Implementações Industriais	9
2.2.1 InPicker	9
2.2.2 HALCON	9
2.2.3 iRVision 3D	10
3 Infraestrutura Experimental	13
3.1 Hardware e Equipamentos	13
3.1.1 Sensor Kinect	13
3.1.2 Objeto e Acessórios de Apoio	16
3.2 Software e Bibliotecas	21
3.2.1 MeshLab	21
3.2.2 Blender	22
3.2.3 Robot Operating System (ROS)	22
3.2.4 Point Cloud Library (PCL)	23
3.3 Arquitetura Geral da Implementação	26
4 Conversão CAD nuvem de pontos	31
4.1 Primeira Abordagem	31
4.2 Metodologia com recurso ao Blender	33

4.3	Modelos parciais	37
4.4	Modelos com ruído	39
5	Deteção do contendor com objetos	45
5.1	Filtragem Inicial	45
5.2	Transformação do Bordo	46
5.3	Isolamento e Extração	49
6	Determinação da Correspondência	53
6.1	Metodologia Base	54
6.2	Modelo Parcial	59
6.3	Cena, Modelo e Ruído	64
7	Testes e Resultados	71
7.1	Descritor SHOT	71
7.2	Descritor FPFH	75
8	Conclusões e Trabalho Futuro	81
	Bibliografia	83
	Apêndices	87
A	Testes de correspondência SHOT	88
B	Testes de correspondência FPFH	91

Lista de Figuras

1.1	Gráfico de número de robôs industriais vendidos por ano, entre 2000 e 2015 [1].	1
1.2	Figura ilustrativa de uma infraestrutura de Bin-Picking [2].	2
2.1	Visualização dos módulos e vias de comunicação[8].	6
2.2	Esquema do reconhecimento de objetos[8].	6
2.3	Área relevantes para o teste de colisão usadas para a determinação do ponto de "graspping"[8].	7
2.4	Estrutura do par de vetores. O círculo azul representa um ponto, o par de setas azuis o par de vetores e as setas a rosa os vetores normais[9].	7
2.5	Exemplo de resultados de reconhecimento para quatro nuvens utilizadas[9]. . .	8
2.6	Figuras representativas da solução InPicker proposta para bin-picking[10]. . . .	9
2.7	Imagem de detecção de objetos realizada em HALCON [13].	10
2.8	Nuvem de pontos de detecção de tubos realizada em HALCON [14].	11
2.9	Figura ilustrativa da solução fornecida pela Fanuc, com o manipulador e sensor da marca [15].	11
3.1	Imagem ilustrativa do sensor kinect [18].	13
3.2	Imagem de infravermelhos recolhida pelo sensor da kinect [19].	14
3.3	Figura esquemática da arquitetura SDK (Software Development Kit)[20].	15
3.4	Janela utilizada pelo modulo de calibração.	16
3.5	Fotografia da peça de Lego utilizada.	16
3.6	Figura da nuvem de pontos da peça Lego, perspetiva de topo.	17
3.7	Figura da nuvem de pontos da peça Lego, perspetiva lateral.	17
3.8	Fotografia do paralelepípedo utilizado.	17
3.9	Figura da nuvem de ponto do paralelepípedo, perspetiva de topo.	18
3.10	Figura da nuvem de ponto do paralelepípedo, perspetiva lateral.	18
3.11	Fotografia da peça utilizada.	18
3.12	Figuras de nuvens de pontos do paralelepípedo, perspetiva de topo.	19
3.13	Figuras de nuvens de pontos do paralelepípedo, perspetiva lateral.	19
3.14	Fotografia da infraestrutura inicial [25].	20
3.15	Figuras da montagem do sensor com suporte em ambiente Solidworks.	20
3.16	Infraestrutura final.	21
3.17	Caixa utilizada na atividade experimental.	21
3.18	Logótipo utilizado pela PCL[28].	23
3.19	Estrutura de assinatura do descritos SHOT [29].	25
3.20	Estrutura de relação de pontos do descritor PFH[30].	25

3.21	Estrutura de relação de pontos do descritor FPFH[30].	26
3.22	Diagrama geral da solução proposta para o problema de bin-picking.	27
3.23	Diagrama de nós e tópicos <i>Robot Operating System</i> (ROS) do sistema implementado.	28
4.1	Cubo convertido no formato " <i>Polygon File Format</i> (PLY)" em ambiente MeshLab.	32
4.2	Fluxograma da primeira abordagem a conversão do modelo CAD. Nela estão representados os formatos e ferramentas utilizadas.	32
4.3	Nuvem de pontos convertida pela função "ply2pcd".	33
4.4	Fluxograma da segunda abordagem a conversão do modelo CAD. Nela estão representados os formatos e ferramentas utilizadas.	33
4.5	Nuvens de pontos do cubo obtida pela função "mesh2pcd".	34
4.6	Fluxograma da abordagem a conversão do modelo CAD. Esta foi possibilitada pela utilização do comando "Remesh" do Blender.	34
4.7	Malhas " <i>STereoLithography</i> (STL)" original e processada em ambiente Blender.	35
4.8	Nuvem de pontos convertida pela função "ply2pcd" usando uma malha modificada.	36
4.9	Provete de teste em ambiente Solidworks.	36
4.10	Malha "STL" do provete de teste em ambiente Blender.	37
4.11	Malha "PLY" do provete de teste obtida após modificação em ambiente Blender.	37
4.12	Nuvem de pontos do provete obtida pela função "ply2pcd", usando uma malha reconstruída em ambiente Blender.	38
4.13	Janela de terminal para a execução do comando <code>virtual_scanner</code> , sem configurações.	38
4.14	Janela de terminal com possibilidades de configuração do comando <code>virtual_scanner</code>	39
4.15	Nuvem de pontos parcial obtida pelo <code>virtual_scanner</code>	40
4.16	Figuras esquemáticas das dimensões do recetor infravermelhos.	41
4.17	Nuvem de pontos obtida pela digitalização do tampo da bancada.	42
4.18	Vista lateral da nuvem de pontos obtida pela digitalização do tampo da bancada.	42
4.19	Imagem do modelo CAD original, utilizado na conversão em nuvem de pontos.	43
4.20	Nuvens de pontos parciais resultantes do <code>virtual_scanner</code> modificado.	43
4.21	Nuvens de pontos parciais resultantes do <code>virtual_scanner</code> modificado, com inclusão de ruído.	44
5.1	Sequência de detecção e isolamento do contentor e objetos.	46
5.2	Nuvem de pontos recolhida pelo sensor na estrutura experimental sem qualquer processamento.	47
5.3	Nuvem de pontos do espaço de trabalho obtida após filtragem.	48
5.4	Nuvens de pontos do espaço de trabalho com identificação e remoção do tampo da bancada, e fundo do contentor.	48
5.5	Nuvem de pontos do contentor após isolamento do bordo.	49
5.6	Nuvem de pontos do contentor após projeção dos pontos do bordo.	49
5.7	Nuvem de pontos do contentor com redução de escala dos pontos anteriormente projetados.	50
5.8	Nuvem de pontos do contentor com pontos do <code>convex hull</code> destacados.	50
5.9	Nuvem de pontos do contentor com pontos correspondentes as peças destacados.	51
5.10	Nuvem de pontos das peças.	51

6.1	Imagem do conjunto peças e nuvem de pontos recolhida das mesmas.	53
6.2	Fluxograma do conjunto de ações na sequência de correspondência.	54
6.3	Exemplo de normais calculadas numa nuvem de pontos recolhida.	54
6.4	Exemplo de amostragem calculada numa nuvem de pontos recolhida, nesta é possível observar o resultado obtido pela redução da densidade de pontos.	55
6.5	Nuvem de pontos da cena recolhida com o sensor, com um conjunto de objetos incluindo o que se pretende detetar(isolar).	56
6.6	Nuvem de pontos do modelo utilizado no primeiro ensaio. É o modelo completo obtido pela função "ply2pcd".	56
6.7	Nuvem de pontos do primeiro ensaio de correspondência.	57
6.8	Nuvem de pontos da cena usada na análise da metodologia de correspondência.	57
6.9	Nuvem de pontos do modelo usado na análise da metodologia de correspondência.	58
6.10	Nuvem de pontos resultante da correspondência.	58
6.11	Nuvem de pontos resultante da correspondência usada para despistagem, utilizando a cena de maior complexidade.	58
6.12	Nuvens de pontos das amostragens(pontos a azul) por diferentes algoritmos.	59
6.13	Nuvem de pontos resultante da correspondência usando amostragem por Intrinsic Shape Signatures.	60
6.14	Nuvem de pontos do modelo parcial gerado por remoção de planos.	60
6.15	Ensaio de correspondência com modelo parcial.	61
6.16	Vista aproximada ensaio de correspondência com modelo parcial.	61
6.17	Ensaio de correspondência com modelo parcial e descritor FPFH.	62
6.18	Ensaio de correspondência com múltiplos modelos das várias vistas parciais do objeto de referência.	63
6.19	Ensaio de correspondência para análise do problema na utilização das várias vistas.	63
6.20	Nuvem de pontos modelo.	64
6.21	Nuvens de pontos recolhidas a diferentes distâncias para comparação.	65
6.22	Esquema de ilustração da secção representada pelas nuvens de pontos recolhidas.	65
6.23	Nuvens de pontos recolhidas a diferentes distâncias para comparação.	66
6.24	Sequencia adaptada de operações para correspondência.	66
6.25	Exemplo do ensaio dos descritores.	67
6.26	Modelo resultante da aplicação de ruído com variação de 2 milímetros.	68
6.27	Correspondência utilizando modelo com ruído de 2 milímetros e descritor FPFH.	68
6.28	Correspondência utilizando modelo com ruído de 2 milímetros e descritor SHOT.	69
7.1	Ensaio de correspondência com o descritor SHOT e modelo sem ruído.	72
7.2	Ensaio de correspondência com o descritor SHOT e modelo com 1 milímetro de ruído.	72
7.3	Ensaio de correspondência com o descritor SHOT e modelo com 2 milímetros de ruído.	72
7.4	Gráfico da percentagem de correspondência em função do ruído imposto ao modelo, para o descritor SHOT.	73
7.5	Ensaio de correspondência com o descritor SHOT e modelo com 3 milímetros de ruído.	73
7.6	Resultados obtidos com vários modelos de ruído 2 milímetros, para o descritor SHOT.	74

7.7	Gráfico dos valores de percentagem de correspondência para o ensaio de sensibilidade ao ruído, com o descritor SHOT.	75
7.8	Ensaio de correspondência com o descritor FPFH e modelo sem ruído.	76
7.9	Ensaio de correspondência com o descritor FPFH e modelo com 1 milímetro de ruído.	76
7.10	Ensaio de correspondência com o descritor FPFH e modelo com 2 milímetros de ruído.	76
7.11	Gráfico da percentagem de correspondência em função do ruído imposto ao modelo.	77
7.12	Resultados obtidos com vários modelos de ruído 2 milímetros, para o descritor FPFH.	78
7.13	Gráfico dos valores de percentagem de correspondência para o ensaio de sensibilidade ao ruído, com o descritor FPFH.	79
A.1	Ensaio de correspondência com o descritor SHOT e modelo com 4 milímetros de ruído.	88
A.2	Ensaio de correspondência com o descritor SHOT e modelo com 5 milímetros de ruído.	88
A.3	Ensaio de correspondência com o descritor SHOT e modelo com 6 milímetros de ruído.	89
A.4	Ensaio de correspondência com o descritor SHOT e modelo com 7 milímetros de ruído.	89
A.5	Ensaio de correspondência com o descritor SHOT e modelo com 8 milímetros de ruído.	89
A.6	Ensaio de correspondência com o descritor SHOT e modelo com 9 milímetros de ruído.	90
A.7	Ensaio de correspondência com o descritor SHOT e modelo com 10 milímetros de ruído.	90
B.1	Ensaio de correspondência com o descritor FPFH e modelo com 3 milímetros de ruído.	91
B.2	Ensaio de correspondência com o descritor FPFH e modelo com 4 milímetros de ruído.	91
B.3	Ensaio de correspondência com o descritor FPFH e modelo com 5 milímetros de ruído.	92
B.4	Ensaio de correspondência com o descritor FPFH e modelo com 6 milímetros de ruído.	92
B.5	Ensaio de correspondência com o descritor FPFH e modelo com 7 milímetros de ruído.	92
B.6	Ensaio de correspondência com o descritor FPFH e modelo com 8 milímetros de ruído.	93
B.7	Ensaio de correspondência com o descritor FPFH e modelo com 9 milímetros de ruído.	93
B.8	Ensaio de correspondência com o descritor FPFH e modelo com 10 milímetros de ruído.	93

Lista de Tabelas

3.1 Tabela de características técnicas do sensor [19]	14
---	----

Lista de Acrónimos e siglas

CAD *Computer Aided Design*

CVFH *Clustered Viewpoint Feature Histogram*

FPFH *Fast Point Feature Histogram*

OBJ *Wavefront Object*

PCD *Point Cloud Data*

PCL *Point Cloud Library*

PFH *Point Feature Histogram*

PLY *Polygon File Format*

ROS *Robot Operating System*

SHOT *Signature Histograms of Orientation*

SPFH *Simplified Point Feature Histogram*

STL *STereoLithography*

Capítulo 1

Introdução

A automação é uma área de engenharia que tem evoluído bastante ao longo dos anos, impulsionando o desenvolvimento de novas tecnologias como a robótica e a visão artificial. A tendência deve-se a diversos fatores competitivos, tais como a redução de custos e o aumento da capacidade e versatilidade produtiva, justificando o investimento na tecnologia.

A robótica industrial é uma das vertentes com maior crescimento nos últimos anos, como podemos ver na figura 1.1. Segundo a International Federation of Robotics, as vendas têm batido recordes consecutivos tendo sido fornecidos duzentos e quarenta mil equipamentos em 2015 [1].

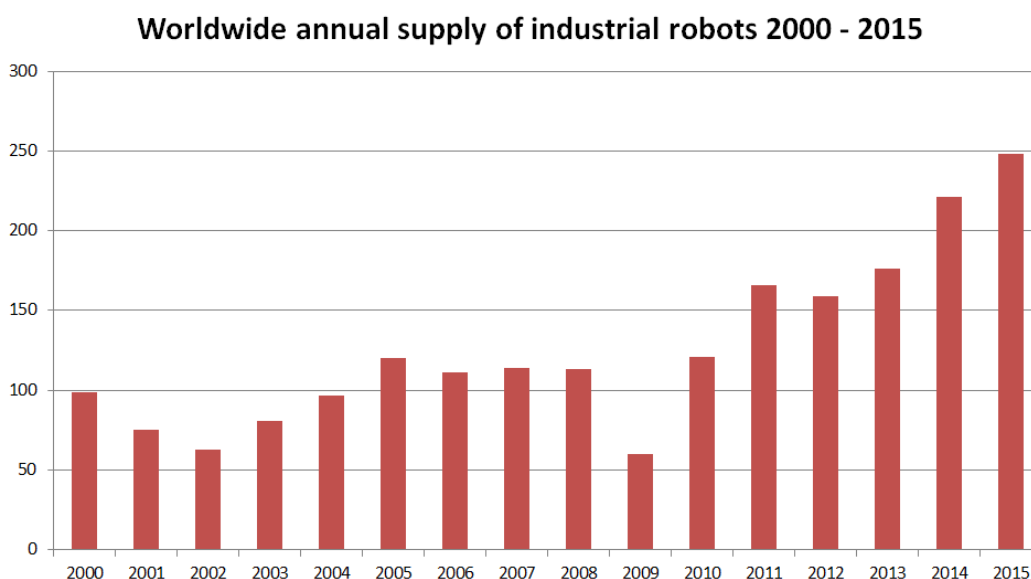


Figura 1.1: Gráfico de número de robôs industriais vendidos por ano, entre 2000 e 2015 [1].

As aplicações onde se integram robôs industriais são bastante diversificadas, não se restringindo apenas às operações mais comuns como a paletização e soldadura. A evolução dos sistemas de percepção foi crucial para potenciar o desenvolvimento de sistemas robóticos mais versáteis e aplicações de maior complexidade. A visão artificial é um dos sistemas de percepção mais utilizados. Contudo, com o desenvolvimento de sensores tri-dimensionais, este novo tipo de percepção tem vindo a ser utilizado para dar resposta a problemas para os quais os sistemas

menos avançados não apresentaram solução.

1.1 Descrição da Problemática

O problema na robótica de manipulação que permanece há mais tempo sem que se encontre uma solução universal é sem dúvida o Bin-Picking. A atividade corresponde à recolha de um objeto do interior de um contentor, dentro do qual se encontra colocado de forma aleatória em conjunto com outros objetos. Quando realizada por um ser humano, esta ação é trivial, mas a sua execução por uma máquina é de extrema dificuldade. Os passos necessários para levar a cabo esta operação dependem de múltiplas vertentes da robótica e perceção, são: a identificação e localização do objeto, a determinação do ponto de "grasping" e o cálculo de trajetória.

A figura 1.2, representa uma ilustração de uma instalação de bin-picking. Nela está presente o robô que executa a manipulação dos objetos, o sensor que recolhe toda a informação para processamento e identificação, e os objetos a recolher no interior do contentor.



Figura 1.2: Figura ilustrativa de uma infraestrutura de Bin-Picking [2].

O trabalho desenvolvido em [3] veio expor o estado da arte na área e descrever o percurso de cinco décadas de desenvolvimento. As primeiras abordagens para a solução da problemática, eram implementadas sem recurso à perceção. Estas consistiam na utilização de grippers de vácuo ou magnéticos movimentados pelo manipulador em trajetórias pré-definidas no interior do caixote, tentando deste modo executar a recolha. A metodologia não é eficaz e tão pouco segura para os equipamentos, pois não era assegurada a não colisão com os próprios objetos no interior do contentor.

O início da identificação de objetos foi desenvolvida em 1963 como descrito em [4]. O trabalho consistiu na utilização de imagens 2D monocromáticas, que eram filtradas de forma a detetar as arestas dos objetos que nelas figuravam, funcionando esta informação como uma marca do objeto. Para executar a identificação e localização, as imagens recolhidas eram comparadas com uma base dados de imagens recolhidas anteriormente em diferentes orientações.

O uso de informação tri-dimensional apenas surgiu mais tarde em trabalhos como [5]. A metodologia baseava-se na utilização de um projetor de um padrão de linhas que incidia sobre

a cena. A deformação imposta ao padrão permitia a determinação da distância de cada plano, formando uma imagem de profundidades.

O trabalho desenvolvido em [6] valeu-se dos dados da imagem de profundidade e imagem de arestas para o desenvolvimento da sua própria solução. Iniciando-se pelo fecho das linhas de contorno e de seguida pelo cálculo dos descritores. A inovação da abordagem é suportada pelo referido uso de descritores como comprimento, raio, ângulo de orientação, entre outros, que serviam de suporte a identificação com valores anteriormente obtidos.

Os avanços tecnológicos e o estudo de novas técnicas impulsionaram o desenvolvimento de novas abordagens. O estudo realizado em [7] é baseado no uso de fotometria estéreo. Esta tecnologia baseia-se na recolha de imagens com fontes de luz em diferentes orientações para a determinação do mapa de normais recolhendo dessa forma a informação da cena.

Os trabalhos mais recentes têm-se baseado, em maioria, na utilização de nuvens de pontos. A opção foi influenciada pela introdução de sensores, que executam o levantamento de nuvens de pontos, e apresentam um baixo custo, como é o exemplo da kinect. O processamento da nuvem recolhida está dependente das condições da mesma e do tipo de solução a desenvolver.

Segundo [3] as abordagens modernas ao problema de bin-picking têm seguido duas linhas de desenvolvimento, para a solução de identificação de objetos: baseada em votação, e baseada em correspondência. A metodologia de votação usa a transformada de Hough e calcula múltiplos alinhamentos entre o modelo e a cena de forma a alcançar a transformação que minimize a distância entre ambos.

A metodologia baseada em correspondência de descritores pode ser dividida em dois tipos: globais e locais. Os descritores globais consideram para cálculo todos os pontos, ao passo que os locais se limitam a uma vizinhança pré-definida. Em ambas as tipologias o alinhamento é executado pela correspondência entre descritores.

As soluções não comerciais que foram disponibilizadas até a data, são desenvolvidas para problemas específicos. As metodologias tornam as soluções muito pouco versáteis não acomodando, por exemplo a recolha de objetos diferentes. Deste modo torna-se muito difícil a viabilizar estas ferramentas.

1.2 Objetivos

Os objetivos deste trabalho estão enquadrados no desenvolvimento de soluções para a execução de bin-picking. A grande contribuição que se pretende acrescentar a uma solução universal é a utilização de modelos *Computer Aided Design* (CAD) 3D dos objetos como referência.

A utilização de modelos CAD permite impulsionar a versatilidade da instalação. Esta estratégia possibilita utilizar a instalação em processos muito dinâmicos. Um exemplo deste tipo de solicitação é a indústria automóvel, nesta área os produtos mudam muito frequentemente e o "time to market" é um fator crítico. A estratégia desenvolvida permite a configuração da solução para uma peça que ainda não foi produzida, esta adaptação para outro objeto resume-se a alteração de um ficheiro em minutos.

A implementação conjugada com um sistema de aquisição 3D, como o sensor kinect, possibilita uma abordagem de baixo custo e elevada versatilidade a operação de um sistema de bin-picking. De forma sucinta, os objetivos principais são os seguintes:

- Processamento de modelos CAD 3D de objetos para a sua integração num processo de bin-picking;

- Estudo e implementação de técnicas de correspondência entre nuvens de pontos para o reconhecimento ou identificação de objetos parcialmente visíveis;
- Detecção de condicionantes geométricas para permitir o "grasping" de um manipulador;

1.3 Estrutura da Dissertação

A estrutura da dissertação caracteriza-se do seguinte modo: em primeiro lugar o presente capítulo, **introdução**. Neste é apresentado o trabalho desenvolvido e a problemática que lhe serve de base.

O capítulo 2, **Estado de Arte**, tem por objetivo apresentar os trabalhos desenvolvidos quer em âmbito acadêmico como industrial. O capítulo 3, **Infraestrutura Experimental**, descreve o software e hardware utilizado, tal como alguns auxiliares desenvolvidos e a arquitetura do sistema implementado.

O capítulo 4, **Conversão CAD nuvem de pontos**, expõe a metodologia testada para a integração dos ficheiros CAD na identificação dos objetos. O capítulo 5, **Detecção do contentor com objetos**, aborda a metodologia desenvolvida para a deteção do contentor de retenção dos objetos, e isolamento das respetivas nuvens de pontos.

O capítulo 6, **Determinação da Correspondência**, contém o desenvolvimento da comparação entre modelo sintetizado e cena recolhida pelo sensor. Neste é implementado o algoritmo testado no capítulo seguinte, **Testes e Resultados**. O documento termina como a exposição de **Conclusões e Trabalho Futuro**, onde é realizada uma reflexão sobre o trabalho desenvolvido, e apresentadas algumas propostas de continuidade.

Capítulo 2

Estado de Arte

O capítulo “Estado de Arte” tem por objetivo a exposição de alguns trabalhos desenvolvidos recentemente na área e algumas soluções já comercializadas em aplicações industriais. Os trabalhos serão de investigação e natureza acadêmica, de forma a perceber o ponto de situação e qual o rumo que a comunidade científica está a impulsionar ao desenvolvimento da área. As implementações industriais têm outro propósito, nomeadamente o conhecimento de quais as soluções com fiabilidade suficiente para integrarem um processo industrial.

2.1 Trabalhos Académicos

2.1.1 Object Recognition: Bin-Picking For Industrial Use

O estudo realizado em [8] definiu uma estratégia para realização de bin-picking. Este descreve todas as etapas desde a definição do hardware até ao cálculo das trajetórias.

A estrutura utiliza três ferramentas de hardware para o funcionamento da solução, um braço robótico, um sensor e um computador. O manipulador é ajustado às condicionantes da instalação, tais como as dimensões a alcançar e as cargas a suportar. Associado ao manipulador está um “gripper” ou garra que possibilita ao robô manipular os objetos, este é selecionado em função dos objetos que são manipulados.

A arquitetura de software foi desenhada de forma a interligar todas as ações necessárias e a isolar cada uma delas em módulos. Os módulos construídos foram cinco, tal como representado na figura 2.1. O módulo “Sensor” referente ao sensor gere a recolha da nuvem de pontos e envia a informação. O módulo de “Recognition” identifica e localiza os objetos. A comunicação com o robô é gerida pelo módulo “Robot” e o de “Gripping” determina para os objetos detetados os possíveis pontos de “graspping”. Por fim o módulo “Control” coordena toda a sequência.

O estratégia de reconhecimento utilizada neste trabalho foi baseada numa busca por uma árvore heurística. Esta estrutura executa a busca de pontos chave na cena recolhida. A referência utilizada para a correspondência é feita pelos modelos CAD dos objetos. A determinação dos pontos chave no modelo e na cena recolhida, permite o cálculo das correspondências e por fim dos alinhamentos pelo método da transformada de Hough. A representação da sequência de correspondência está ilustrada na figura 2.2.

A etapa final da metodologia é a execução do “graspping”. A correta movimentação do robô sem riscos de colisão é assegurada por uma simulação dos movimentos a “posteriori”. A simulação desempenha-se pelo uso da nuvem de pontos recolhida pelo sensor, o modelo CAD do “gripper” e as coordenadas e orientações para a abordagem a cada objeto. O critério para

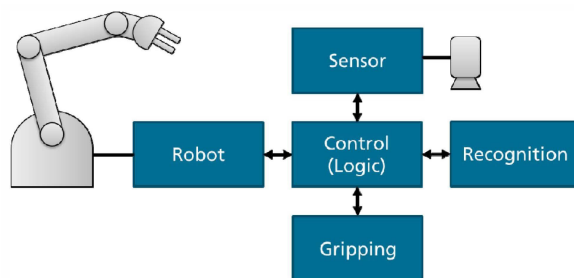


Figura 2.1: Visualização dos módulos e vias de comunicação[8].

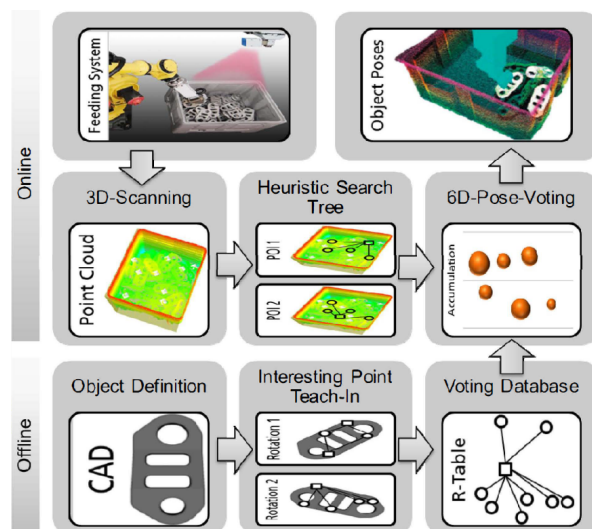


Figura 2.2: Esquema do reconhecimento de objetos[8].

validação do movimento do robô é a área a ser ultrapassada durante a atividade. A figura 2.3 esquematiza o processo de simulação, como podemos observar são definidas três áreas: livre, colisão e desconhecida. A simulação coloca o modelo do “gripper” na posição de “grasping”, se este se mantiver na área livre a movimentação é aprovada, caso ocorra interferência com a área de colisão ou desconhecida, a movimentação é abortada.

O trabalho expôs uma sequência de execução de bin-picking interessante e viável. Apesar da sua descrição ser pouco aprofundada permite reter uma estrutura de desenvolvimento para diferentes problemas com diversas conjunções de soluções para cada módulo.

2.1.2 Position and Pose Recognition of Randomly Stacked Objects Using Highly Observable 3D Vector Pairs

O trabalho desenvolvido em [9] consistiu numa metodologia para identificação e localização de objetos em cenas de elevada complexidade. Os objetos encontravam-se amontoados em caixotes, em conformidade com uma situação de bin-picking com elevados níveis de oclusão dos objetos.

A abordagem utilizada baseou-se no desenvolvimento de um tipo de descritor, os pares de vetores. Os pares de vetores são calculados recorrendo a três pontos, como ilustra a figura 2.4. O primeiro ponto é definido como origem de ambos os vetores, terminando estes em cada um

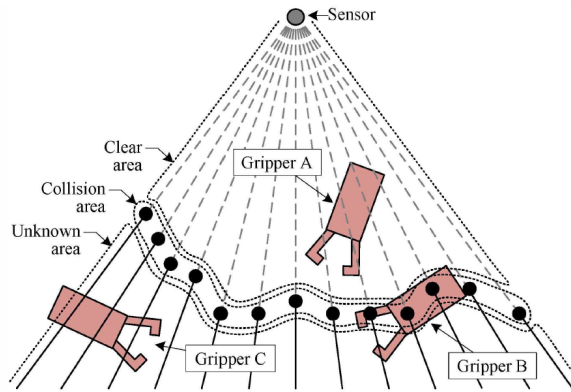


Figura 2.3: Área relevantes para o teste de colisão usadas para a determinação do ponto de "grasping"[8].

dos outros dois pontos, formando um ângulo entre si. Definida a estrutura do par de vetores o descritor é calculado pelo conjunto dos produtos entre as normais em cada ponto e a normal comum a ambos os vetores.

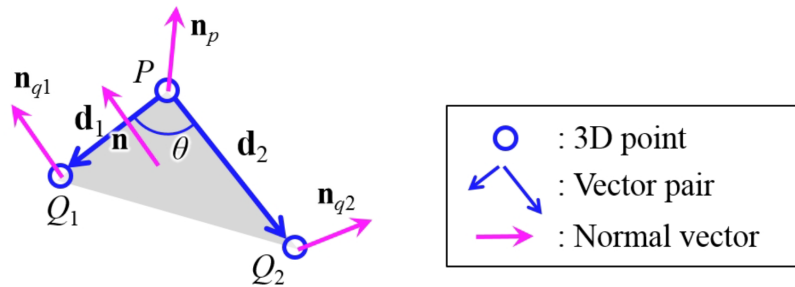


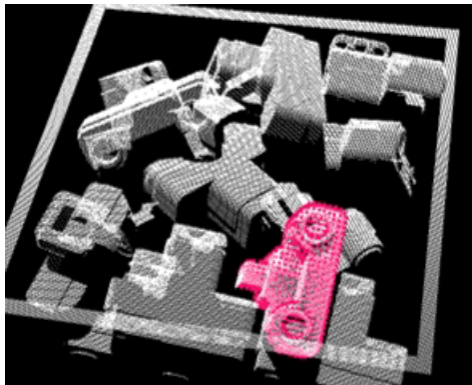
Figura 2.4: Estrutura do par de vetores. O círculo azul representa um ponto, o par de setas azuis o par de vetores e as setas a rosa os vetores normais[9].

Determinados os vetores, foram implementados dois tipos de amostragem, baseada na distinção e baseada na exposição. A distinção consiste nos descritores que apresentam menor ocorrência no objeto sendo mais distintos dos restantes. O critério de exposição consiste nos descritores que são mais facilmente observáveis em múltiplas perspectivas e a diferentes distâncias.

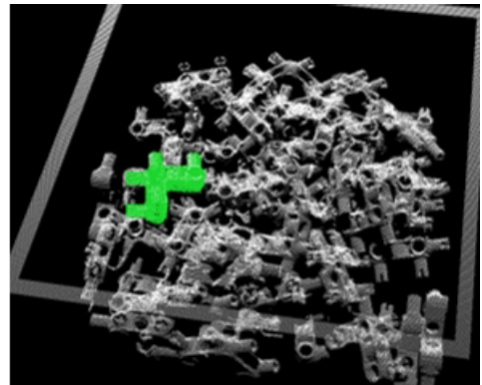
A etapa seguinte a amostragem foi a correspondência entre descritores e o cálculo do alinhamento. Os ensaios foram realizados com um modelo CAD convertido em nuvem de pontos. As cenas utilizadas consistiram em nuvens sintéticas modeladas por software e uma digitalização real com um sensor.

A figura 2.5 ilustra as cenas utilizadas e representa algumas identificações. Nas figuras 2.5a, 2.5b e 2.5c foram utilizadas nuvens sintetizadas, e na figura 2.5d está presente uma recolha com sensor.

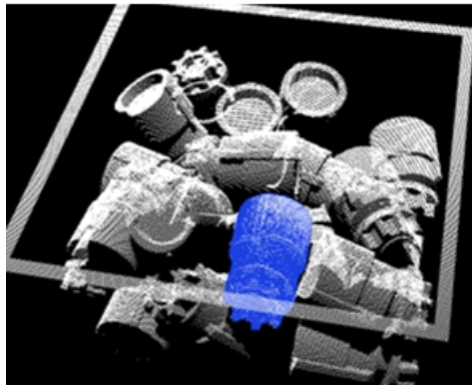
Os resultados obtidos por esta metodologia foram satisfatórios, com taxas de reconhecimento de 85% e tempos de ciclo de 2 segundos para a nuvem digitalizada. Estes resultados validam a metodologia e suportam este tipo de descritores para a utilização de modelos CAD.



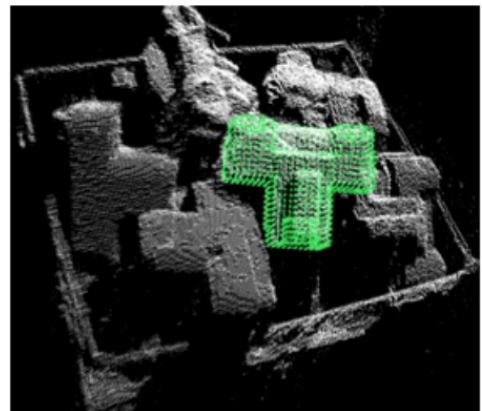
(a) Resultado da nuvem de pontos A (sintetizada).



(b) Resultado da nuvem de pontos B (sintetizada).



(c) Resultado da nuvem de pontos C (sintetizada).



(d) Resultado da nuvem de pontos D (Digitalizada).

Figura 2.5: Exemplo de resultados de reconhecimento para quatro nuvens utilizadas[9].

2.2 Implementações Industriais

2.2.1 InPicker

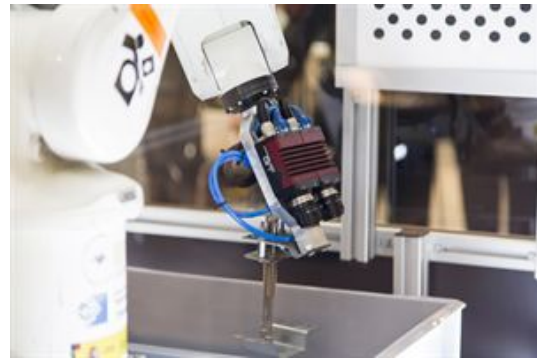
O software InPicker [10] é numa solução para bin-picking, desenvolvida pela empresa Infaimon em colaboração com a Universidade de Saragoça. O sistema baseia-se na utilização de um sistema estéreo de duas câmaras, na configuração exposta na figura 2.6b.

O sistema estéreo permite o levantamento de uma imagem de distâncias do contentor com objetos, baseando-se na disparidade entre as imagens das duas câmaras. As referências utilizadas para a identificação são definidas em função do tipo de objetos, podendo ser consideradas as furações, proporções ou outros detalhes descritivos.

O software possui uma interface com o utilizador, esta expõe ambas as imagens e representa as deteções no decorrer da execução, como ilustra a figura 2.6a.



(a) Janela de visualização do software.



(b) Manipulador e sistema estéreo.

Figura 2.6: Figuras representativas da solução InPicker proposta para bin-picking[10].

2.2.2 HALCON

O HALCON é um software destinado ao processamento de imagem para perceção. A sua utilização é feita como ambiente de desenvolvimento para soluções em visão artificial e outros tipos de perceção [11].

No contexto da temática o HALCON apresenta-se como uma ferramenta que possibilita o desenvolvimento de soluções de bin-picking. As funções que dispõe permitem utilizar diversos equipamentos como sistemas estéreo, câmaras 2D, e outros tipos de sensores, dada a sua capacidade para operar com imagens, imagens de profundidade e nuvens de pontos. Os modelos que podem ser utilizados também são bastante variados tal como digitalizações, modelos CAD ou até mesmo modelos construídos por programação [12].

As valências do software permitem, por exemplo utilizando um sistema estéreo e um modelo CAD do objeto executar a identificação e localização como a ilustrada na figura 2.7 [13].

O software como é uma ferramenta de desenvolvimento tem por objetivo o uso por outras empresas dedicadas a construção e instalação de soluções industriais a medida de cada cliente. A empresa Beltech é um desses exemplos de desenvolvimento de produtos com base em HALCON [14]. A figura 2.8 ilustra uma solução para bin-picking de tubos desenvolvida pela empresa, utilizando nuvens de pontos.

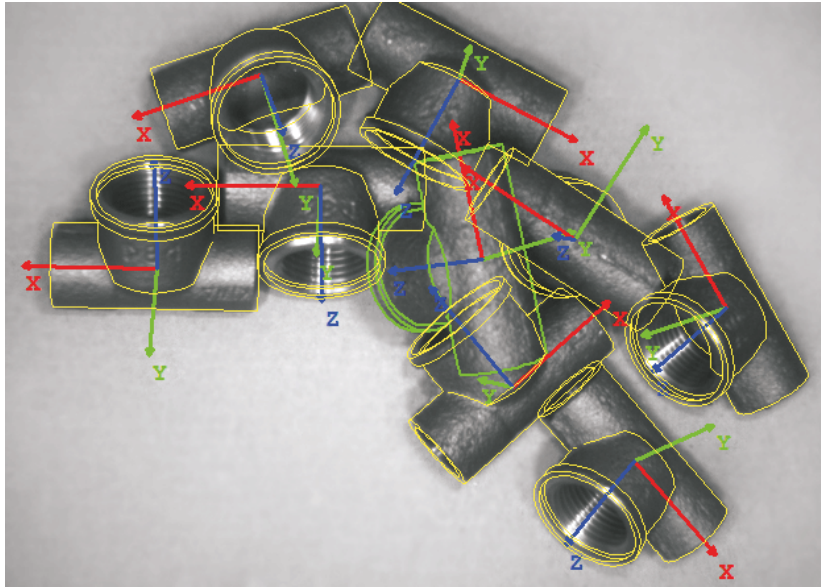


Figura 2.7: Imagem de detecção de objetos realizada em HALCON [13].

2.2.3 iRVision 3D

A marca de robôs Fanuc comercializa o seu próprio sistema para bin-picking [15], em que para além de um manipulador da sua marca dispõe também de um sistema de perceção 3D e o software necessário ao processamento, como se observa na figura 2.9.

O tipo de sensor utilizado consiste em duas câmaras e um projetor, operando como um scanner de luz estruturada. O projetor faz incidir um padrão de linhas sobre a cena, esta reflete o padrão de forma deformada. As imagens que são recolhidas pelas câmaras são processadas e permitem obter a informação 3D.

O processamento de imagem é executado por uma carta colocada no interior do controlador do manipulador. Esta abordagem dispensa o uso de processamento externo tornando a solução mais integrada.

O software de processamento de imagem, iRVision, é um produto também da marca. Este atua como ambiente de desenvolvimento para os algoritmos de perceção não só em 3D como em 2D [16].

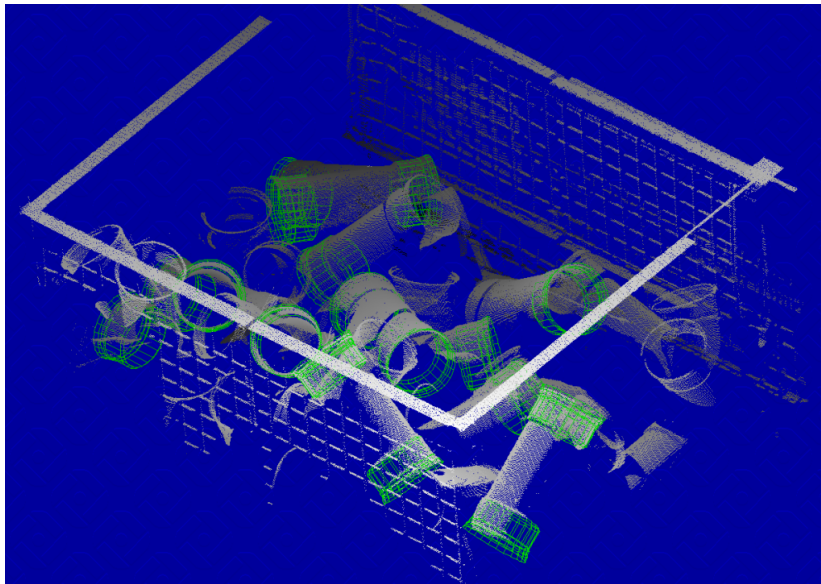


Figura 2.8: Nuvem de pontos de detecção de tubos realizada em HALCON [14].



Figura 2.9: Figura ilustrativa da solução fornecida pela Fanuc, com o manipulador e sensor da marca [15].

Capítulo 3

Infraestrutura Experimental

3.1 Hardware e Equipamentos

3.1.1 Sensor Kinect

O sensor kinect foi desenvolvido pela Microsoft em parceria com a empresa PrimeSense. O objetivo para a utilização deste equipamento foi a de um acessório para a consola de jogos Xbox 360, as vendas do sensor revelaram-se um sucesso tendo atingido um número de vendas superior a 10 milhões de unidades [17]. O baixo custo do equipamento despertou o interesse da comunidade utilizadora de perceção tri-dimensional, que o considerou uma alternativa em relação aos equipamentos utilizados até então.

O equipamento é constituído por um conjunto de sensores, como se pode observar na figura 3.1, sendo eles um conjunto de microfones, uma câmara rgb, um emissor de padrão infravermelhos e uma câmara de infravermelhos.

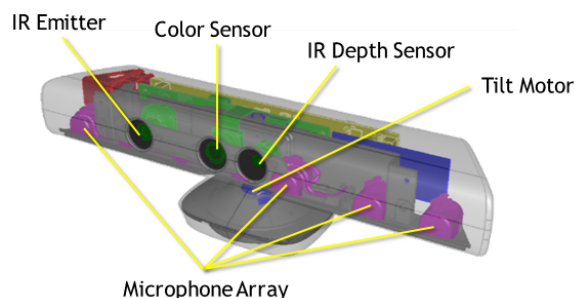


Figura 3.1: Imagem ilustrativa do sensor kinect [18].

O sistema foi pensado para uma utilização num contexto de periférico da consola de jogos, nessa utilização eram detetados os jogadores e respetivos movimentos que eram interpretados como comandos para o jogo, a decorrer no sistema. Deste as suas características estão otimizadas para esse efeito, como se observa na tabela 3.1.

O princípio de funcionamento do sensor para análise da profundidade, baseia-se na análise de distorção de um padrão infravermelho por ele emitido. A sequência processa-se do seguinte modo: o emissor de infravermelhos projeta um padrão constituído por pontos, como representa a figura 3.2. O espaçamento entre os pontos projetados aumenta em conformidade com o afastamento ao equipamento, deste modo a imagem de infravermelhos recolhida pelo sensor,

Tabela 3.1: Tabela de características técnicas do sensor [19]

Características do sensor	
Campo de visão angular	57° horizontal e 43° vertical
Frequência de aquisição	30 Hz
Alcance espacial nominal	640 x 480 (VGA)
Resolução espacial nominal (2 m)	3 mm
Alcance de profundidade nominal	0.5 m - 5.0 m (em OpenNi)
Resolução da profundidade nominal (2 m)	1cm
Tipo de conexão	USB (+fonte externa)

permite o cálculo das distâncias em cada pixel pela análise da distância entre pontos.

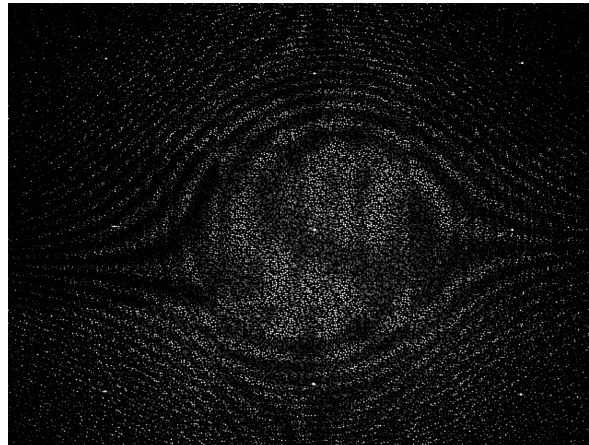


Figura 3.2: Imagem de infravermelhos recolhida pelo sensor da Kinect [19].

O princípio de funcionamento do sensor, baseado no espectro infravermelho, apresenta algumas limitações. O tipo de material é uma condicionante, dado que a sua superfície influencia os resultados obtidos. Para superfícies lisas e espelhadas o padrão é defletido não sendo detetado pelo sensor, e analogamente as superfícies mais rugosas provocam a difusão do padrão resultando no mesmo efeito. O ambiente de utilização do equipamento é também condicionado, uma vez que ao ar livre o nível de infravermelhos é bastante intenso, impossibilitando a utilização do equipamento. Este impedimento deve-se à saturação do sensor de infravermelhos, restringindo-se assim a aplicações dentro de portas.

OpenNI

O equipamento Kinect é constituído por um conjunto de sensores que permitem a sua funcionalidade, no entanto para a sua utilização é necessário também um software controlador que o torne operacional.

A API (Application Programming Interface) seleccionada para operar a Kinect foi a OpenNI, esta gere o tráfego de informação recolhendo e processando os dados fornecidos pelo equipamento para a aplicação. Conceptualmente a estrutura pode ser vislumbrada na figura 3.3.

A instalação do OpenNI careceu de alguma pesquisa, uma vez que existe um problema já documentado por outros utilizadores. Este é gerado pela versão Ubuntu 14.04 LTS utili-

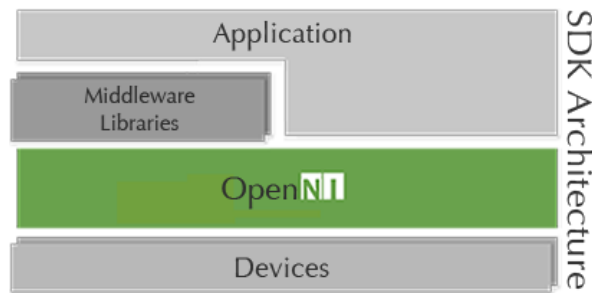


Figura 3.3: Figura esquemática da arquitetura SDK (Software Development Kit)[20].

zada entrar em conflito com o OpenNI bloqueando o acesso a Kinect. A solução passa pela recompilação da OpenNI, com algumas modificações descritas em [21].

Os ajustes aplicados ao OpenNI permitiram a continuidade da utilização do sensor sem mais problemas. A estrutura pretendida para solução baseia-se em ROS, e desse modo foi indispensável que a API se integrasse nessa arquitetura. O sistema ROS já se encontra preparado para este tipo de implementação, e utilização do sensor kinect, tendo um pacote que pode ser adicionado e inclui o OpenNI [22]. Este opera como um nó que desempenha as funções normais da API no entanto disponibiliza a informação pela publicação em tópicos.

Calibração da Kinect

O equipamento kinect, como já foi referido, é composto por um conjunto de sensores, de salientar duas câmaras, sendo uma RGB e outra de infravermelhos. Este tipo de sensores em particular requer uma calibração dos seus parâmetros intrínsecos, no entanto o controlador utilizado já possui esses parâmetros calculados para um equipamento padrão.

A calibração do sensor não é necessária para a maioria das aplicações. Contudo a utilização pretendida requer um rigor acima da média e, desse modo, realizou-se uma calibração de forma a que a parametrização fosse o mais adequada ao equipamento utilizado.

O sistema ROS possui um pacote dedicado a calibração de câmaras o *camera_calibration*, este permite a calibração dos parâmetros intrínsecos de câmaras em ambiente ROS. A metodologia baseia-se na utilização de um tabuleiro xadrez, deste vão sendo recolhidas várias imagens que fornecem diversidade para o cálculo. O processo está descrito com maior detalhe em [23]. A janela de utilização da ferramenta está representada na figura 3.4.

A calibração processa-se do mesmo modo para qualquer câmara, tendo sido calibradas a rgb e a de infravermelhos. Dado que estas câmaras estão integradas num equipamento é indispensável acautelar alguns detalhes. As particularidades para a calibração do sensor kinect são por exemplo os tópicos a subscrever para cada câmara, e os cuidados a ter com a câmara de infravermelhos, estando estes discriminados em [24].



Figura 3.4: Janela utilizada pelo modulo de calibração.

3.1.2 Objeto e Acessórios de Apoio

A escolha do objeto a utilizar no estudo sofreu diversas etapas, tendo-se estes tornado consecutivamente mais adequados às condicionantes da instalação experimental. O primeiro objeto a ser utilizado foi uma peça lego, ilustrada na figura 3.5. Este encontrava-se disponível no material do laboratório. Os motivos da opção pela mesma foram: o facto da sua dimensão ser apropriada a escala da instalação, a sua geometria ser facilmente modelável e possuir alguns detalhes que enriqueciam a nuvem recolhida.



Figura 3.5: Fotografia da peça de Lego utilizada.

No entanto surgiram de imediato alguns problemas. A peça devido a sua natureza plástica apresentava uma superfície lisa que refletia o padrão emitido pelo sensor, e deste modo gerava lacunas na nuvem de pontos recolhida. Os detalhes dos encaixes contidos na peça foram também eles foco de problema, dada a resolução do sensor não ser suficiente para que a nuvem de pontos recolhida descreva com exatidão as referidas geometrias. O descrito pode ser apreciado nas figuras 3.6 e 3.7.

Os problemas que surgiram na utilização da peça lego levaram a sua substituição por uma outra. Os principais aspetos que foram tidos em conta na escolha do novo objeto foram: uma maior rugosidade da superfície, de forma a evitar a reflexão do padrão infravermelho, e a ausência de detalhes de dimensão inferior a resolução do sensor. Dado que estes não seriam

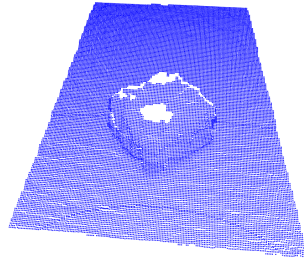


Figura 3.6: Figura da nuvem de pontos da peça Lego, perspectiva de topo.



Figura 3.7: Figura da nuvem de pontos da peça Lego, perspectiva lateral.

perceptíveis na nuvem recolhida. As especificações necessárias e os objetos disponíveis no laboratório levaram a utilização de um paralelepípedo de cartolina, como pode ser observado na figura 3.8.

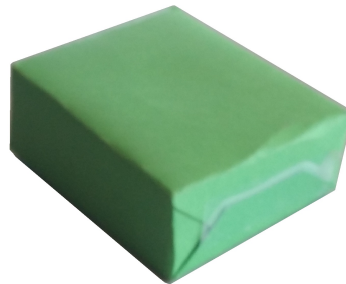


Figura 3.8: Fotografia do paralelepípedo utilizado.

O paralelepípedo utilizado não revelou qualquer problema no levantamento da nuvem de pontos. Contrariamente ao objeto anterior não existiam lacunas de pontos e a sua geometria, por certo que mais simples, é corretamente caracterizada. Os referidos resultados são visíveis nas figuras 3.9 e 3.10.

A nuvem de pontos obtida do paralelepípedo apresentou uma qualidade bastante aceitável, no entanto pela sua simplicidade e simetria acabou por ser trocada por uma alternativa. Estando esta dissertação focada na utilização de modelos CAD como referência, o uso de um objeto tão simples não demonstraria verdadeiramente a vantagem desta implementação em

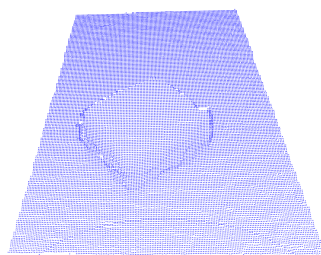


Figura 3.9: Figura da nuvem de ponto do paralelepípedo, perspectiva de topo.



Figura 3.10: Figura da nuvem de ponto do paralelepípedo, perspectiva lateral.

relação a outras metodologias.

A escolha de um novo objeto que atendesse a todas as especificações até a data levantadas, e de igual forma demonstrasse a mais valia desta implementação em detrimento de outras não foi imediata. Após alguma discussão e reflexão, foi idealizada e posteriormente modelada uma peça com uma maior variedade de superfícies e geometrias, e cuja dimensão e detalhes se adequassem aos equipamentos usados, como a figura 3.11 retrata.



Figura 3.11: Fotografia da peça utilizada.

A seleção do material teve em conta não só os requisitos impostos para detecção pelo sensor, mas também a sua futura manipulação pelo robô. A madeira surgiu como a escolha certa, dando uma rigidez suficiente e facilidade de maquinagem, mantendo tanto o preço como o peso baixos.

A construção das peças foi executada em partes, tendo como constituintes uma secção de varão redondo e outra de varão quadrado, que foram posteriormente colados. O acabamento da superfície foi terminado com lixa de forma a reduzir a reflexividade da mesma. Estando

as peças terminadas procedeu-se aos ensaios de digitalização com o sensor de forma a testar a qualidade das nuvens geradas. Como se pode observar nas figuras 3.12 e 3.13. As nuvens obtidas são de boa qualidade, conseguindo descrever as peças com detalhe, sem que se observe ruído excessivo ou lacunas de pontos. Os resultados foram de encontro ao esperado validando a geometria da peça e o seu acabamento superficial.

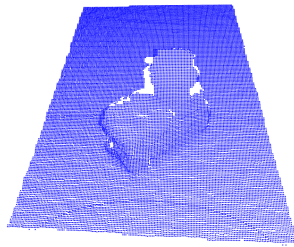


Figura 3.12: Figuras de nuvens de pontos do paralelepípedo, perspectiva de topo.



Figura 3.13: Figuras de nuvens de pontos do paralelepípedo, perspectiva lateral.

A fixação dos sistemas de visão e perceção tanto como os seus auxiliares sempre foi um problema, como ilustra a figura 3.14. No início a infraestrutura destinada ao uso do manipulador não possuía condições para assegurar essas funcionalidades. Ainda no decorrer da unidade curricular de Robótica industrial, e já conscientes de que a necessidade ia perdurar durante a dissertação, foi instalado um pórtico em perfil de alumínio. O pórtico permite a fixação de vários sensores, câmaras e sistemas de iluminação. A estrutura pela sua natureza e construção possibilita um ajuste da posição nos três eixos coordenados, deste modo podendo a mesma ser adaptada aos tipos de equipamentos e aplicações utilizados.

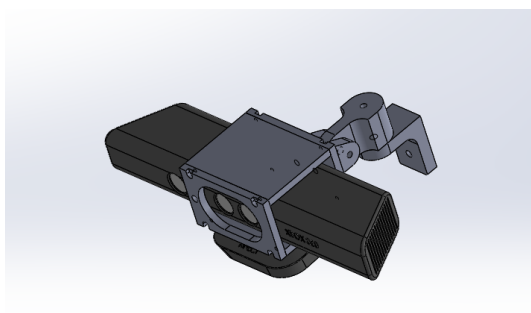
O sensor kinect foi projetado para um funcionamento apoiado numa superfície plana, como uma mesa, e dessa forma a sua base está adaptada para esse fim. A utilização neste trabalho requeria uma utilização do sensor suspenso, o que o pórtico referido anteriormente veio possibilitar. No entanto, o sensor não estava preparado para ser fixado a estrutura o que levou a uma fixação adaptada. A primeira solução passou por uma chapa dobrada com alguns parafusos, esta funcionou como algo provisório e para alguns ensaios. Contudo o equipamento não estava seguro e sendo este utilizado como referencial era imprescindível que a sua posição e orientação fossem não só estáveis como ajustáveis.

A solução definitiva para o apoio do sensor encontrou-se na modelação e conceção de um suporte para fixação do sensor a estrutura. Em primeiro lugar utilizou-se um modelo stl do sensor de forma a que o suporte desenvolvido fosse o adequado, de seguida foi modelado e feita uma montagem do conjunto como ilustram as figuras 3.15a e 3.15b.

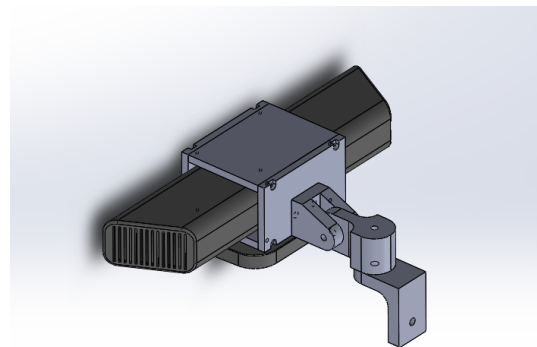
O suporte desenvolvido permite um aperto firme do sensor, sem que o danifique. O uso



Figura 3.14: Fotografia da infraestrutura inicial [25].



(a) Perspetiva de frente.



(b) Perspetiva perspectiva traseira.

Figura 3.15: Figuras da montagem do sensor com suporte em ambiente Solidworks.

de um caixilho, que o envolve, resolve os problemas de vibrações que ocorriam anteriormente. O caixilho é fixado ao pórtico pelo conjunto de três elos que permitem uma regulação da orientação do equipamento em três eixos de rotação. O resultado da instalação final pode ser observado na imagem da figura 3.16



Figura 3.16: Infraestrutura final.

O trabalho desenvolvido baseia-se num sistema de bin-picking, e como tal era indispensável um contentor onde os objetos a recolher se encontrassem armazenados. A escolha do contentor a utilizar baseou-se na dimensão da infraestrutura experimental e na maior semelhança possível com o utilizado industrialmente. Após alguma pesquisa e dentro do material disponível selecionou-se uma caixa plástica desdobrável, como ilustra a figura 3.17. Esta apesar de não ser exatamente igual, assemelha-se a um tipo de contentor plástico também ele desdobrável usado industrialmente para armazenamento e transporte de peças de média dimensão.



Figura 3.17: Caixa utilizada na atividade experimental.

3.2 Software e Bibliotecas

3.2.1 MeshLab

A necessidade de conversão de formatos de modelos utilizados foi o motivo que levou ao uso do Meshlab. O software é de código aberto, e tem por objetivo o processamento e edição de malhas tri-dimensionais. O MeshLab tem como base a biblioteca VCG dedicada a tarefas

de processamento de malhas.

A principal função do software é o processamento de malhas. Neste trabalho as funções utilizadas foram de conversão, dado que este permite a leitura e escrita em múltiplos formatos. A aplicação de filtros também é possível, quer para limpeza como para reconstrução. Dada a sua apetência para o pré-processamento de malhas para impressão 3D, permite o alinhamento por ICP e o fecho de lacunas, funções bastante úteis para digitalizações 3D [26].

3.2.2 Blender

O rumo tomado pelo trabalho gerou a necessidade de alterar os modelos 3D utilizados, o Blender surgiu como a ferramenta adequada para o fazer. O Blender consiste num ambiente de desenvolvimento de modelação 3D, em código aberto, este possibilita a execução de modelação, animação, edição de vídeo entre outros, sendo esta ferramenta bastante poderosa. A utilização deste programa é feita maioritariamente na criação de realidades aumentadas, filmes de animação e motor de jogos[27].

3.2.3 Robot Operating System (ROS)

O desenvolvimento de software para sistemas robóticos não é fácil, devido á complexidade e diversidade dos mesmos. As dificuldades são geradas por múltiplas causas tais como: diferentes equipamentos, comunicação complexas ou linguagens de programação distintas. A superação desses desafios é feito pelo desenvolvimento de diversas estruturas de integração de software sendo o ROS uma das soluções.

O ROS providencia uma solução do tipo sistema operativo, não sendo igual aos sistemas operativos convencionais. A estrutura do sistema é descentralizada, permitindo uma vasta adaptabilidade. Alicerçando o sistema existem quatro estruturas sendo elas pacotes, nós, tópicos e serviços.

Os pacotes são conjuntos de programas, estes podem conter nós, bibliotecas independentes, ficheiros de configuração ou qualquer outro módulo isolado. A divisão por este modelo leva a que os algoritmos sejam isolados e de complexidade baixa facilitando a sua reutilização.

Os nós são processos que correm paralelamente, estes podem ser desenvolvidos em diferentes linguagens como C++ e Python, no entanto o ROS possibilita a sua coexistência numa solução comum.

A comunicação entre nós é assegurada pelas estruturas: tópicos e serviços. Os tópicos baseiam-se no paradigma de publicador subscritor, ou seja um nó é a fonte de informação que publica num determinado tópico. A leitura dessa informação é executada por um subscritor ou vários que estão permanentemente a escuta. O nós podem ser simultaneamente publicadores e subscritores do mesmo tópico, tendo deste modo o fluxo de informação contínuo em ambos os sentidos.

Os serviços baseiam-se na filosofia de servidor cliente, o cliente requisita a informação ao servidor e este retorna-a. A tipologia permite também o fluxo de informação, embora a informação não seja publicada em contínuo mas apenas quando solicitada.

A comunicação utilizada em ROS impõe uma standardização, que desse modo simplifica a interação de diferentes equipamentos.

3.2.4 Point Cloud Library (PCL)

A evolução da complexidade de implementações em robótica gerou a necessidade de utilização de sistemas de percepção. As soluções utilizadas tiveram início em sensores de distância baseados em sonar e infravermelhos, contudo a evolução foi ocorrendo passando-se para equipamentos mais complexos que recolhem informação em maior quantidade e qualidade. Os sistemas mais avançados baseiam-se na recolha de nuvens de pontos, sendo estes sistemas de percepção 3D, muitos baseados em varrimento laser como a tipologia LIDAR. Os equipamentos para este tipo de aplicação, que surgiram inicialmente eram bastante dispendiosos não estando ao dispor de muitos laboratórios e investigadores, no entanto este cenário inverteu-se com o aparecimento da Kinect. O sensor devido ao seu baixo custo e boa qualidade veio expandir uso de nuvens de pontos em sistemas de percepção [28].

A massificação do uso de nuvens de pontos gerou a necessidade de uma estrutura para o processamento das mesmas, neste contexto surge a *Point Cloud Library* (PCL), sob o logótipo ilustrado na figura 3.18. A PCL consiste numa biblioteca de uso livre dedicada ao processamento para percepção, sendo a maioria das suas funções dedicadas ao uso em nuvens de pontos.



Figura 3.18: Logótipo utilizado pela PCL[28].

A programação foi estruturada e desenvolvida na totalidade na linguagem de programação C++, devido a essa característica permite-lhe fazer uso de outras bibliotecas, e aproveitar programação já desenvolvida para ações mais generalizadas. As bibliotecas utilizadas pela PCL são: a Eigen na utilização da álgebra linear, OpenMP e TBB (Intel Threading Building Blocks) para a paralelização do processamento, FLANN nas estruturas de busca, Boost na paralelização do processamento no processamento de imagem e na álgebra linear, e a VTK para a visualização gráfica.

A biblioteca PCL é facilmente utilizada na arquitetura ROS, vindo até integrada na mesma. Desta forma o desenvolvimento de soluções, e integração com outras, encontra-se bastante simplificado.

A crescente dimensão e complexidade características da PCL levaram a que esta fosse dividida num conjunto de módulos de pequenas bibliotecas, tal como se enumeram:

common Contêm as estruturas de dados comuns e métodos usados na maioria dos restantes módulos.

features Refere-se a atividade com descritores.

filters Engloba os filtros existentes tal como as estruturas para a sua implementação.

geometry Contêm as estruturas de dados para geometrias e algoritmos.

io Implementa as operações de escrita e leitura de ficheiros PCD(point cloud data) tal como as ações associadas.

kdtree Fornece a estrutura de dados do tipo Kd-tree.

keypoints Implementa as diferentes extrações de pontos chave, usados em pré-processamento.

octree Providência métodos eficientes para a criação e hierarquização de estruturas de dados em árvore.

outofcore Permite uma estrutura de dados do tipo octree para nuvens de pontos, que são demasiado grandes para armazenamento na memória principal.

recognition Contêm os algoritmos utilizados para operações de reconhecimento.

registation Engloba as metodologias para o alinhamento de nuvens.

sample_consensus Engloba as metodologias do tipo Sample Consensus, como o RAN-SAC.

search Fornece os métodos para busca de semelhança em diferentes estruturas de dados.

segmentation Agrupa os algoritmos referentes a segmentação de nuvens de pontos.

surface Agrupa todo o sector respeitante a manipulação de superfícies.

visualization É responsável por todas as classes de visualização.

O módulo de maior relevância para este trabalho foi o referente às Features, este corresponde as operações relacionadas com os descritores. As operações para todos os tipos de descritores implementados na PCL, são executadas recorrendo a classes e funções existentes neste módulo. Nas seguintes secções são descritos de forma detalhada os dois descritores utilizados neste trabalho.

Signature Histograms of Orientation(SHOT)

O descritor *Signature Histograms of Orientation* (SHOT) foi o resultado de um trabalho descrito em [29]. O descritor proposto baseia-se numa junção de duas tipologias de descritores usados até então, os baseados em histogramas e os baseados na marca gerada por conjuntos de histogramas, a assinatura.

O descritor é determinado do seguinte modo, em cada ponto chave é definido um referencial local, com base na normal do ponto e na sua vizinhança. O referencial e um raio previamente estipulado são utilizados para a criação da estrutura esférica ilustrada na figura 3.19.

A estrutura globular é então dividida em oito azimutes, dois hemisférios e duas divisões radiais (a figura 3.19 apenas apresenta quatro azimutes para simplificar a esquemática) perfazendo trinta e duas regiões.

Em cada região é calculado um histograma de onze elementos onde é armazenado o número de pontos nas várias posições do histograma. A distribuição ao longo do mesmo é definida em função do ângulo entre o vetor normal do ponto chave e os sucessivos pontos enquadrados em

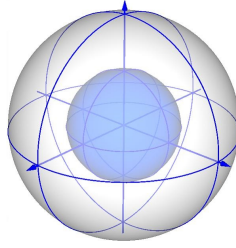


Figura 3.19: Estrutura de assinatura do descritor SHOT [29].

cada região. A divisão em trinta e duas regiões e os histogramas de onze elementos em cada uma delas perfazem o descritor, com trezentos e cinquenta e dois elementos, em cada ponto chave.

Fast Point Feature Histogram(FPFH)

O descritor *Fast Point Feature Histogram* (FPFH) foi apresentado em [30], este tem origem no descritor *Point Feature Histogram* (PFH) após algumas evoluções. De forma a melhorar a exposição será primeiro descrito o PFH e de seguida a transição para FPFH.

O descritor PFH baseia-se na vizinhança e respetivas normais de cada ponto chave, como ilustra a figura 3.20. Em cada posição de cálculo do descritor é definido um espaço esférico, os pontos contidos nesse volume constituem a referida vizinhança.

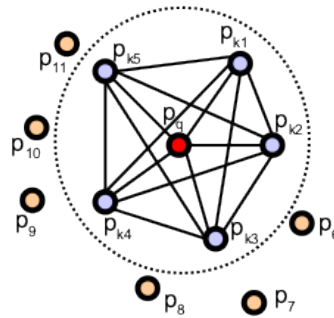


Figura 3.20: Estrutura de relação de pontos do descritor PFH[30].

O cálculo do descritor considera a relação entre todos os pares possíveis dos pontos da vizinhança, tendo em conta as seguintes imposições: considerando os pontos p_i e p_j ($i \neq j$) sendo p_i o ponto com menor ângulo entre a sua normal e a linha que liga o par. A metodologia baseia-se em primeiro lugar na definição de um referencial uvw determinado pelas seguintes expressões $u = n_i$, $v = (p_j - p_i) \times u$ e $w = u \times v$.

O referencial local serve de suporte ao cálculo de três ângulos descritivos. A determinação dos mesmos é executada pelas seguintes expressões:

$$\alpha = v \cdot n_j \quad (3.1)$$

$$\phi = \frac{u \cdot (p_j - p_i)}{\|p_j \times p_i\|} \quad (3.2)$$

$$\theta = \arctan(w \cdot n_j, u \cdot n_j) \quad (3.3)$$

O descritor FPFH é construído nos anteriores pressupostos com as seguintes alterações: os pares de pontos são apenas constituídos entre o ponto chave e os pertencentes a vizinhança, não sendo considerados os pares formados entre todos os pontos da vizinhança. A esta estrutura de cálculo é chamada *Simplified Point Feature Histogram* (SPFH).

O volume utilizado para a determinação do descritor é também distinto. Como podemos observar na figura 3.21, após definida a vizinhança do ponto chave é repetido o processo para cada um dos pontos contidos nessa vizinhança.

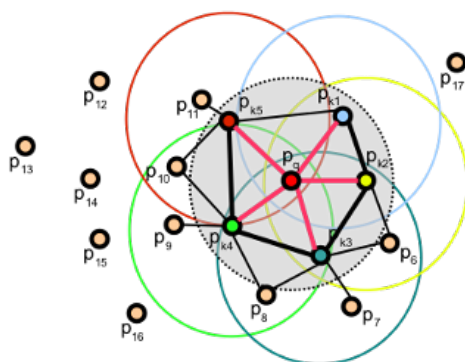


Figura 3.21: Estrutura de relação de pontos do descritor FPFH[30].

A construção da estrutura espacial determina todos os pontos a utilizar no cálculo do descritor, este é executado com base na seguinte expressão em cada ponto p ,

$$FPFH(p) = SPF(p) + \frac{1}{k} \sum_{i=1}^k \frac{1}{w_k} \cdot SPF(p_k) \quad (3.4)$$

O cálculo por este método leva a que a vizinhança dos pontos, da vizinhança do ponto chave, também afetem o descritor. O efeito não se manifesta com o mesmo peso por todos os pontos, isto porque é ponderado pelo fator w_k , que representa a distância entre os pontos p chave e p_k da respectiva vizinhança.

3.3 Arquitetura Geral da Implementação

A estrutura em que a implementação se constrói é condicionada pelo tipo de solução que se pretende desenvolver. Esta à partida não está definida, sendo moldada por cada módulo desenvolvido. Os módulos foram sendo constituídos à medida que se iam solucionando os problemas de base e em conformidade com as necessidades que surgiam.

O desenvolvimento de cada algoritmo tinha sempre em conta a sua integração numa metodologia ROS, em que cada módulo representava um nó, e a comunicação entre eles era assegurada através de tópicos. A escolha desta arquitetura baseada em ROS, prendeu-se pela versatilidade que permitia e com a simplicidade de implementar alterações.

As soluções que foram encontradas, e os tipos de dados a ser transferidos, levaram a sequência ilustrada no fluxograma 3.22. Apesar de não terem sido todos implementados, foram considerados sete módulos, que conjugados, representam a solução de bin-picking proposta.

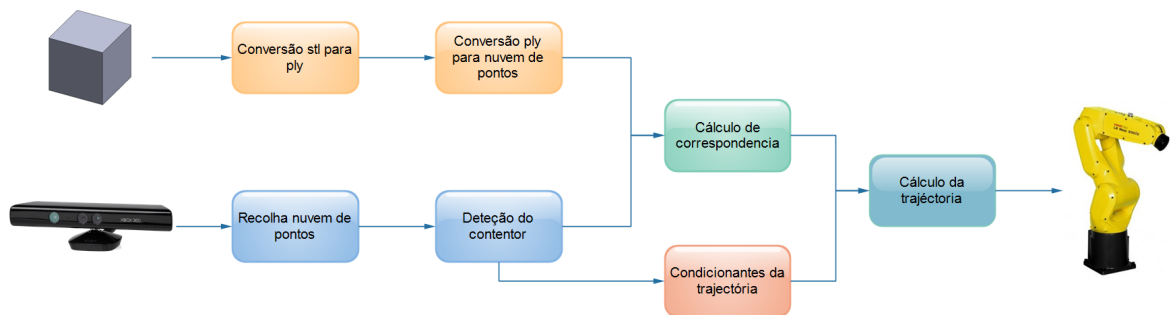


Figura 3.22: Diagrama geral da solução proposta para o problema de bin-picking.

Na utilização do modelo CAD em formato "STL" é indispensável a execução de duas tarefas: a conversão do ficheiro de "STL" para "PLY" e num outro módulo de "PLY" para nuvem de pontos (*Point Cloud Data* (PCD)). A primeira conversão é feita por uma razão de compatibilidade de formatos, pois a PCL não está preparada para a manipulação de ficheiros "STL", mas no entanto permite o uso de "PLY" apesar de ambos serem formatos semelhantes [31] [32] [33]. O formato "PLY" já permite a segunda conversão para o formato final de nuvem de pontos. Esta transformação é fortemente configurável permitindo o ajuste dos modelos obtidos às características desejadas.

A integração do sensor na implementação, e consequentemente da informação por ele recolhida, é executada pelo controlador OpenNI, que gere a interface com o sensor kinect. A forma como o controlador é utilizado consiste numa arquitetura ROS, como ilustrado na figura 3.23. O nó do controlador (elipse a vermelho) estabelece a ligação com o sensor, e de seguida publica a informação recolhida pelo mesmo num conjunto de tópicos (retângulos a azul). O tópico ROS que interessa à problemática é apenas o correspondente a nuvem de pontos `/camera/depth_registered/points` (destacado a verde), sendo este o único tópico subscrito pelo módulo desenvolvido para posterior gravação de dados e processamento.

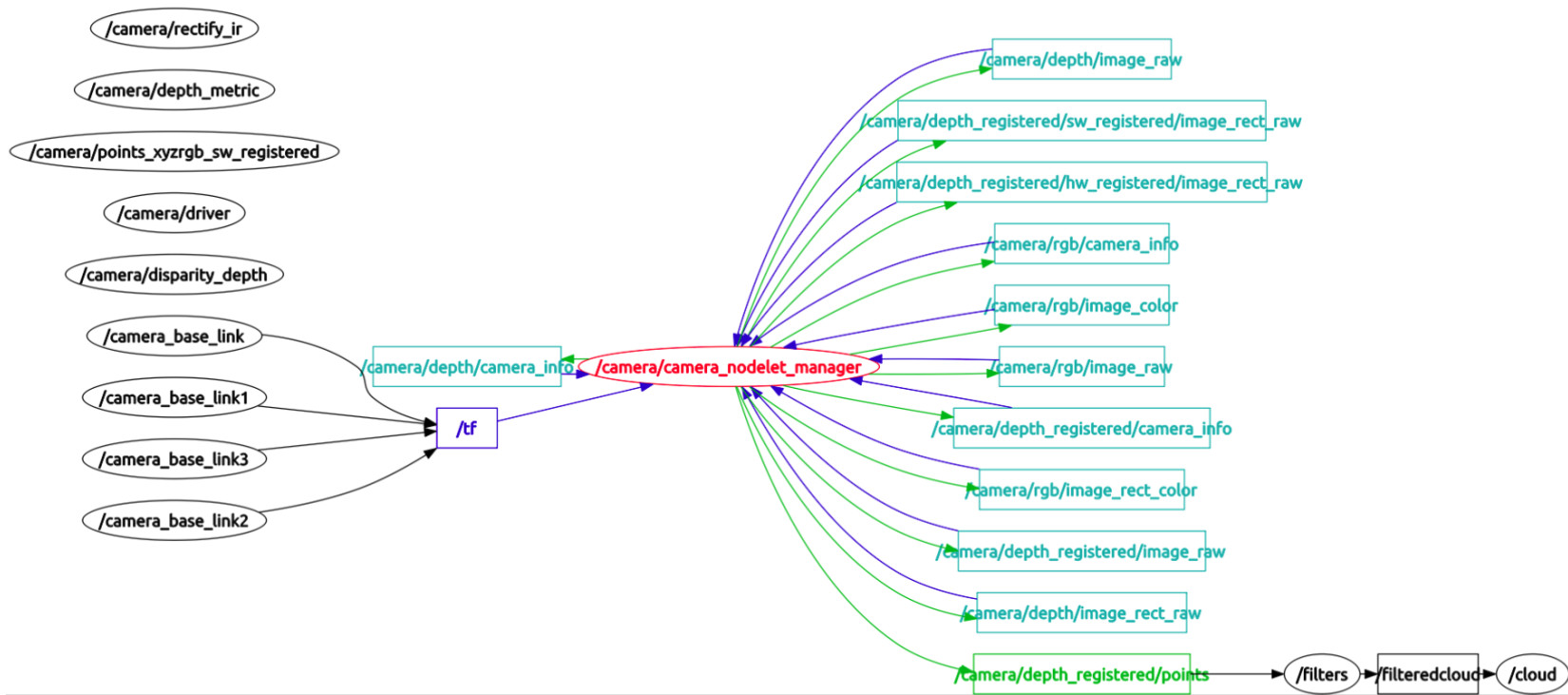


Figura 3.23: Diagrama de nós e tópicos ROS do sistema implementado.

A nuvem de pontos fornecida pelo sensor apresenta-se num estado bruto não tendo sofrido qualquer processamento. Deste modo o módulo que se segue à recolha consiste na deteção do contentor, como se observa na figura 3.22. A sua função é, de forma sequencial, isolar a informação útil dos pontos referentes às peças a recolher. A seleção permite o envio da informação estritamente necessária a cada módulo seguinte, facilitando o processamento e otimizando a execução. Neste caso, as peças a detetar seguem para o módulo de correspondência, e os limites do contentor para o condicionamento da trajetória.

A obtenção de uma nuvem de pontos modelo, com base no ficheiro CAD, e a recolha com o sensor da nuvem de pontos dos objetos a recolher convergem na execução do cálculo da correspondência. O referido módulo, utilizando a informação da nuvem de pontos modelo e da recolhida pelo sensor, determina as peças existentes e a respetiva posição.

As duas etapas restantes que não foram implementadas, consistem respetivamente: na definição do condicionamento imposto pelo contentor à movimentação do robô, e na determinação da trajetória condicionada, para a abordagem à posição e orientação determinadas no cálculo de correspondência.

O capítulo anterior veio expor as ferramentas utilizadas no trabalho desenvolvido, e a estrutura desenvolvida. O hardware foi selecionado em primeiro lugar em função da disponibilidade, no entanto foi necessário algum desenvolvimento tal como foi feito com as peças utilizadas e o suporte para o sensor. O software foi escolhido com vista a uma solução de baixo custo, tendo as soluções de código aberto sido privilegiadas. A opção foi interessante não só do ponto de vista dos custos como também pela possibilidade de modificar a programação base e desse modo aumentar a versatilidade. A arquitetura geral da implementação foi desenhada com o objetivo de responder às diferentes solicitações de uma implementação de bin-picking. Os módulos foram definidos de forma a isolar as diferentes ações necessárias, e possibilitar diferentes configurações.

Capítulo 4

Conversão CAD nuvem de pontos

A utilização do modelo CAD do objeto foi o primeiro problema abordado. É certo que o modelo CAD contém toda a informação geométrica e dimensional da peça, no entanto não é um formato idêntico à nuvem de pontos fornecida pelo sensor.

A disparidade de formatos inviabiliza a comparação pela metodologia utilizada. Este facto justifica a necessidade de conversões para igualizar os ficheiros utilizados e, dessa forma, permitir a análise comparativa por um algoritmo dedicado.

A utilização de modelos CAD como modelos em sistemas de identificação de objetos é uma metodologia já testada em diferentes variantes. O artigo apresentado em [34] é um exemplo deste tipo de implementação, utilizando o descritor local "Spin Image" e posteriormente uma classificação do tipo "bag-of-words" para a identificação de nuvens parciais.

A publicação [35] surgiu do trabalho de desenvolvimento do descritor "*Clustered Viewpoint Feature Histogram* (CVFH)". O descritor é do tipo global e tem como objetivo o reconhecimento e localização rápida de objetos rígidos. Os ensaios realizados basearam-se em modelos sintetizados de ficheiros CAD, e na recolha de nuvens com o sensor kinect.

No trabalho desenvolvido em [36] foi testada uma solução de conversão utilizando o modelo "STL" convertido para "*Wavefront Object* (OBJ)", com o software MeshLab e, de seguida, valendo-se da função "obj2pcd" da PCL, obtinham a nuvem de pontos. A metodologia acabou por ser abandonada pelos autores, uma vez que segundo eles ocorria um efeito de escala na nuvem de pontos produzida, o que impossibilitava a sua utilização como modelo. As conclusões referidas levaram-nos a optar pelo uso de digitalizações das peças feitas "a priori" com o sensor, abandonando a utilização do modelo CAD.

4.1 Primeira Abordagem

A primeira abordagem testada neste trabalho procedeu do seguinte modo: um cubo foi modelado no SolidWorks e guardado no formato "STL". A gravação do ficheiro foi, e terá sempre que ser, realizada em metros. Este detalhe é essencial para a posterior comparação com a nuvem de pontos que o sensor providencia na mesma escala. Pelo facto não existir nenhuma função na PCL para a conversão direta de "STL" para "PCD", foi utilizado o software MeshLab, que importou o ficheiro como "STL" e exportou-o como "PLY". Para além da conversão, não realizou qualquer alteração às dimensões ou geometria do modelo CAD contido no ficheiro, como ilustra a figura 4.1.

A biblioteca PCL possui funções para conversão do formato "PLY" em nuvem de pontos,

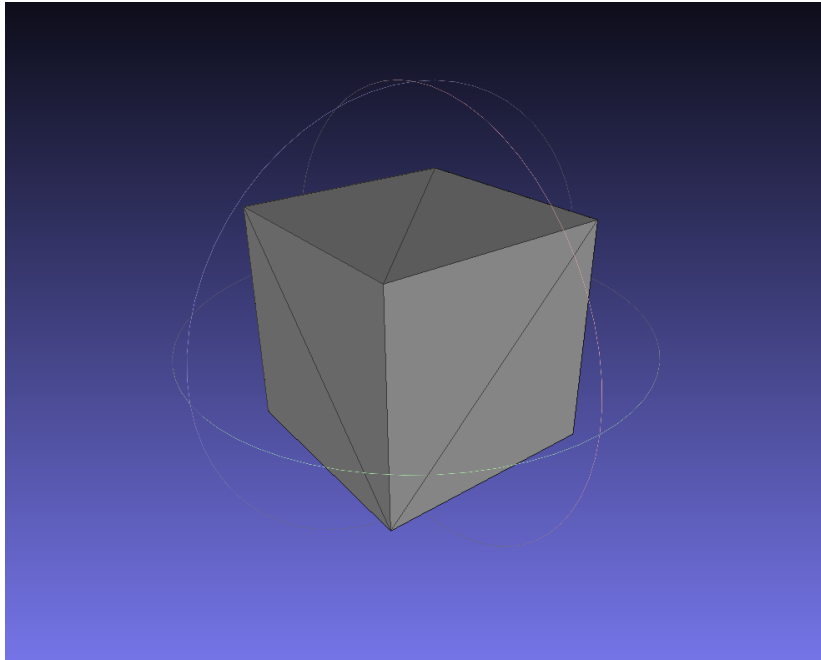


Figura 4.1: Cubo convertido no formato "PLY" em ambiente MeshLab.

tal como a "ply2pcd" que foi utilizada. A função "ply2pcd" tem como parâmetros de entrada o modelo em "PLY" e o ficheiro "PCD" no qual será guardada a nuvem de pontos convertida. A sequência de conversão de formatos e as respetivas ferramentas utilizadas, estão ilustrados na figura 4.2.



Figura 4.2: Fluxograma da primeira abordagem a conversão do modelo CAD. Nela estão representados os formatos e ferramentas utilizadas.

A conversão foi realizada com sucesso mas devido ao princípio de funcionamento da função "ply2pcd", o resultado não foi o desejado. Como se pode observar na figura 4.3 onde apenas os vértices do cubo foram convertidos em pontos. Este detalhe deve-se ao facto da função determinar os pontos da nuvem pelas coordenadas dos triângulos da malha "STL", ou seja, sendo o ficheiro "STL" um conjunto de triângulos definidos pelas coordenadas dos seus vértices, a função gera uma nuvem de pontos, correspondentes aos vértices da malha de triângulos.

A descrição por uma malha de triângulos de um sólido constituído por planos, como um cubo ou uma pirâmide, é conseguida com poucos elementos, o que resulta numa nuvem de pontos de resolução insuficiente.

A alternativa testada para uma melhor resolução da nuvem obtida foi a função "mesh2pcd" que ao contrário da anterior, simula a digitalização de um sensor fazendo a construção da nuvem aglutinando as nuvens parciais das diferentes perspetivas. A função baseia-se na utilização da classe "PCLVisualizer", onde encontramos funções como "renderView" e "renderViewTes-



Figura 4.3: Nuvem de pontos convertida pela função "ply2pcd".

relatedSphere" que possibilitam a simulação da digitalização de um sensor [37]. A sequência de conversão de formatos e as respectivas ferramentas utilizadas, estão ilustrados na figura 4.4.



Figura 4.4: Fluxograma da segunda abordagem a conversão do modelo CAD. Nela estão representados os formatos e ferramentas utilizadas.

A utilização da função "mesh2pcd" não gerou resultados satisfatórios. É certo que a resolução da nuvem obtida aumentou muito, como se constata na figura 4.5a, mas no entanto surgiu também bastante ruído em torno da nuvem, tanto com flutuações como em pontos isolados, representado na figura 4.5b.

Os resultados obtidos pelas funções "ply2pcd" e "mesh2pcd" foram conclusivos em relação a necessidade de um pré- / ou pós-processamento ao procedimento de conversão. Baseado nos requisitos e condicionantes da nuvem de pontos sintetizada procurou-se a melhor solução.

4.2 Metodologia com recurso ao Blender

Na mesma vertente do MeshLab, e também de código aberto, testou-se o software Blender, que permite uma manipulação muito mais alargada e aprofundada do ficheiro "STL". A função que suscitou o interesse pela utilização do Blender foi o comando "Remesh", que atua como um modificador e permite aumentar artificialmente a densidade da malha de triângulos. As potencialidades apresentadas pelo Blender surgiram como uma possibilidade para a reconsideração do uso da função "ply2pcd". Esta sequência de conversão, segue o procedimento ilustrado na figura 4.6.

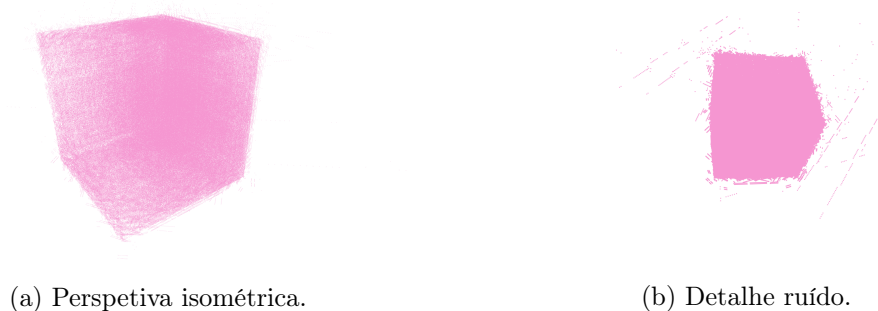


Figura 4.5: Nuvens de pontos do cubo obtida pela função "mesh2pcd".



Figura 4.6: Fluxograma da abordagem a conversão do modelo CAD. Esta foi possibilitada pela utilização do comando "Remesh" do Blender.

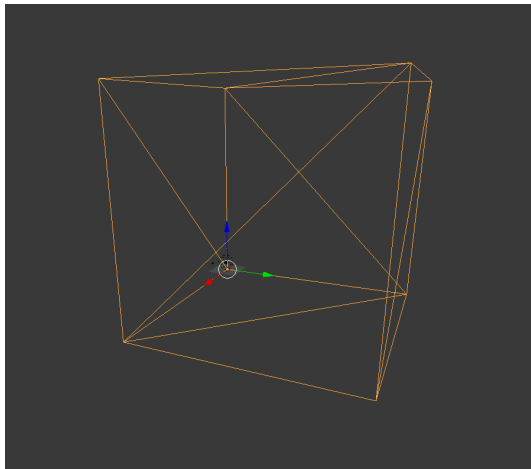
A análise que foi realizada ao Blender consistiu na importação de uma malha "STL", como está representado na figura 4.7a, e de seguida a aplicação do modificador "Remesh", resultando na malha ilustrada em 4.7b. O modificador pode ser utilizado com diferentes níveis que variam a resolução da malha obtida em consonância. Por fim a malha modificada foi exportada como "PLY", excluindo a necessidade de utilização do MeshLab.

A experiência realizada com o Blender foi bem sucedida e revelou resultados que validam a metodologia. A figura 4.8 apresenta a nuvem de pontos resultante que, ao contrário da anteriormente obtida pela função "ply2pcd", possui uma resolução ajustada à utilização pretendida, mantendo o rigor geométrico e dimensional.

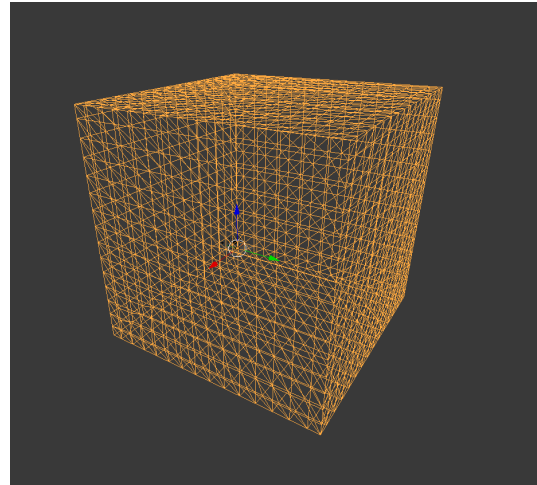
Os resultados obtidos no modelo do cubo foram satisfatórios, no entanto, o modelo utilizado era bastante simples. A validação da metodologia executou-se com modelos de maior complexidade. A forma correta de realizar o ensaio, e como a implementação também o permitia, foi a modelação e conversão de um provete de teste. O provete agrupou em si um conjunto de geometrias variadas e as respetivas transições entre elas, como representado na figura 4.9. O modelo é constituído por um toróide, um paralelepípedo e uma pirâmide quadrangular.

O provete modelado foi guardado no formato "STL", e de seguida importado para o Blender. A malha importada surgiu com a aparência ilustrada na figura 4.10; como é possível observar, a malha possui densidades de triângulos bastante distintas consoante as especificidades geométricas. A zona correspondente a pirâmide triangular é a que menos elementos possui. Este detalhe era esperado dado que cada face consiste num triângulo tal como os elementos da malha. No entanto, para o que é pretendido, o número é insuficiente.

A região paralelepipedica possui uma particularidade interessante. Na extremidade junto a pirâmide é composta por poucos elementos e na extremidade oposta apresenta uma densidade substancialmente superior, este detalhe deve-se as zonas para as quais a geometria transita. A passagem para a geometria de planos é descrita com poucos elementos, ao contrário de superfícies curvas que necessitam de melhor resolução. Na união do paralelepípedo com o



(a) Malha original.



(b) Malha modificada.

Figura 4.7: Malhas "STL"original e processada em ambiente Blender.

toróide, a secção passa de um quadrado para um círculo o que impõe a malha bastante diversidade nos elementos, quer em número como em geometria. O toróide está descrito por uma malha bem definida e uniforme, que para a conversão pretendida seria o suficiente.

O modificador "Remesh" foi aplicado e a malha foi exportada no formato "PLY". Como se pode observar na figura 4.11. A estrutura da malha modificou-se bastante, no ficheiro original ocorriam grandes diferenças na densidade de elementos, enquanto que após a aplicação do modificador a distribuição dos elementos e a sua dimensão tornou-se uniforme. De salientar as diferenças na extremidade da pirâmide em que o número de elementos aumentou para uma quantidade aceitável, para a utilização pretendida.

A uniformização dos elementos ao longo da zona paralelepipedica que, após conversão, permitirá a sua correta descrição. O detalhe do toróide foi algo que se julgou ser um foco de problemas na utilização do modificador, em que pelo facto de já apresentar originalmente um elevado número de elementos acabaria por ficar excessivamente definido: felizmente tal não aconteceu! Ao executar os diferentes testes anteriormente referidos, constatou-se que o modificador "Remesh" não aumenta o número de elementos pela divisão dos elementos já existentes, mas sim refazendo toda a malha, o que resulta na uniformidade obtida.

O fim do ensaio foi conseguido após a conversão em nuvem de pontos. A malha obtida "PLY" possibilita a utilização da função "ply2pcd" e desse modo foi a escolhida. A nuvem de pontos sintetizada está ilustrada na figura 4.12, como é visível no resultado obtido com o provete de teste, a conversão foi bem sucedida e deste modo validasse o método para peças de maior diversidade e complexidade geométrica.

O método de conversão do modelo CAD no formato "STL" para uma nuvem de pontos foi terminado e validado. No entanto nos ensaios de correspondência da nuvem de pontos sintetizada, com a cena recolhida pelo sensor kinect, não eram obtidos resultados de correspondência válida. Este insucesso levou a uma nova análise do modelo que estava a ser utilizado como referência e quais os possíveis motivos da sua ineficácia.

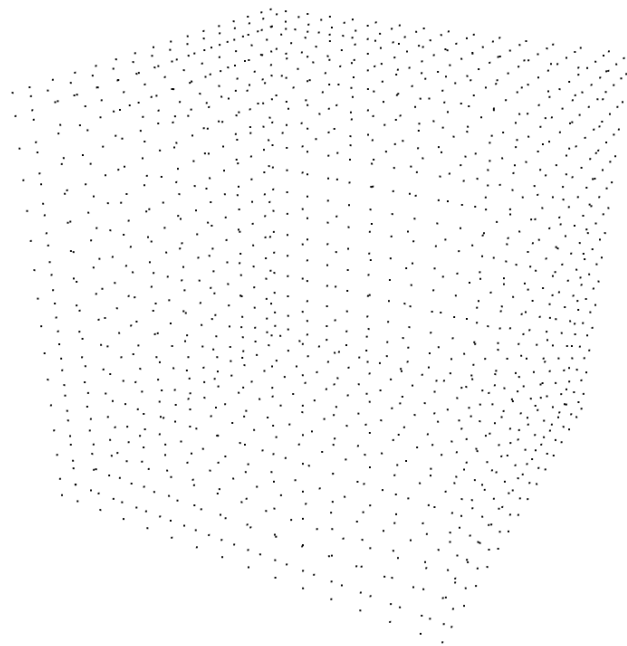


Figura 4.8: Nuvem de pontos convertida pela função "ply2pcd" usando uma malha modificada.

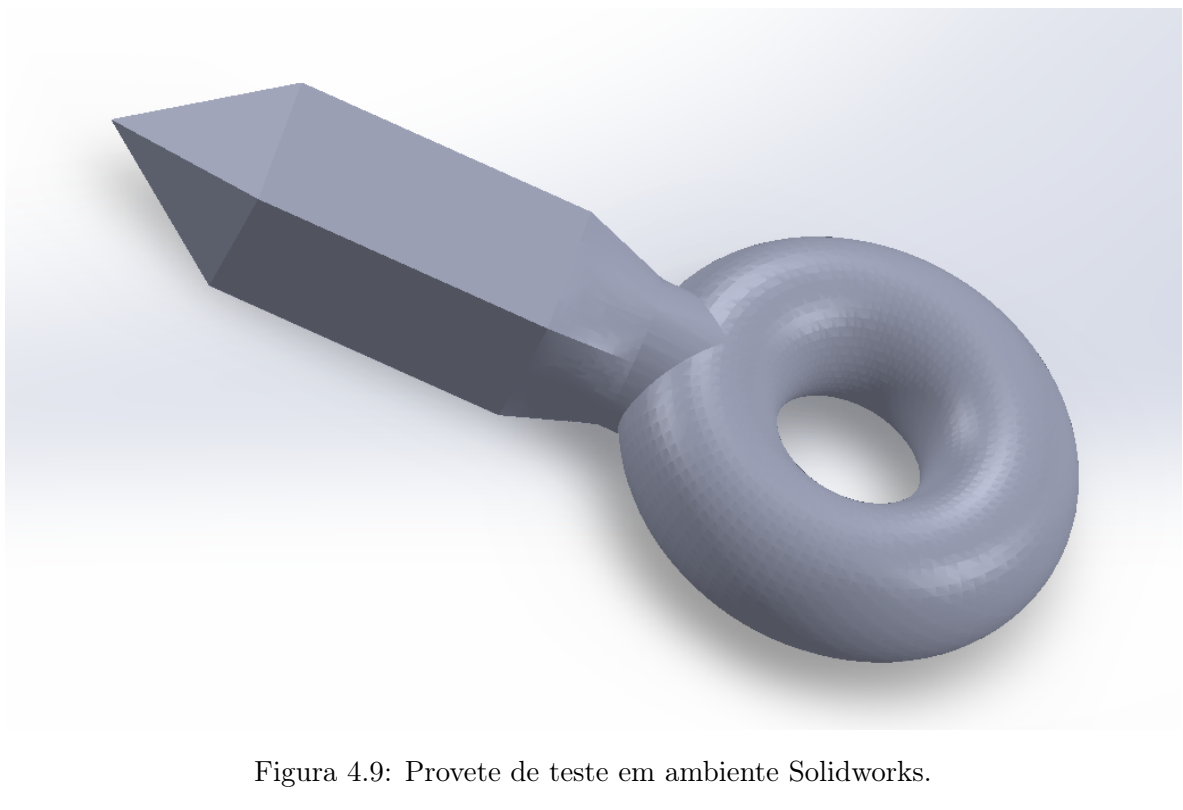


Figura 4.9: Provede de teste em ambiente Solidworks.

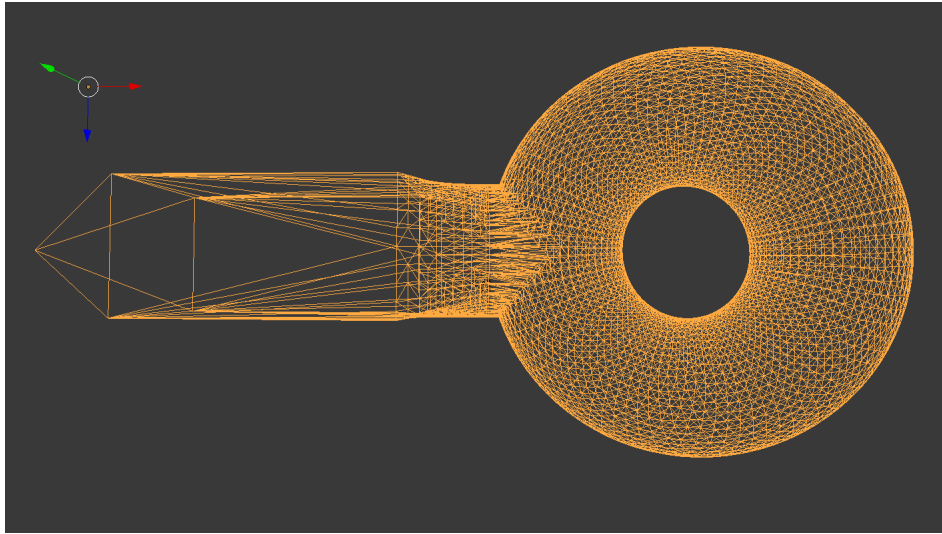


Figura 4.10: Malha "STL" do provete de teste em ambiente Blender.

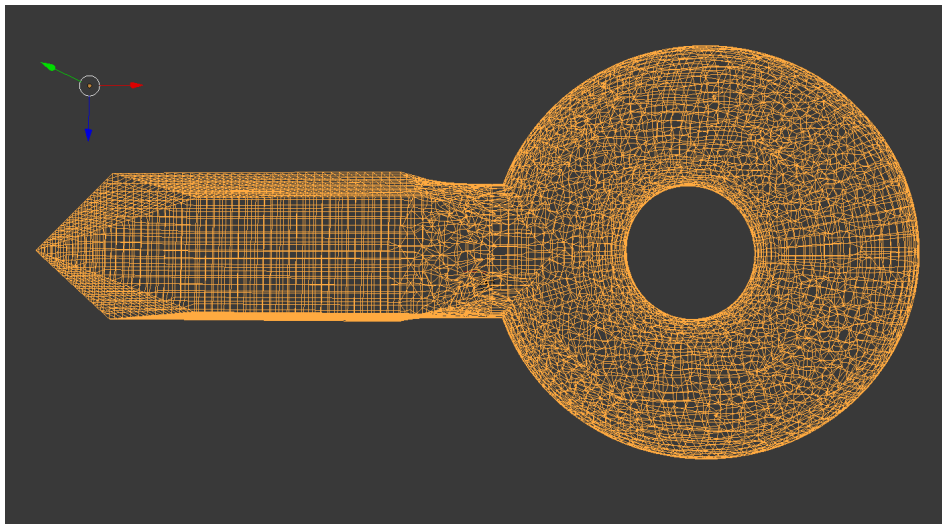


Figura 4.11: Malha "PLY" do provete de teste obtida após modificação em ambiente Blender.

4.3 Modelos parciais

A análise feita ao algoritmo de correspondência gerou a necessidade de criar modelos parciais, ou seja, que simulassem a oclusão que ocorre na digitalização de um sensor. O fenômeno consiste no efeito de perspectiva em que apenas a informação que é visível na direção do eixo óptico é recolhida, a restante encontra-se ocluída e portanto não é visível.

A criação de nuvens parciais foi ensaiada em primeiro lugar pela remoção de pontos a uma nuvem completa, desse modo a uma nuvem sintetizada do modelo foi aplicado um filtro em que era guardada uma nova nuvem com apenas os pontos que obedecem a uma ou várias condicionantes espaciais. O exemplo utilizado foi a sua posição em relação a um plano, contudo os resultados não foram os esperados, dado que o que se obtinha eram nuvens incompletas, como que cortadas e não com oclusões, o que não simula uma digitalização do sensor.

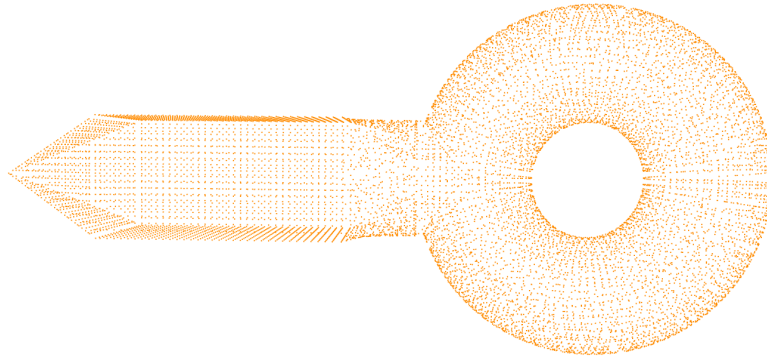


Figura 4.12: Nuvem de pontos do provete obtida pela função "ply2pcd", usando uma malha reconstruída em ambiente Blender.

A simulação do sensor não era uma tarefa de implementação trivial e, como tal, foi necessário procurar soluções existentes. Baseado em funções da PCL testou-se uma ferramenta que foi concebida para esse fim, o Virtual Scanner. Este algoritmo da PCL baseia-se no mesmo princípio do "mesh2pcd", no entanto não cria uma nuvem completa do objeto mas sim um diretório de nuvens parciais, e cada uma dessas nuvens simula uma digitalização do sensor.

A utilização do virtual scanner é feita pela invocação da função num terminal seguido do nome do ficheiro "PLY". A figura 4.13 representa a operação referida.

```
rodrigosalgueiro@rodrigosalgueiro-N56VZ: ~  
rodrigosalgueiro@rodrigosalgueiro-N56VZ:~$ pcl_virtual_scanner newcube.ply
```

Figura 4.13: Janela de terminal para a execução do comando virtual_scanner, sem configurações.

A potencialidade desta ferramenta é bastante elevada e, como tal, havia que estudar e tirar partido das suas configurações para a nossa aplicação. A primeira abordagem baseou-se na análise da opção de ajuda onde se apresentaram as configurações, como se observa na figura 4.14.

A informação disposta no terminal leva a uma análise mais detalhada das configurações a impor ao algoritmo:

- object_coordinates <0|1>** Define o referencial utilizado, podendo-se optar pelo do objeto ou do sensor.
- single_view <0|1>** Seleciona o modo de operação da função para uma única nuvem ou múltiplas ao redor do objeto.

```
rodrigosalgueiro@rodrigosalgueiro-N56VZ: ~
rodrigosalgueiro@rodrigosalgueiro-N56VZ:~$ pcl_virtual_scanner
Usage pcl_virtual_scanner [options] <model.ply | model.vtk>
* where options are:
  -object_coordinates <0|1> : save the dataset in object coordinates (1) or camera coordinates (0)
  -single_view <0|1>       : take a single snapshot (1) or record a lot of camera poses on a view sphere (0)
  -view_point <x,y,z>      : set the camera viewpoint from where the acquisition will take place
  -target_point <x,y,z>    : the target point that the camera should look at (default: 0, 0, 0)
  -organized <0|1>        : create an organized, grid-like point cloud of width x height (1), or keep it unorganized with height = 1 (0)
  -noise <0|1>            : add gaussian noise (1) or keep the model noiseless (0)
  -noise_std <x>          : use X times the standard deviation
rodrigosalgueiro@rodrigosalgueiro-N56VZ:~$
```

Figura 4.14: Janela de terminal com possibilidades de configuração do comando `virtual_scanner`.

-view_point <x,y,z> Estabelece a posição na qual se encontra a câmara no referencial do objeto.

-target_point <x,y,z> Estabelece a posição para a qual o eixo ótico da câmara se encontra direcionado.

-organized <0|1> Condiciona a disposição os pontos na nuvem recolhida, podendo estes estar organizados ou não.

-noise <0|1> Define a inclusão de ruído gaussiano.

-noise <x> Define o número de vezes que se aplica o desvio padrão ao ruído.

As configurações foram testadas e avaliadas, e nesse processo foi atingido o objetivo das vistas parciais, como se apresenta na figura 4.14, no entanto detetaram-se alguns detalhes que poderiam comprometer o funcionamento da metodologia, que foram os seguintes:

- A alteração da densidade de pontos ou resolução da nuvem não é possível;
- Em modo sequencial, onde são geradas várias vistas em torno do objeto, a distância do sensor ao modelo não é parametrizável;
- Numa amostragem de várias perspectivas, as nuvens de pontos são acumulativas, ou seja, à medida que são simuladas perspectivas elas vão se acumulando às anteriores, levando por exemplo, a que o último ficheiro seja uma digitalização completa do objeto;

O resultado da configuração, apresentado na figura 4.15, apesar de animador, apresentou algumas falhas como já foram apontadas. Desse modo foi necessário aprofundar o conhecimento sobre a ferramenta e para isso foi realizada mais alguma pesquisa. Dado que a PCL é uma biblioteca de código aberto foi possível aceder a programação do Virtual Scanner [38]. O código fonte `virtual_scanner.cpp` foi descarregado e analisado o que revelou algumas particularidades da ferramenta.

4.4 Modelos com ruído

A resolução da nuvem de pontos obtida é ditada pelo ângulo de afastamento entre os feixes simulados. O princípio de funcionamento do virtual scanner baseia-se na simulação de uma

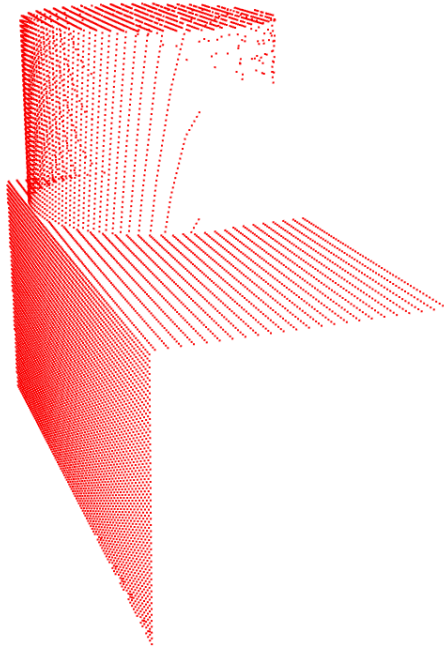


Figura 4.15: Nuvem de pontos parcial obtida pelo virtual scanner.

grelha laser, constituída por um conjunto de feixes, com origem no referencial do sensor, cuja intersecção com o modelo define um ponto da nuvem. Por este motivo, a densidade de pontos é imposta pelo ângulo de afastamento dos feixes. Apesar de não estar referido na mensagem de ajuda do comando, é possível a alteração do ângulo entre feixes. De forma à simulação apresentar bons resultados, foi necessário determinar esse valor para o sensor kinect.

A pesquisa efetuada sobre as características do sensor revelou os seguintes dados: em relação ao recetor infravermelhos as suas dimensões correspondem a 640x480 e o seu campo de visão angular 57° horizontal e 43° vertical [19]. Com base nesta informação, foi possível calcular a resolução angular, e a figura 4.16 esquematiza os dados referidos.

O cálculo das resoluções angulares foi executado da seguinte forma: α é a resolução angular horizontal e β é a resolução angular vertical, e definem-se pelos quocientes entre o campo de visão angular pelo número de pixels, como demonstram as equações 4.1 e 4.2.

$$\alpha = \frac{\text{amplitude angular horizontal}}{\text{número de pixels}} = \frac{57^\circ}{640} = 0.089 \text{ }^\circ/\text{pixel} \quad (4.1)$$

$$\beta = \frac{\text{amplitude angular vertical}}{\text{número de pixels}} = \frac{43^\circ}{480} = 0.089 \text{ }^\circ/\text{pixel} \quad (4.2)$$

As resoluções angulares calculadas apresentam o mesmo valor na horizontal e na vertical, implicando a uniformidade da grelha gerada. Utilizando os valores referidos para a simulação obtiveram-se nuvens de pontos com resolução próxima à das nuvens recolhidas com o sensor.

O modo de funcionamento sequencial, onde são geradas varias vistas parciais. Apresentou algumas lacunas na configuração, como já foi referido. No processo continuo de criação de vistas a distância a que o sensor simulado se encontra estava definido como um valor fixo,

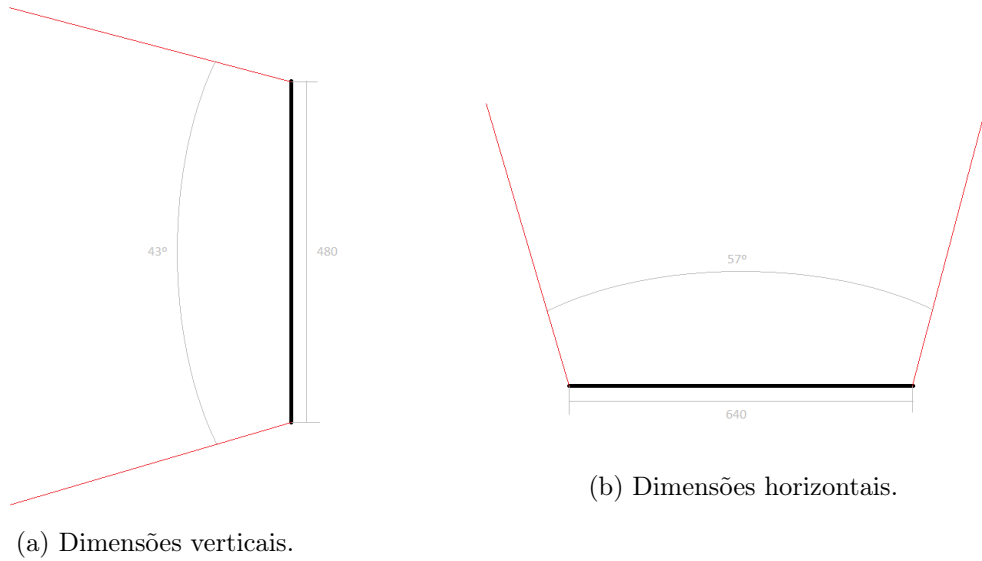


Figura 4.16: Figuras esquemáticas das dimensões do recetor infravermelhos.

sendo a resolução da nuvem gerada dependente da distância, este deveria ser reconfigurável para uma melhor aproximação ao gerado pelo sensor. A distância de digitalização passou a ser uma variável de entrada e, desse modo, solucionou-se essa fragilidade.

No decorrer das análises realizadas e do estudo dos descritores, constatou-se que a uniformidade e qualidade das nuvens geradas como modelos não produziam bons resultados de correspondência. Estas conclusões foram essenciais para perceber que o caminho que estava a ser seguido não seria o mais correto e era necessário encontrar alternativas no modelo gerado.

A primeira abordagem em busca de uma alternativa para o modelo utilizado, passou pelo estudo da tipologia de ruído do sensor kinect. Considerou-se o levantamento da nuvem de pontos de um plano, no caso específico o tampo da bancada do robô, pois é a geometria em que mais facilmente se detetam os desvios e variações. A nuvem recolhida, representada na figura 4.17, foi utilizada para fazer essa análise.

O algoritmo utilizado opera do seguinte modo, inicia a análise das dimensões extremas da nuvem e de seguida divide a nuvem em linhas e colunas, formando uma grelha. O número de linhas e colunas é por defeito 6, mas pode ser parametrizado ao iniciar o comando. Estando a nuvem dividida num conjunto de quadriculas, para cada uma delas calcula-se a média das distâncias, ao longo do eixo ótico, segundo a equação 4.3 e o desvio padrão por 4.4. Nas expressões são considerados n - número de pontos; e x_i - a distância em cada ponto medida pelo sensor.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (4.3)$$

$$\sigma = \sqrt{\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2} \quad (4.4)$$

Os valores de média e desvio padrão permitem uma análise mais objetiva sobre o ruído do sensor. Os ensaios executados com a instalação utilizada revelaram que o valor do desvio

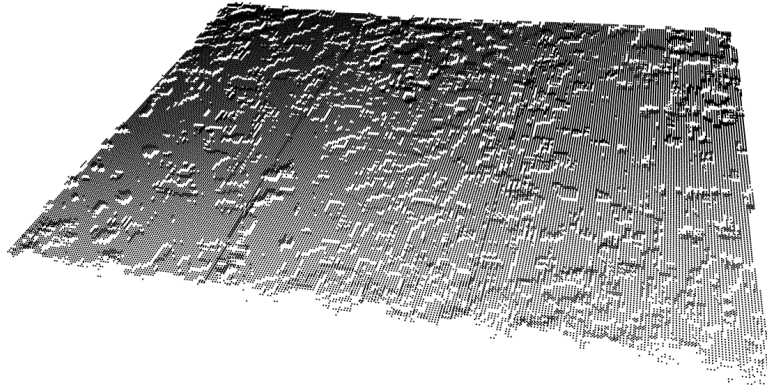


Figura 4.17: Nuvem de pontos obtida pela digitalização do tampo da bancada.

padrão, ao longo de toda a nuvem, oscilava em torno dos 2 milímetros.

O valor do desvio tão uniforme e ao manipular a nuvem de pontos, constatou-se que a nuvem apresentava estratificação, ou seja os pontos da nuvem, ao longo do eixo ótico do sensor, encontram-se agrupados por níveis, como se pode observar na figura 4.18.

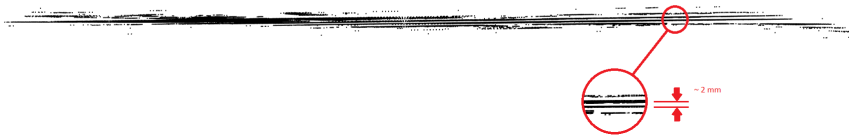


Figura 4.18: Vista lateral da nuvem de pontos obtida pela digitalização do tampo da bancada.

Os detalhes referidos foram seguidos do estudo da resolução do sensor. Após alguma pesquisa foi possível concluir que se tratava de um erro de quantização, sabendo que são utilizados 11 bit por pixel para armazenamento das distâncias e que o sensor mede valores entre 500 e 5000 milímetros [39], o erro de quantização é calculado da seguinte forma 4.5:

$$\Delta Z = \frac{Z_{max} - Z_{min}}{\text{número de bit}} = \frac{5000 - 500}{2^{11}} = \frac{4500}{2048} = 2.197 \text{ mm} \quad (4.5)$$

O resultado obtido vem justificar as suspeitas anteriormente referidas, sendo o erro de quantização de 2,197 milímetros. A limitação do sensor a este valor de erro é inevitável, e é o melhor valor a esperar dado que é influenciado por outros fatores agravantes. O erro pode parecer pouco significativo mas em peças de pequena dimensão, ou em detalhes menos pronunciados, assume-se como um fator crítico.

A criação de modelos com ruído surge como uma necessidade para ensaios de correspondência. Por esse motivo, produziu-se um pequeno algoritmo que introduz ruído artificial nas nuvens de pontos sintetizadas, do modelo CAD ilustrado na figura 4.19.

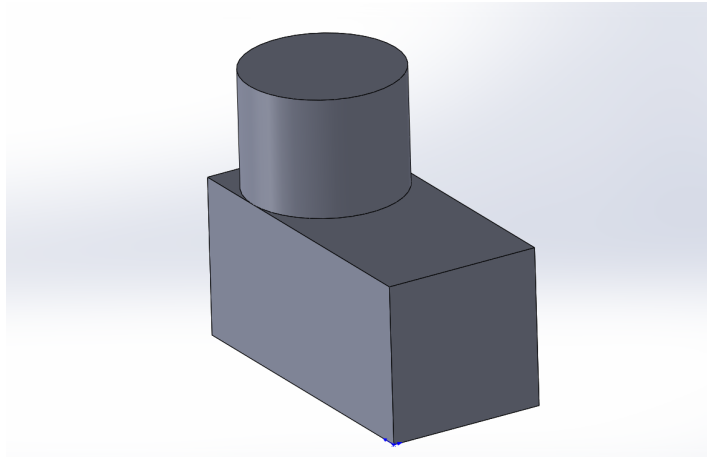
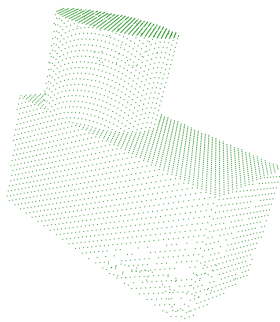
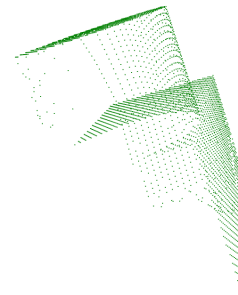


Figura 4.19: Imagem do modelo CAD original, utilizado na conversão em nuvem de pontos.

O algoritmo opera do seguinte modo: em cada ponto da nuvem, e tirando partido da função `rand()`, provoca uma variação. A variação é aleatória e de amplitude máxima definida previamente, nos três valores de coordenadas. A figura 4.20 é um exemplo de uma nuvem sintetizada parcial resultante do comando `virtual_scanner` adaptado, e como comparação a figura 4.21 ilustra a mesma nuvem após a aplicação de ruído de 2 milímetros em cada coordenada.



(a) Perspetiva sensor.



(b) Perspetiva lateral.

Figura 4.20: Nuvens de pontos parciais resultantes do `virtual_scanner` modificado.



Figura 4.21: Nuvens de pontos parciais resultantes do `virtual_scanner` modificado, com inclusão de ruído.

O tipo de ruído imposto não é exatamente igual ao gerado pelo sensor, pois tal depende não só do erro de quantização mas também das condições de iluminação e da própria tipologia das superfícies. No entanto, as variações uniformes de amplitude controlada manifestam-se importantes no funcionamento das correspondências.

O algoritmo de ruído foi incluído na versão adaptada do `virtual_scanner` de forma a agilizar o processo de construção de modelos, assim todas as etapas encontram-se agrupadas facilitando o processo de criação de modelos.

O processo de conversão do modelo CAD em nuvens de pontos passou por muitos avanços e recuos, tendo sido algumas vezes necessárias mudanças de estratégia. Revista toda a trajetória, e focando-nos sobre a implementação final podemos afirmar que a metodologia é fiável para a construção de modelos, altamente configuráveis e ajustáveis às solicitações dos algoritmos de correspondência.

Capítulo 5

Detecção do contentor com objetos

As problemáticas de bin-picking utilizam um contentor posicionado no espaço de trabalho. O contentor é uma parte integrante do sistema uma vez que condiciona as peças utilizadas, e atua como obstáculo na movimentação do robô. Na abordagem desenvolvida para uma solução de bin-picking o contentor desempenha outra valência pois serve de referência para a detecção e isolamento das peças.

A utilização do contentor como referência justifica-se pela robustez que confere à solução final. Desde que este se encontre no espaço de trabalho, é detetado, bem como as peças que se encontram no seu interior. A operação de um sistema com esta valência suplanta a necessidade de fixações e ajustamentos de posição, tornado-se mais rápido e versátil do que os sistemas que utilizam posições fixas dos contentores.

A implementação utilizada para a detecção e isolamento é composta por um conjunto de passos que, conjugados, completam a função do módulo. A referida estrutura é ilustrada na figura 5.1, que representa a sequência e as ações detalhadas em seguida.

5.1 Filtragem Inicial

A nuvem de pontos recolhida inicialmente pelo sensor contém mais informação do que a necessária, o que dificulta o posterior processamento. A atuação é dificultada pois a informação desnecessária atua como ruído, o que prejudica a eficácia das deteções. Como podemos verificar na figura 5.2, onde uma parte do chão também é capturada pelo sensor.

A solução encontrada para o problema passou pela execução de uma filtragem, com o filtro `pcl::PassThrough`. O filtro limitou as coordenadas dos pontos da nuvem, definindo um espaço de trabalho, como exemplifica a figura 5.3. Esta ação simplifica todo o futuro processamento.

O maior conjunto de pontos que permanece na nuvem, não sendo necessário, é a superfície da bancada. Assim a operação seguinte consiste na sua remoção. Na nuvem de pontos do espaço de trabalho, podemos observar que, devido a característica plana da bancada, podemos aproximar todos os seus pontos a um plano, e esse será o processo que nos permitirá a remoção.

A classe `pcl::SACSegmentation` permite identificar geometrias simples numa nuvem de pontos. O funcionamento procede da seguinte forma: na nuvem de pontos fornecida o algoritmo ajusta de forma iterativa, usando o método RANSAC, os pontos da nuvem a um plano dominante. Sendo o tampo da bancada o aglomerado plano com o maior número de pontos, é desse modo o identificado. A figura 5.4a ilustra a identificação do plano. Nela podemos ver os pontos integrados no plano destacados a vermelho. As zonas do fundo do contentor que



Figura 5.1: Sequência de detecção e isolamento do contentor e objetos.

não foram ocluídas pela existência de peças são, deste modo, também excluídas pelo uso de uma distância dos pontos ao referido plano mais abrangente.

O resultado da remoção do tampo da bancada e do fundo do contentor está representado na figura 5.4b, onde, para além do conjunto de pontos pertencente as peças, ainda permanece o bordo superior do contentor e os detalhes das peças. Estes pelo enquadramento do sensor também foram recolhidos.

5.2 Transformação do Bordo

O bordo superior do contentor delimita a área por ele ocupada e representa a barreira física que retém todas as peças. Estes detalhes fazem do aglomerado de pontos por si formado uma excelente referência para o isolamento das peças. De forma a isolar o bordo, e dado que o sensor se encontra normal à bancada para reduzir as oclusões, utilizou-se um filtro. O filtro já antes utilizado, `pcl::PassThrough`, impôs uma restrição na distância dos pontos ao sensor formando uma banda onde apenas o bordo do contentor ficou contido. Os pontos isolados do

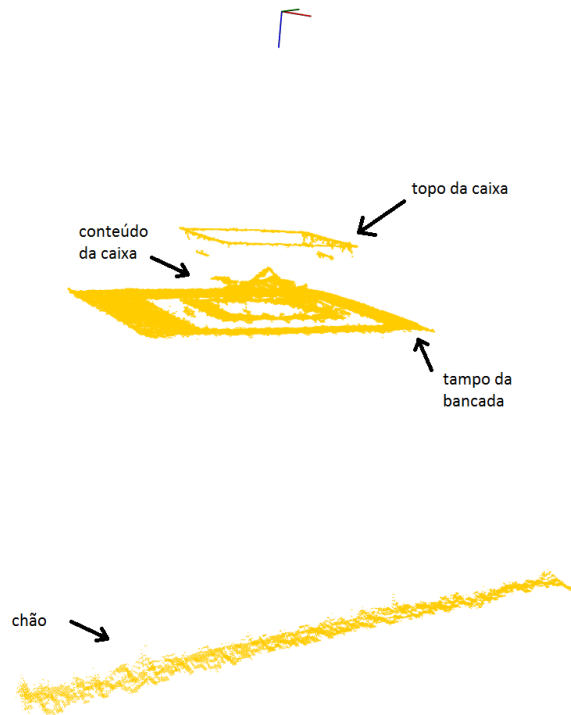


Figura 5.2: Nuvem de pontos recolhida pelo sensor na estrutura experimental sem qualquer processamento.

bordo estão representados na figura 5.5 com a cor vermelha.

O processamento seguinte consistiu numa etapa intermédia para a sequência pretendida. Nos pontos do bordo previamente isolados foi calculado o plano que mais pontos continha, usando o `pcl::SACSegmentation`, como explicado anteriormente, e de seguida os mesmos pontos foram projetados nesse plano. A projeção dos pontos foi executada com recurso ao algoritmo `pcl::ProjectInliers`, que permite o ajustamento dos pontos da nuvem a um modelo paramétrico previamente detetado, neste caso, o plano. O resultado obtido está representado na figura 5.6.

Nesta fase da implementação, o bordo já se encontra isolado e projetado num plano normal a superfície da bancada. Contudo este continua a englobar todos os pontos da nuvem quando o objetivo é o isolamento das peças. A metodologia utilizada passou pela redução da dimensão do bordo de referência, deste modo passando a formar uma barreira entre as peças e os pontos remanescentes do contentor. A redução do bordo foi executada com uma pequena função desenvolvida para esse efeito: a função "scale".

A função "scale" baseia-se na seguinte metodologia: em primeiro lugar é calculado o centróide da nuvem de pontos com o cálculo da média aritmética das coordenadas dos pontos. Utilizando o centróide como referência, em cada ponto são subtraídas as correspondentes coordenadas do centróide, tornando deste modo a posição do centróide no referencial absoluto. As coordenadas dos pontos no novo referencial são multiplicadas por um fator de escala que reduz a dimensão do aglomerado formado pelos pontos. No entanto não perdendo as suas pro-

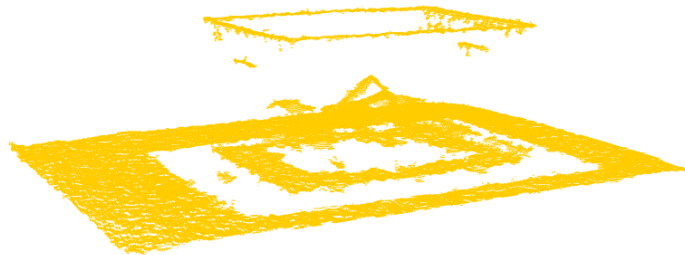


Figura 5.3: Nuvem de pontos do espaço de trabalho obtida após filtragem.



(a) Nuvem de pontos com tampo da bancada destacado a vermelho.

(b) Nuvem de pontos após remoção do tampo da bancada.

Figura 5.4: Nuvens de pontos do espaço de trabalho com identificação e remoção do tampo da bancada, e fundo do contentor.

porções. Para que o conjunto de todos os pontos utilize o mesmo referencial as coordenadas do centróide são somadas a cada ponto escalado. O resultado das transformações impostas aos pontos do bordo está destacado a vermelho na figura 5.7.

O bordo escalado representa um limite de separação das peças, que, no entanto por si só não é suficiente para essa ação. Para a obtenção de um formato que suporta a futura extração, foi calculado o polígono convexo que delimita o referido bordo.

A biblioteca PCL possui uma classe que permite a determinação do polígono convexo: a `pcl::ConvexHull`. Esta classe(função) tem como parâmetro de entrada a nuvem de pontos da qual se pretende que a geometria seja calculada. O retorno da geometria convexa é feito numa

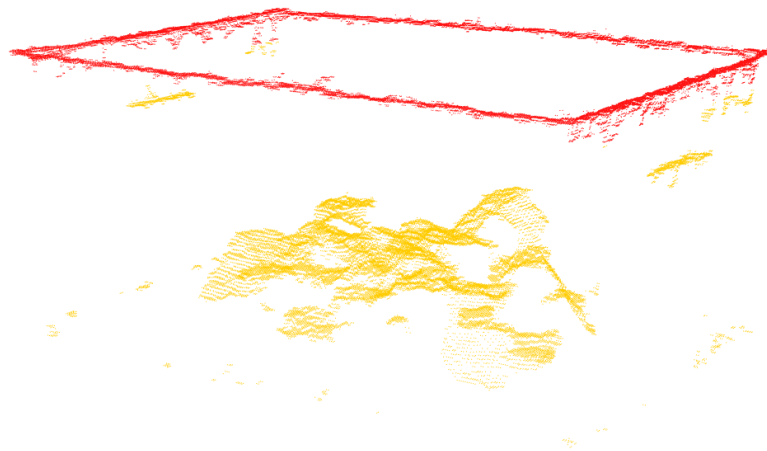


Figura 5.5: Nuvem de pontos do contentor após isolamento do bordo.

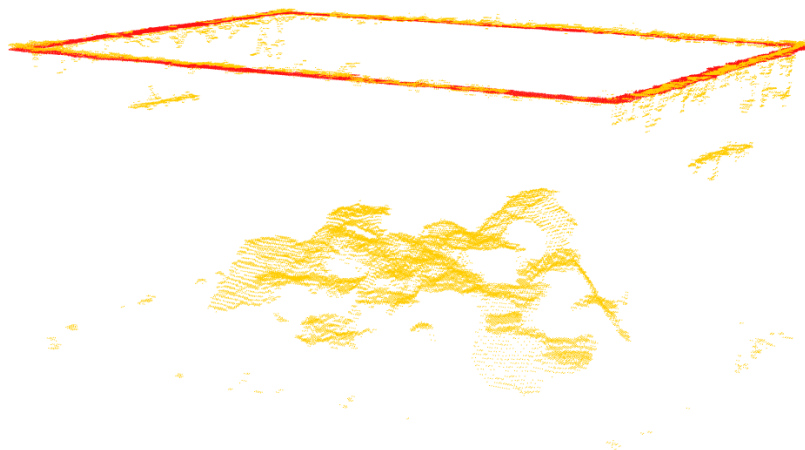


Figura 5.6: Nuvem de pontos do contentor após projeção dos pontos do bordo.

nuvem de pontos onde os pontos que delimitam a nuvem são armazenados, (figura 5.8).

5.3 Isolamento e Extração

A geometria, do polígono convexo, calculada possibilita a extração dos pontos referentes às peças. Para esse efeito recorreu-se a classe `pcl::ExtractPolygonalPrismData`. O algoritmo opera do seguinte modo: é definida a nuvem de entrada onde estão contidos todos os pontos, a extrair e a excluir, e posteriormente a geometria convexa. Para ser utilizada neste algoritmo a geometria tem de estar obrigatoriamente contida num plano, e deste modo justifica-se a anterior projeção dos pontos no plano.

A extração é executada numa operação semelhante a extrusão, ou seja a geometria fornecida é prolongada na direção normal ao plano onde está contida em ambos os sentidos. As

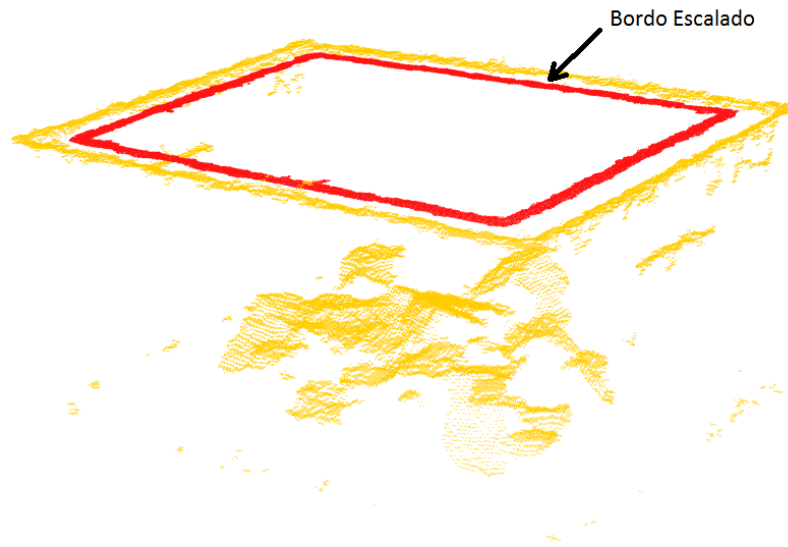


Figura 5.7: Nuvem de pontos do contentor com redução de escala dos pontos anteriormente projetados.

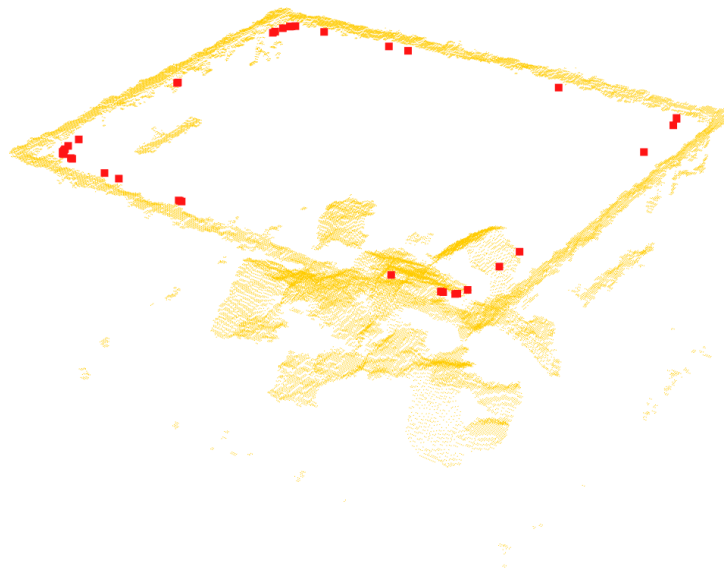


Figura 5.8: Nuvem de pontos do contentor com pontos do convex hull destacados.

distâncias da referida extrusão em ambos os sentidos são parâmetros de entrada do algoritmo. Por fim são obtidos os índices dos pontos incluídos no volume definido, como é visível na figura 5.9 com os pontos a azul. O tipo de retorno do algoritmo não permite que o processo termine nesta etapa tendo a estrutura de ser convertida em nuvem de pontos.

O isolamento da nuvem de pontos das peças requer uma pequena conversão, dado que os índices dos referidos pontos já são conhecidos basta extrair os pontos correspondentes a esses índices para uma nova nuvem. A classe `pcl::ExtractIndices` foi pensada para a operação

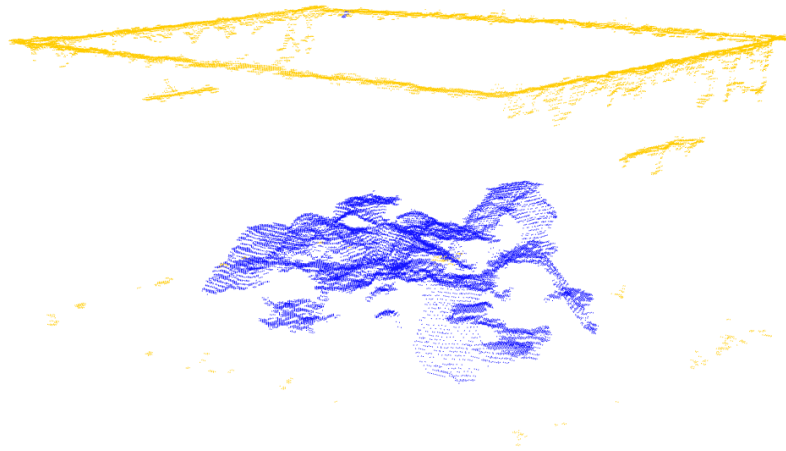


Figura 5.9: Nuvem de pontos do contentor com pontos correspondentes as peças destacados.

pretendida, logo bastou fornecer a nuvem original e os indices dos pontos a extrair para obter a nuvem final com as peças isoladas, como se ilustra e na figura 5.10.

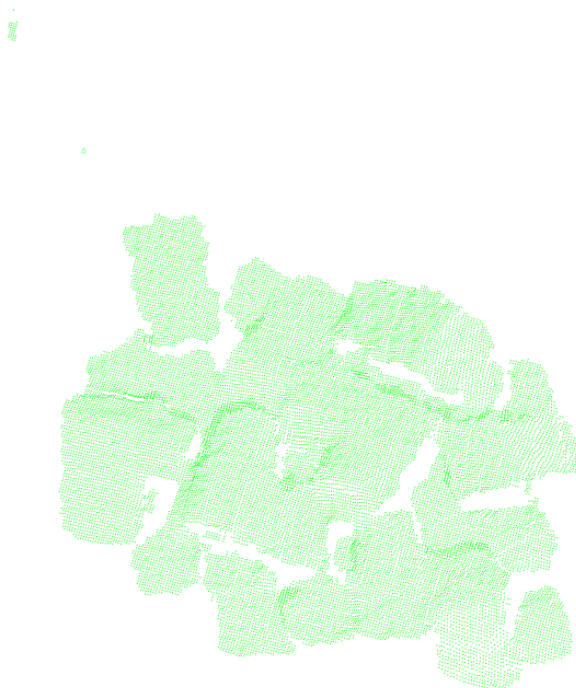


Figura 5.10: Nuvem de pontos das peças.

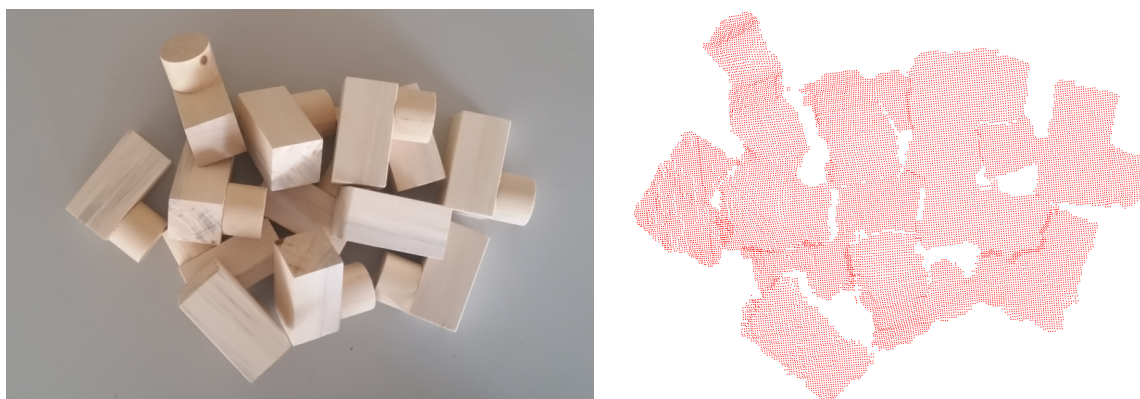
Capítulo 6

Determinação da Correspondência

O foco de um trabalho em "bin-picking" é a identificação e localização dos objetos, esse é o processo central que conjugado com outras tarefas perfaz o ato de bin-picking na verdadeira acepção da palavra.

De forma a identificar um objeto é necessário conhecer-lo, só assim podemos distingui-lo de entre outros. A metodologia para a identificação do objeto é a utilização de uma referência, neste caso um modelo do objeto. Este capítulo vem expor a metodologia para, utilizando a informação recolhida pelo sensor e o modelo, executar a identificação e localização do objeto.

Em primeiro lugar foi executada alguma pesquisa, sobre os tipos de implementação já feitos e em especial nos tipos de implementação que a PCL possibilitava. Os resultados da pesquisa remeteram-nos para uma metodologia baseada em descritores. A análise da cena de trabalho que se observa na figura 6.1a, representa o conjunto das peças. A observação da imagem sugere uma fácil identificação de cada objeto, no entanto a nuvem de pontos recolhida, como ilustra a figura 6.1b, revela muito mais homogeneidade. Conclui-se de imediato a ausência de um critério de segmentação e como tal a impossibilidade da execução da mesma. Este facto que afastou a hipótese de utilização de descritores globais.



(a) Imagem do conjunto de peças a recolher, da perspectiva do sensor. (b) Nuvem de pontos de conjunto de peças obtida com o sensor.

Figura 6.1: Imagem do conjunto peças e nuvem de pontos recolhida das mesmas.

A utilização de descritores locais é uma metodologia para a qual a PCL já se encontra preparada e na qual já existe bastante trabalho desenvolvido. O tutorial em [40] formou a

base para a construção da primeira sequência de correspondência. A abordagem aos descritores foi facilitada pela consulta do artigo [41], que expôs as possibilidades que a PCL dispunha tal como as suas características e método de utilização.

6.1 Metodologia Base

A sequência implementada pode ser analisada na figura 6.2, que representa o fluxograma de operações realizadas na nuvem de pontos do modelo, e da cena, até à determinação das posições e orientações das peças.

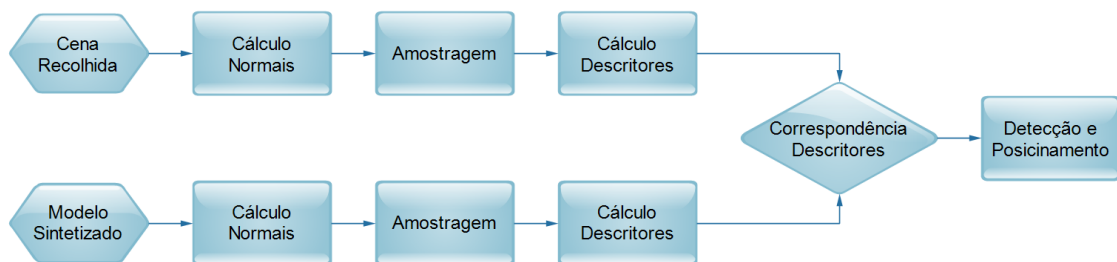


Figura 6.2: Fluxograma do conjunto de ações na sequência de correspondência.

A primeira ação imposta tanto à nuvem modelo como à recolhida pelo sensor é o cálculo das normais. O cálculo das normais foi executado pela utilização da classe `pcl::NormalEstimation`, fornecendo a nuvem de pontos e o número de pontos da vizinhança a utilizar para o cálculo o algoritmo devolve a estrutura das normais calculadas, com o aspeto ilustrado na figura 6.3.

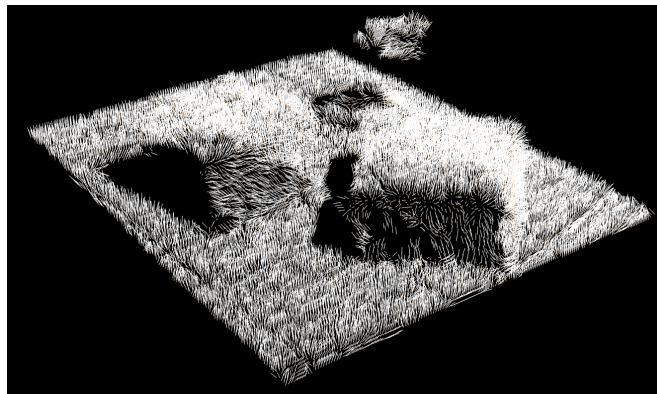


Figura 6.3: Exemplo de normais calculadas numa nuvem de pontos recolhida.

Obtidas as normais de cada nuvem, o passo seguinte consistiu numa amostragem das nuvens de forma a extrair os pontos chave. A amostragem justifica-se pela redução do peso computacional imposto e pelo notório aumento da velocidade de processamento. A classe `pcl::UniformSampling` foi a utilizada que se baseia numa redução da densidade de pontos sendo a distância mínima entre pontos, previamente fornecida, o critério de exclusão. A redução de dimensão está ilustrada na figura 6.4, onde os pontos a laranja representam a nuvem original, e os azuis os pontos chave.

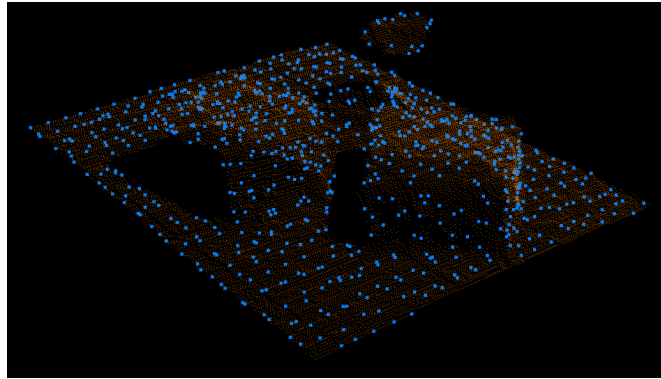


Figura 6.4: Exemplo de amostragem calculada numa nuvem de pontos recolhida, nesta é possível observar o resultado obtido pela redução da densidade de pontos.

As normais calculadas e os pontos chave selecionados são a base para a etapa seguinte. O cálculo dos descritores varia em termos de método de cálculo e dimensão, porém baseiam-se maioritariamente nas normais da vizinhança de cada ponto chave. E, em alguns casos, nos valores de cor caso estes a possuam. Deste modo, as classes de cálculo dos descritores são bastante semelhantes, variando nas estruturas de armazenamento dos descritores. O descritor utilizado foi o SHOT pois este era o exemplificado no documento de apoio e era considerado um bom descritor na restante literatura.

Os descritores de cada nuvem permitem a comparação entre elas. Esta análise é feita pela busca de correspondências. A análise baseia-se na classe `pcl::KdTreeFLANN` que permite que em cada ponto chave da cena recolhida se procure uma correspondência nos pontos chave do modelo. A metodologia utilizada está representada no algoritmo 1.

```

Define descritores do modelo como entrada;
for cada descriptor da cena do
    | Calcula a correspondência de maior semelhança;
    | if distância(correspondência) < distância mínima then
    | | Armazena a correspondência entre descritores;
    | end
end

```

Algorithm 1: Algoritmo de determinação de correspondência entre descritores.

A última etapa da obtenção da correspondência consiste na deteção e localização dos objetos. O algoritmo baseia-se no "clustering"/ ou agrupamento de correspondências de descritores consistentes entre si, e que determina o número de deteções válidas e as transformações correspondentes em relação ao referencial do sensor. A execução foi implementada na classe `pcl::GeometricConsistencyGrouping`.

O primeiro ensaio foi realizado sobre uma nuvem recolhida do paralelepípedo referido anteriormente na subsecção 3.1.2. Para além deste, foram colocadas outras peças distintas de forma a impor diversidade e complexidade ao teste. A nuvem utilizada está representada na figura 6.5.

O modelo escolhido para o primeiro ensaio consistiu na nuvem completa sintetizada pelo algoritmo "ply2pcd", após "remesh" da malha em "Blender". A opção pelo modelo completo

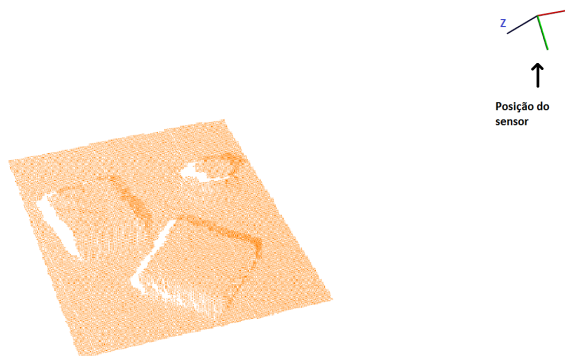


Figura 6.5: Nuvem de pontos da cena recolhida com o sensor, com um conjunto de objetos incluindo o que se pretende detetar(isolar).

foi tida em primeira escolha dado que se considerou ser aquela que maior informação descritiva possuía do objeto. Como podemos observar na figura 6.6, o ficheiro CAD foi convertido na sua totalidade.

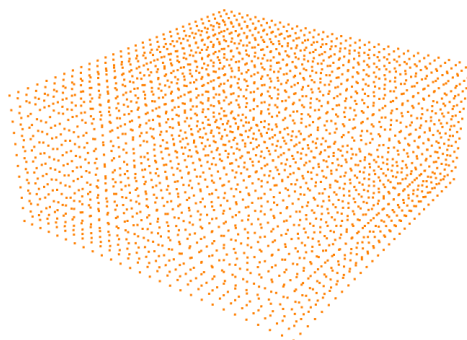


Figura 6.6: Nuvem de pontos do modelo utilizado no primeiro ensaio. É o modelo completo obtido pela função "ply2pcd".

O ensaio foi realizado utilizando a cena e o modelo referidos, no entanto, o algoritmo não gerou qualquer correspondência. Foram ajustados todos os parâmetros tais como: a amostragem do modelo e da cena, os raios de cálculo dos descritores, o valor limite de correspondência entre pares de descritores, tal como a parametrização para o clustering, sem qualquer melhoria nos resultados obtidos. O resultado obtido no final da sequência surgia sempre como representado na figura 6.7 ou, por vezes, ao atribuírem-se valores excessivamente abrangentes obtinham-se falsas correspondências sem coerência.

A despistagem da programação executada, nomeadamente a implementação dos algoritmos, foi crucial para a continuação do trabalho. O método utilizado consistiu na alteração do modelo. Em alternativa à nuvem sintetizada, considerou-se um modelo gerado pelo sensor à semelhança do trabalho desenvolvido em [36]. Deste modo, com o sensor recolheu-se uma nova nuvem de pontos, que podemos ver na figura 6.8, que serviu como cena. O modelo utilizado

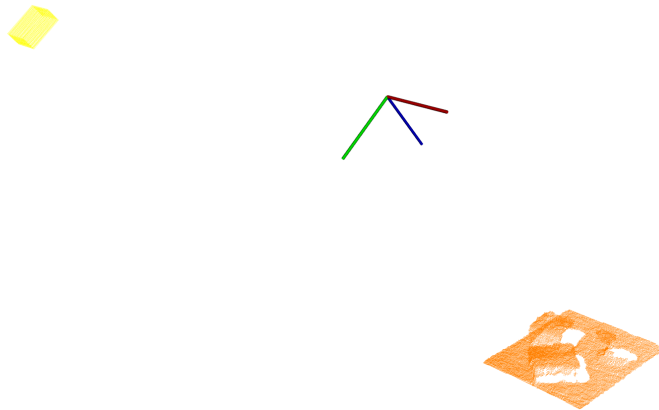


Figura 6.7: Nuvem de pontos do primeiro ensaio de correspondência.

consistiu na mesma nuvem, mas após a remoção do plano de suporte como observamos na figura 6.9, para que não restassem dúvidas na validade do modelo utilizado.

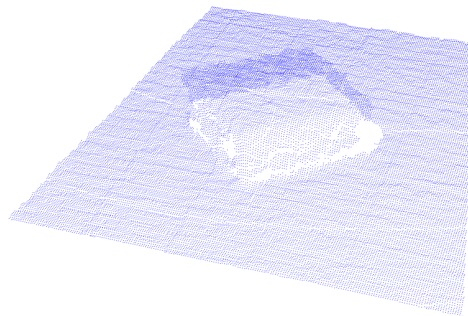


Figura 6.8: Nuvem de pontos da cena usada na análise da metodologia de correspondência.

O ensaio foi realizado e o correto funcionamento da implementação confirmada. Na figura 6.10, resultante da correspondência entre o modelo e a cena de teste, podemos constatar que ocorre uma identificação perfeita da posição do modelo na cena e entre pares de descritores. O resultado devolvido pelo sensor na forma da matriz R de rotação e T de translação vêm confirmar o que visualmente se supunha em relação ao alinhamento entre ambos.

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}; T = [0 \quad 0 \quad 0]$$

O teste descarta as suspeitas da existência de um erro na programação que pudesse inviabilizar o correto funcionamento da implementação. Sabendo que o modelo utilizado correspondia a uma secção da cena, a identificação era inevitável. Logo, o passo seguinte foi, utilizando o mesmo modelo, alterar a cena de detecção. A cena utilizada foi a anterior de maior com-

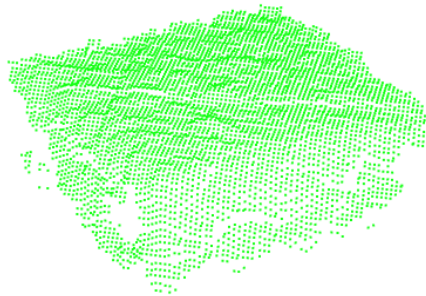


Figura 6.9: Nuvem de pontos do modelo usado na análise da metodologia de correspondência.



Figura 6.10: Nuvem de pontos resultante da correspondência.

plexidade, representada na figura 6.5 que continha outros objetos e cuja nuvem do objeto a detectar, era distinta da utilizada como modelo.

O novo ensaio foi realizado e, após alguns ajustamentos no algoritmo no valor mínimo de correspondência de descritores, obteve-o ilustrado na figura 6.11.

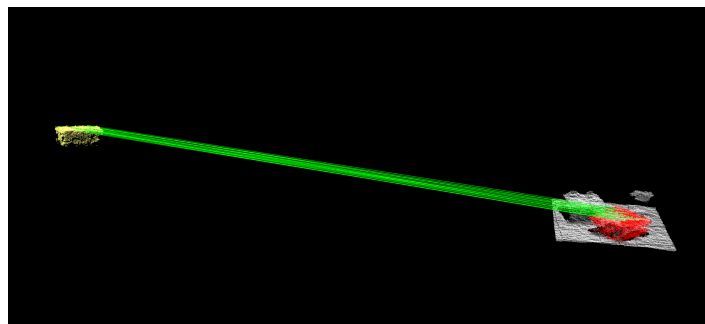


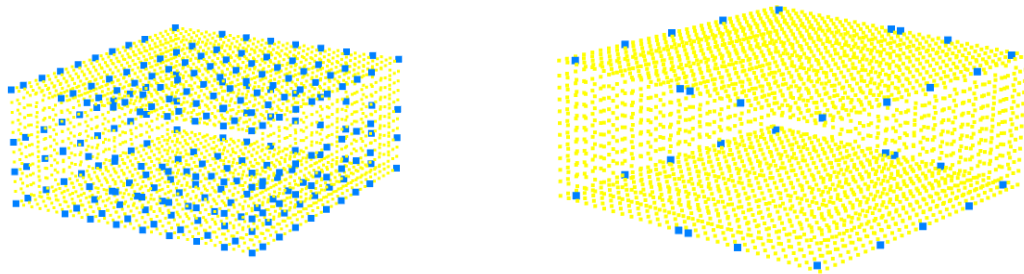
Figura 6.11: Nuvem de pontos resultante da correspondência usada para despistagem, utilizando a cena de maior complexidade.

Os resultados obtidos indicam que a fonte do problema não estava na programação dado que com a mudança de modelo a correspondência era executada, logo o problema era do modelo utilizado. A conversão pela função "ply2pcd", com o ajustamento no "Blender", era a única metodologia que se considerava correta e por isso foi a usada.

De forma a possibilitar a utilização do modelo, testou-se uma pequena modificação na

sequência de correspondência, ao invés da amostragem ser gerada pela classe `pcl::UniformSampling`, em que os pontos chave são selecionados pelo espaçamento entre si, utilizou-se a classe `pcl::ISSKeypoint3D` que se baseia no algoritmo Intrinsic Shape Signatures que seleciona os pontos chave pela sua riqueza descritiva [42].

Devido a sua tipologia de cálculos a classe `pcl::ISSKeypoint3D`, gera uma amostragem do modelo muito distinta da usada anteriormente. O exemplo disso está presente na figura 6.12 em que podemos observar em 6.12a a seleção por Uniform Sampling onde é gerada uma malha de espaçamento uniforme ao longo de todo o modelo, representada nos pontos azuis. O resultado do cálculo por Intrinsic Shape Signatures apenas seleciona os pontos do bordo, como representa a figura 6.12b, pois são os que apresentam maior variabilidade na nuvem.



(a) Amostragem por Uniform Sampling.

(b) Amostragem por Intrinsic Shape Signatures.

Figura 6.12: Nuvens de pontos das amostragens (pontos a azul) por diferentes algoritmos.

A nova amostragem foi utilizada na tentativa de execução de correspondência entre o modelo completo sintetizado e a nuvem recolhida. Contudo, os resultados não foram positivos não se tendo conseguido alcançar a identificação do objeto.

A execução deste tipo de seleção de pontos chave para cálculo dos descritores revelou vários detalhes que colocaram em causa o seu uso. Apesar de utilizada a mesma configuração, o modelo e a cena sofreram amostragens bastante distintas, como podemos analisar na figura 6.13. É certo que diferem na própria origem e esse detalhe influencia os resultados do algoritmo, no entanto, devido ao número de pontos da nuvem reamostrada, não se encontram correspondências com a cena utilizada.

6.2 Modelo Parcial

Os resultados não eram animadores e era necessário encontrar uma solução. A digitalização feita pelo sensor funcionava como modelo e desta forma foi tida como referência. Para aproximar o modelo sintetizado a uma digitalização do sensor considerou-se a utilização de modelos parciais.

O modelo utilizado até então consistia numa digitalização completa, resultado da metodologia utilizada para a conversão do ficheiro CAD em nuvem de pontos. Para obter um modelo parcial optou-se por remover pontos ao modelo completo. Dado que o objeto utilizado caracterizava-se por uma geometria simples, a parcialização baseou-se na remoção de



Figura 6.13: Nuvem de pontos resultante da correspondência usando amostragem por Intrinsic Shape Signatures.

algumas faces do paralelepípedo. A remoção das faces foi executada por filtragem dos pontos pertencentes aos planos a remover. O modelo parcial obtido está representado na figura 6.14.

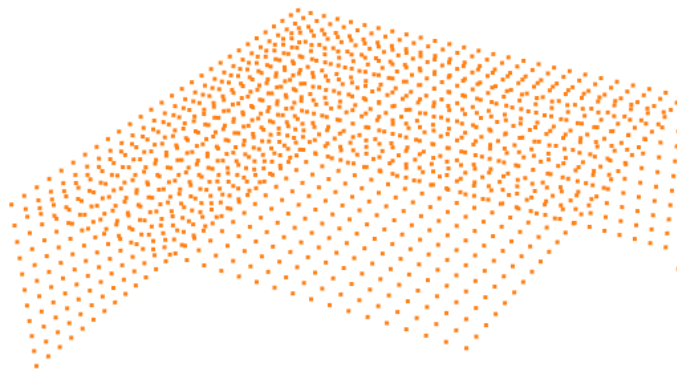


Figura 6.14: Nuvem de pontos do modelo parcial gerado por remoção de planos.

O modelo parcial permitiu um novo ensaio com resultados mais favoráveis. Como se pode ver na figura 6.15, foi possível identificar e localizar o objeto na cena utilizada anteriormente. A configuração do algoritmo não foi alterada em relação ao ensaio anterior, mantendo-se todos os parâmetros, tais como: distâncias de amostragem, vizinhanças de cálculo das normais e dos descritores, bem como o limiar de aceitação de correspondência entre descritores. A distinção constou apenas na utilização de um modelo parcial.

A análise mais detalhada do alinhamento entre as nuvens de pontos revela algumas imperfeições. Ao observarmos de um ponto de vista mais próximo, na figura 6.16 podemos constatar que a transformação calculada não gera um alinhamento perfeito entre o modelo e a seção da cena pertencente ao objeto. O fator que impôs esse desalinhamento foi a ocorrência de todas as correspondências na face superior do objeto, ficando este ligeiramente rodado na normal ao plano.

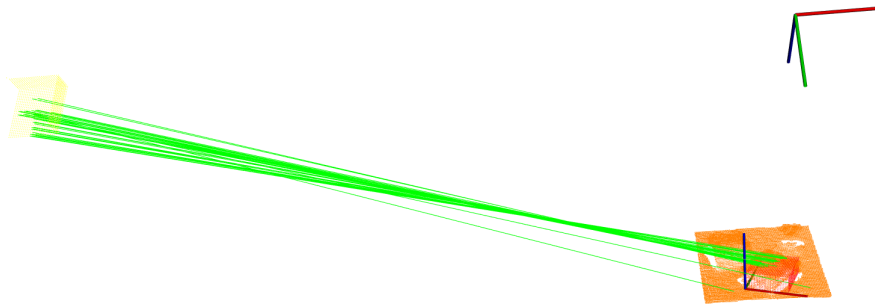


Figura 6.15: Ensaio de correspondência com modelo parcial.

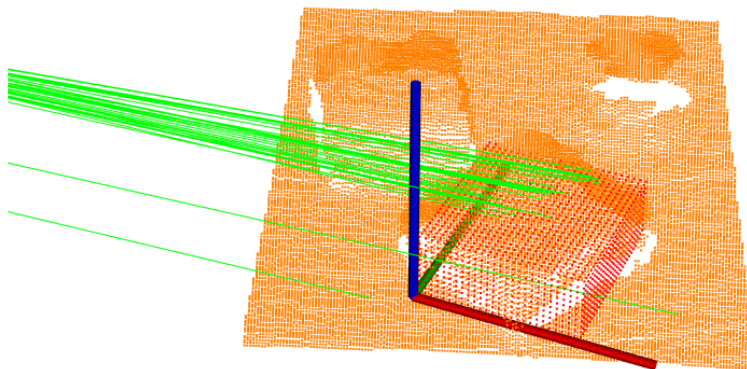


Figura 6.16: Vista aproximada ensaio de correspondência com modelo parcial.

Os ensaios realizados levam a que se tirem algumas conclusões sobre as características necessárias na nuvem de pontos do modelo, a utilização de uma nuvem de pontos parcial surge como indispensável. O método anteriormente utilizado para a criação do modelo parcial foi uma solução provisória, que funcionou para aquela perspectiva em particular e geometria simplista do objeto. No entanto foi necessário procurar uma alternativa mais versátil.

A pesquisa por uma alternativa para a sintetização da nuvem de pontos modelo revelou um algoritmo da PCL que se adequava ao pretendido, o VirtualScanner. Após algumas alterações para corresponder aos requisitos da implementação, este algoritmo tornou-se na metodologia utilizada para construção de modelos no decorrer do resto do trabalho. A descrição do algoritmo, e das respetivas alterações implementadas, está contida na secção 4.3.

A determinação de um modelo sintetizado que funcionava no reconhecimento do objeto suscitou o interesse por adaptações e novos ensaios da implementação. Baseada na implementação até então utilizada, criou-se uma variante do algoritmo recorrendo ao descritor FPFH.

Este desenvolvimento foi impulsionado pela oportunidade de comparação de dois descritores da PCL, tendo deste modo também uma alternativa para a implementação desenvolvida.

A correspondência recorrendo ao descritor FPFH não correu como esperado. Como podemos observar na figura 6.17, o desempenho não foi equivalente ao conseguido pelo descritor anteriormente utilizado. Após alguns ajustamentos ao valor de distância mínima entre descritores, a identificação da peça foi conseguida, no entanto o cálculo da posição do objeto não foi satisfatório. O desfasamento na orientação determinada deve-se ao reduzido número de correspondências entre descritores, contudo ao baixarmos o limiar de admissão as novas correspondências geradas não melhoram a orientação determinada.

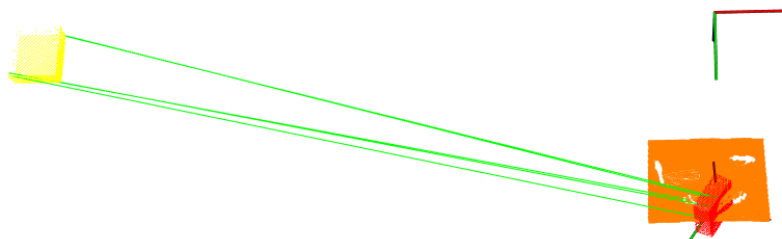


Figura 6.17: Ensaio de correspondência com modelo parcial e descritor FPFH.

O ponto de situação da implementação, e as conclusões retiradas, levaram à definição dos passos seguintes no desenvolvimento. Nesta etapa foram aplicadas várias mudanças.

O objeto utilizado até então, o paralelepípedo, como já foi mencionado, possui uma geometria muito simples, contendo muitas simetrias e pouca riqueza descritiva. Deste modo foi modelada e construída uma nova peça que não apresenta as limitações da anterior. A sua descrição é feita mais detalhadamente em 3.1.2.

A solução desenvolvida baseia-se na utilização de modelos parciais e estes são dependentes da perspetiva, como também é necessário um conjunto de modelos deste tipo e não apenas um para descrever o objeto na sua totalidade. O algoritmo desenvolvido estava preparado para a utilização de apenas um modelo. Para dar continuidade a estratégia delineada foi desenvolvido o sistema multi-vista.

O sistema multi-vista baseia-se no sistema de um único modelo integrado num procedimento para o encapsulamento da informação de cada modelo utilizado, de acordo com o algoritmo 2.

Define um vetor com a lista de vistas;

for *cada modelo* **do**

 Calcula as possíveis deteções e respetivas posições;

 Armazena as transformações e número de correspondências entre descritores;

end

Algorithm 2: Sequência de encapsulamento de informação, para acomodar a utilização de várias vistas do objeto.

O encapsulamento é estruturado do seguinte modo: é definido um vetor com a dimensão

do número de modelos utilizados. Em cada elemento deste vetor está contida uma estrutura com dois vetores, o primeiro vetor contém o conjunto de matrizes de transformação para cada detecção referente ao modelo utilizado, e o segundo vetor os números de correspondências entre descritores para cada identificação.

O conjunto de alterações executadas foram: uma nova peça, a mudança de posição do sensor e a utilização do algoritmo para várias vistas, após a implementação das mesmas foram realizados alguns ensaios numa cena de vários objetos em múltiplas posições.

Os ensaios realizados não resultaram como esperado. Utilizando todas as modificações o comportamento do algoritmo alternava entre a não detecção de qualquer objeto e um conjunto amorfo de falsas detecções, como o exemplo da figura 6.18.

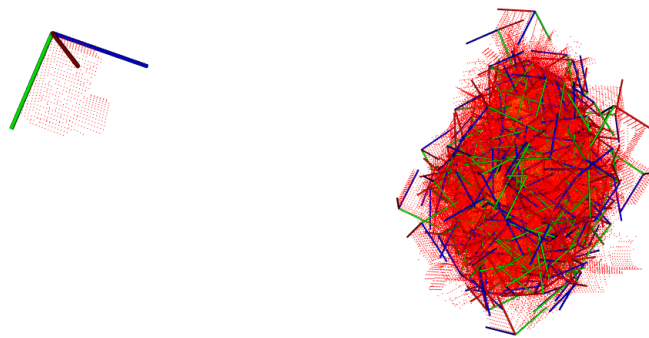


Figura 6.18: Ensaio de correspondência com múltiplos modelos das várias vistas parciais do objeto de referência.

O resultado obtido levou à necessidade de reavaliar o procedimento. O primeiro passo consistiu no uso do algoritmo para múltiplos modelos, mas apenas utilizando um. O ensaio teve por objetivo a despistagem de possíveis problemas causados pela alteração do algoritmo. Os resultados revelaram que a não correspondência não era provocada pela utilização de vários modelos. Como ilustra a figura 6.19, o resultado permaneceu para o uso de apenas um modelo.



Figura 6.19: Ensaio de correspondência para análise do problema na utilização das várias vistas.

A programação do algoritmo foi revista várias vezes sem sucesso na detecção da falha, o

que levou a um ensaio com a mesma cena e modelo no algoritmo desenvolvido para apenas uma vista, usado anteriormente. O resultado obtido foi inesperado dado que também não era concretizada qualquer correspondência, a conclusão que podemos retirar foi de que o problema não estaria na programação.

As conclusões retiradas levaram a centrar as atenções na cena e modelo utilizados, o modelo havia sido sintetizado do mesmo modo que anteriormente mas, no entanto o objeto não era o mesmo. De forma a excluir qualquer dúvida, a nuvem de pontos do modelo foi analisada com maior detalhe, como podemos ver na figura 6.20, não apresenta qualquer problema. A nuvem descrevia o objeto com grande detalhe e simulando a oclusão da vista parcial.

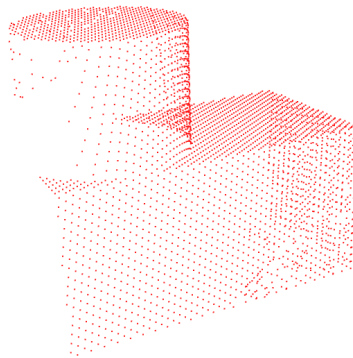


Figura 6.20: Nuvem de pontos modelo.

6.3 Cena, Modelo e Ruído

Decidiu-se analisar a nuvem recolhida, e aí detetou-se que a sua qualidade se havia degradado bastante. A fixação do sensor no suporte causou alterações que não tinham sido detetadas. A nova localização do sensor implicou um aumento da distância aos objetos e conseqüentemente a redução da resolução da nuvem recolhida.

A distância do objeto ao sensor medida pelo equipamento era de 840 milímetros, para comparar as diferenças de qualidade foi recolhida uma nuvem de pontos a 660 milímetros, sendo esta distância aproximada ao valor a que se havia operado anteriormente. A figura 6.21 ilustra as duas nuvens, onde podemos comparar a qualidade de ambas, e até que ponto essa característica afetou o algoritmo. A figura 6.21a revela uma nuvem com espaçamento entre pontos superior, a face lateral plana é descrita geometricamente com maior rigor em 6.21b, e, na figura 6.21a, o ruído é mais pronunciado.

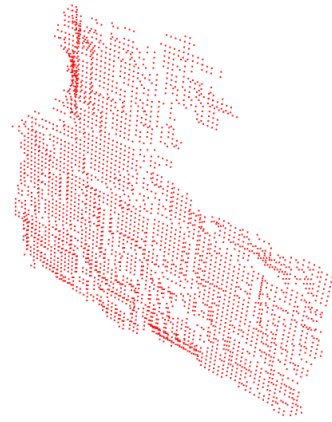
A manipulação das nuvens revelou que, na configuração ilustrada na figura 6.22, a qualidade descritiva de cada nuvem era bastante distinta.

A figura 6.23 mostra uma vista de topo na qual a descrição do tipo de faces do objeto está bem patente: na nuvem 6.23a é impercetível a existência de duas geometrias dado que a resolução é insuficiente, e por outro lado, na nuvem recolhida a maior proximidade, em 6.23b, o detalhe da superfície plana é visível e uniforme. A destacar também a menor variação dos pontos em relação à face onde estão inseridos.

As diferenças nas nuvens recolhidas em função da distância ao sensor, são a confirmação de que o funcionamento da implementação é suportado pelo uso do sensor o mais próximo possível



(a) Nuvem recolhida a 890 milímetros.



(b) Nuvem recolhida a 660 milímetros.

Figura 6.21: Nuvens de pontos recolhidas a diferentes distâncias para comparação.

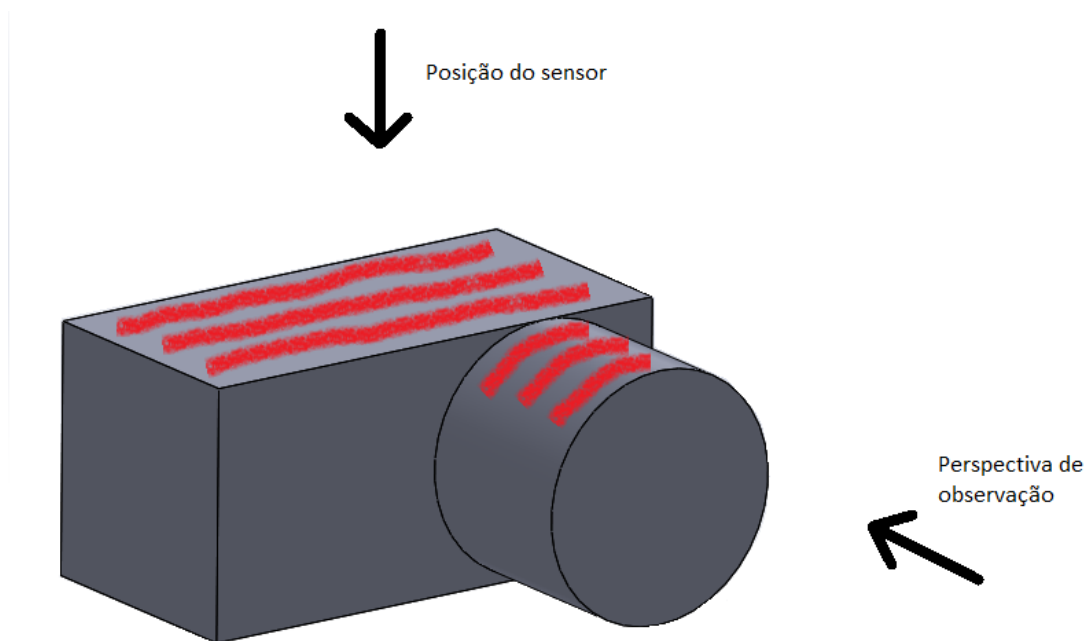


Figura 6.22: Esquema de ilustração da secção representada pelas nuvens de pontos recolhidas.

dos objetos. A condicionante da distância é resultado das limitações do sensor utilizado. Desse modo, os restantes ensaios e desenvolvimentos terão de o ter em conta.

No processo de correção da implementação para várias vistas parciais, e após a análise das nuvens utilizadas, mostrou-se oportuno continuar a análise aos restantes passos da metodologia. A sequência de operações foi revista em cada etapa após o carregamento de cada nuvem: na fase inicial, a primeira ação consistia no cálculo das normais, seguida da amos-



(a) Nuvem recolhida a 890 milímetros.

(b) Nuvem recolhida a 660 milímetros.

Figura 6.23: Nuvens de pontos recolhidas a diferentes distâncias para comparação.

tragem. A sequência de operações não está errada, mas dado que o tipo de amostragem que é feita baseia-se na redução uniforme da densidade de pontos pela distância entre eles, esta não requer as normais. Este detalhe revela que a sequência: cálculo das normais e posterior amostragem pode ser invertida. O interesse no cálculo das normais após a amostragem deve-se às conclusões retiradas da influência da qualidade das nuvens recolhidas. Dada a importância da densidade de pontos e do ruído, ao calcular as normais apenas dos pontos chave, a operação de amostragem funciona também como filtro, melhorando a qualidade das normais calculadas. No entanto esta estratégia implica que a amostragem não reduza o número de pontos em excesso, de forma a que exista uma vizinhança de pontos suficiente que viabilize o cálculo das normais.

A nova sequência caracteriza-se agora, como ilustrado na figura 6.24, pela execução da amostragem seguida do cálculo das normais dos pontos selecionados. Esta abordagem leva a que o grau de semelhança entre a nuvem modelo e a cena seja superior, dado que a amostragem é idêntica nas duas nuvens e consequentemente as normais obtidas.

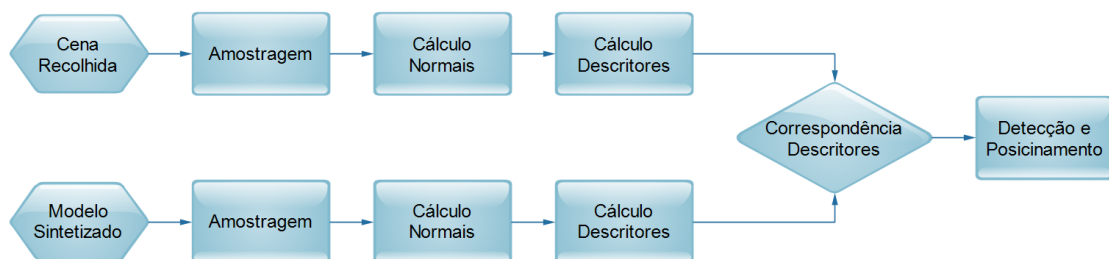


Figura 6.24: Sequencia adaptada de operações para correspondência.

O passo seguinte que se impunha era na parte dos descritores que já haviam sido modificados anteriormente, mas desta vez seguiu-se uma abordagem diferente. De forma a perceber os valores assumidos pelos descritores, quer do modelo quer da cena recolhida, nas diferentes zonas do objeto fez-se uma análise detalhada dos mesmos. A análise dos descritores foi implementada do seguinte modo: na janela de visualização, os pontos chave da nuvem estavam etiquetados com um número, como se pode observar na figura 6.25. A cada etiqueta correspondia um descritor, que podia ser visto na janela de comando pois a listagem era lá exposta.

O ensaio não foi executado recorrendo ao descritor SHOT mas sim ao FPFH. A causa da alteração foi a dimensão dos descritores, enquanto que o SHOT usa vetores de 352 elementos para o armazenamento dos descritores o FPFH necessita apenas de 33, o que facilita a análise visual.

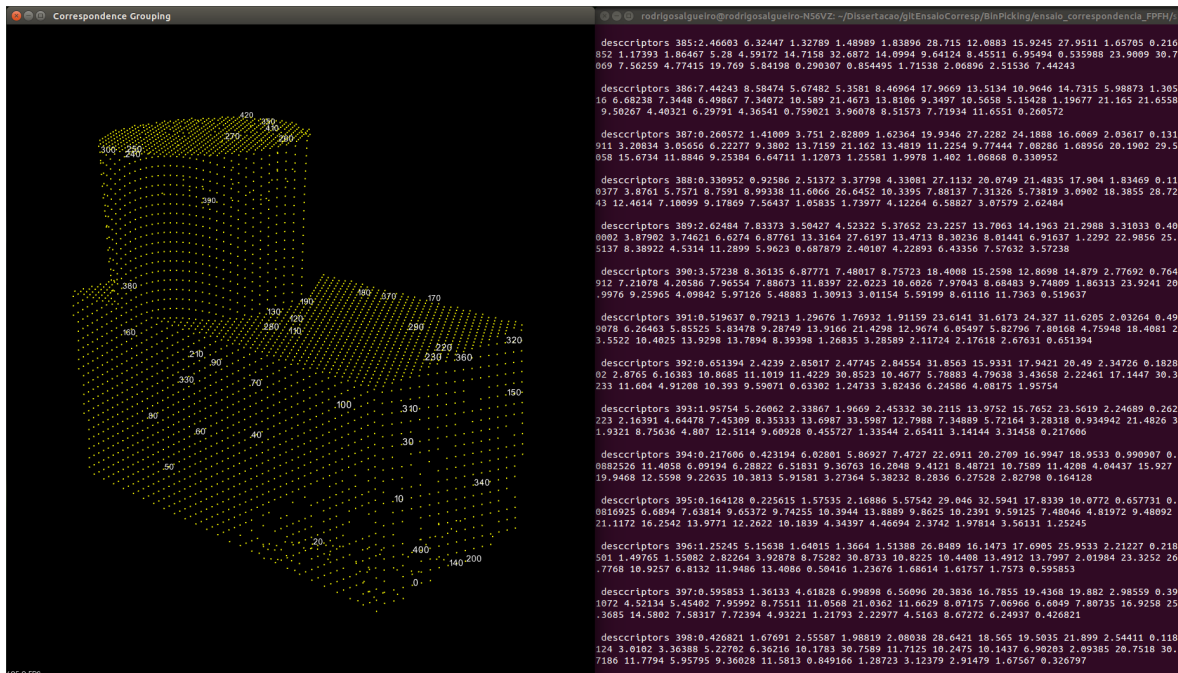


Figura 6.25: Exemplo do ensaio dos descritores.

O resultado da análise realizada revelou um detalhe interessante, a diferença entre os descritores calculados no modelo e na cena para uma localização aproximada está focada na variabilidade dos descritores calculados em cada ponto chave. A verificação dos vetores mostrou que para os descritores calculados no modelo os valores encontravam-se mais extremados, com alguns valores bem destacados e outros muitas vezes residuais e próximos de zero. Nos vetores da nuvem recolhida, pelo contrário, as variações não são tão acentuadas, apesar de perceptíveis.

O sucesso de correspondência entre modelo e cena reside na semelhança entre ambos. Fizeram-se então novos ensaios baseados numa pequena alteração aos modelos: a introdução de ruído. O ruído consistiu numa pequena variação induzida nas coordenadas de cada ponto, simulando o ruído inerente às nuvens recolhidas pelo sensor e, com isso, aproximar os descritores calculados em ambas. A metodologia para introdução de ruído encontra-se descrita na secção 4.3, tendo como resultado a nuvem que surge na figura 6.26.

Os resultados obtidos vieram demover algumas ideias e metodologias que se acreditava certas no início do trabalho: a introdução de algum ruído no modelo utilizado facilita o funcionamento da correspondência e do mesmo modo a identificação dos objetos. A figura 6.27 ilustra o resultado da correspondência recorrendo a um modelo com ruído de 2 milímetros, onde se recorreu ao descritor FPFH.

O alinhamento entre o modelo e a cena recolhida não foi perfeito. No entanto, foi o melhor desempenho conseguido para o novo objeto. No ensaio recorreu-se ao descritor FPFH e esse não era o que melhor desempenho havia mostrado até então. Dada essa questão realizou-se

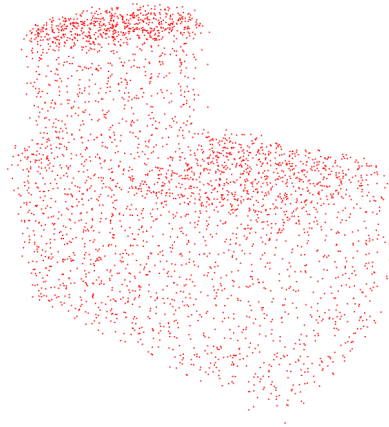


Figura 6.26: Modelo resultante da aplicação de ruído com variação de 2 milímetros.

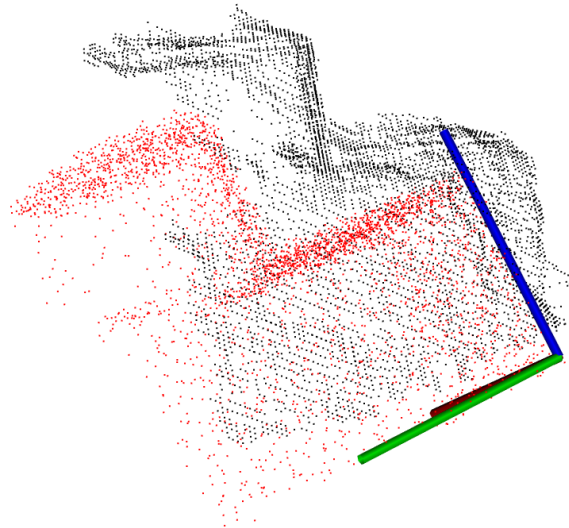


Figura 6.27: Correspondência utilizando modelo com ruído de 2 milímetros e descritor FPFH.

um novo ensaio mas utilizando o descritor SHOT. O resultado está representado na figura 6.28.

As conclusões que se tiram ao olhar para o alinhamento entre as duas nuvens evidenciam que este é o melhor resultado obtido. O cálculo da posição e orientação do objeto melhorou, o que vem confirmar a importância da introdução de ruído nos modelos, obtendo com isso resultados de melhor qualidade.

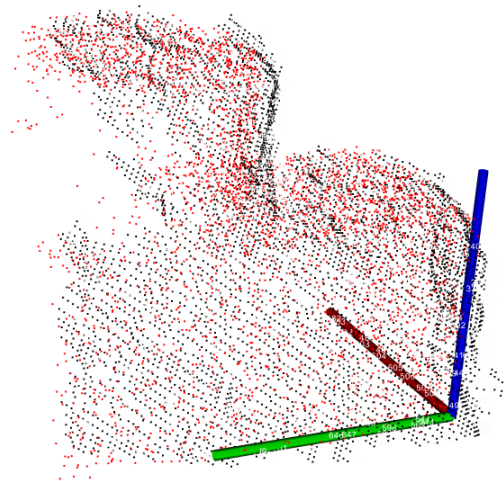


Figura 6.28: Correspondência utilizando modelo com ruído de 2 milímetros e descritor SHOT.

Capítulo 7

Testes e Resultados

A análise de correspondência entre o modelo e a cena revelou que a introdução de ruído no modelo representava um fator importante na qualidade dos resultados de correspondência obtidos. O comportamento observado com a utilização de ruído, levou a realização do estudo da influência do ruído no desempenho da identificação e localização dos objetos.

A fundamentação do estudo carecia de uma métrica que permitisse avaliar a dependência do ruído. Após alguma análise estabeleceu-se que, o valor do quociente do número de correspondências entre descritores(**NC**), pelo número de pontos chave do modelo(**NPC**), seria a melhor métrica para aferir os resultados obtidos. A equação está descrita na expressão 7.1.

$$\text{Porcentagem de Corrêspndencia(PC)} = \frac{\text{NC}}{\text{NPC}} \cdot 100 \quad (7.1)$$

O conjunto de ensaios realizados baseou-se nos dois descritores utilizados, o SHOT e o FPFH. O estudo foi realizado com níveis crescentes de ruído, começando no uso do modelo não modificado até ao valor de 10 milímetros de variação.

7.1 Descritor SHOT

A parametrização do algoritmo foi igual para todos os ensaios, sendo apenas utilizados diferentes modelos. Os valores utilizados foram respetivamente: 3 milímetros para a amostragem dos pontos chave, 10 milímetros para a vizinhança de cálculo das normais e 40 milímetros para a esfera de abrangência dos descritores.

A análise do comportamento do descritor SHOT iniciou-se com a correspondência usando um modelo sem ruído. A figura 7.1 ilustra o resultado do ensaio. As conclusões que podemos retirar visualmente são de uma identificação correta mas com uma localização um pouco grosseira. O valor de percentagem de correspondência foi de 92,6 o que deveria gerar uma deteção com melhor resultado.

O modelo que se seguiu no ensaio foi o de ruído de 1 milímetro. O resultado está ilustrado na figura 7.2, como podemos observar o alinhamento em termos visuais melhorou. Após o cálculo da percentagem de correspondência obteve-se o valor de 57,7 o que contraria a análise visual. Dada a boa localização do objeto era expectável uma percentagem superior.

O ensaio seguinte, referente ao descritor SHOT, foi realizado com um modelo de 2 milímetros de ruído, a figura 7.3 retrata o referido teste. A observação do alinhamento das nuvens

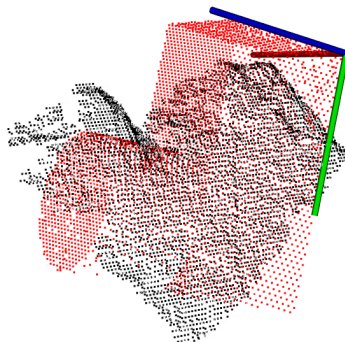


Figura 7.1: Ensaio de correspondência com o descritor SHOT e modelo sem ruído.

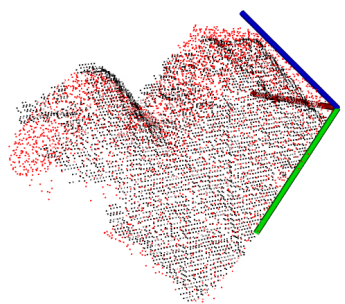


Figura 7.2: Ensaio de correspondência com o descritor SHOT e modelo com 1 milímetro de ruído.

não permite afirmar que o resultado seja melhor ou pior do que o anterior dada a semelhança de ambos. Segundo a métrica utilizada a percentagem de correspondência foi de 54,2 valor bastante próximo do obtido com o modelo de ruído de 1 milímetro.

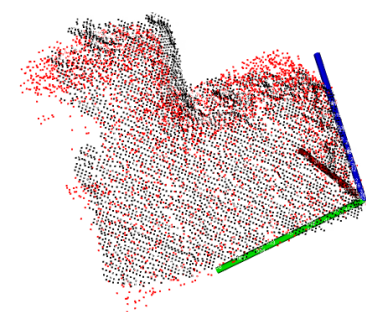


Figura 7.3: Ensaio de correspondência com o descritor SHOT e modelo com 2 milímetros de ruído.

A realização de ensaios foi continuada com intervalos de 1 milímetro, até ao máximo de 10

milímetros. A recolha de todos os valores de percentagem de correspondência permitiu traçar um gráfico para uma análise comparativa. A figura 7.4 expõe o referido gráfico. Os limites são definidos pelo máximo de 98,6 para o modelo sem ruído, e o mínimo de 28,8 para o modelo com ruído de 10 milímetros.

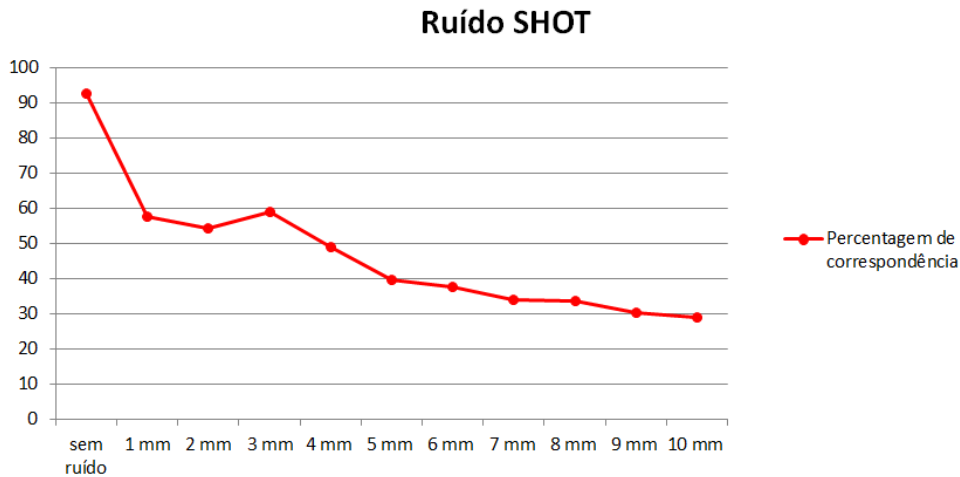


Figura 7.4: Gráfico da percentagem de correspondência em função do ruído imposto ao modelo, para o descritor SHOT.

A tendência observada descreve uma diminuição abrupta da percentagem de correspondência obtida no modelo sem ruído para o de 1 milímetro, mantendo-se mais ligeiramente para o modelo de 2 milímetros. O ensaio com o modelo de ruído 3 milímetros, mostrou um ligeiro aumento da percentagem de correspondência, em relação ao anterior, com um valor de 59,0. A análise visual da correspondência conseguida nesse ensaio, revelou que o alinhamento não era melhor do que o anterior. A figura 7.5, ilustra isso mesmo, em que o ensaio com o modelo de 3 milímetros não alcança um alinhamento perfeito entre nuvens.

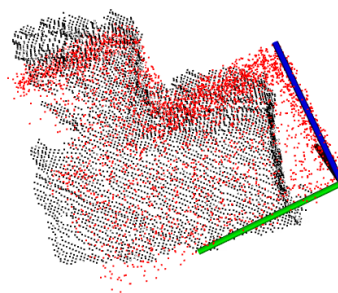
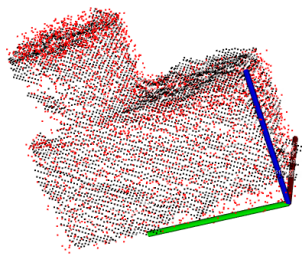
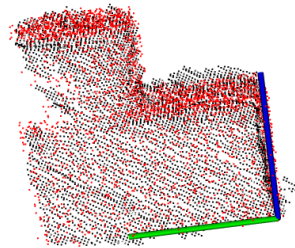


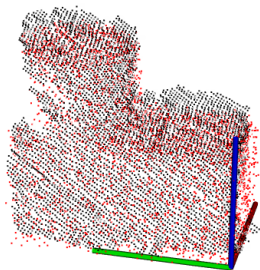
Figura 7.5: Ensaio de correspondência com o descritor SHOT e modelo com 3 milímetros de ruído.



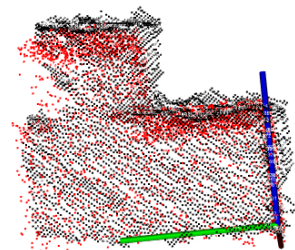
(a) Resultado de correspondência da nuvem de pontos A.



(b) Resultado de correspondência da nuvem de pontos B.



(c) Resultado de correspondência da nuvem de pontos C.



(d) Resultado de correspondência da nuvem de pontos D.

Figura 7.6: Resultados obtidos com vários modelos de ruído 2 milímetros, para o descritor SHOT.

Os ensaios seguintes foram resultando numa gradual redução do valor de percentagem de correspondência. Ao observarmos esta tendência podemos dizer que vai de encontro ao esperado. A um aumento de ruído corresponde uma diminuição da percentagem de correspondência. No entanto ao confrontarmos com a análise visual possível, constatamos que não conduzem as mesmas conclusões. Nos ensaios iniciais com a ligeira introdução de ruído nos modelos, os alinhamentos das nuvens de pontos recolhida e modelo melhoraram. Contudo, com valores de ruído superiores a 3 milímetros os alinhamentos tornaram-se mais instáveis, não apresentando uma melhoria contínua com a gradual introdução de ruído. O conjunto das figuras respeitantes a todos os ensaios estão presentes no apêndice A.

No seguimento dos testes de correspondência realizados com o descritor SHOT, foi realizado um ensaio de sensibilidade ao ruído. O ensaio consistiu na geração de vários modelos com o mesmo nível de ruído, e na posterior execução de correspondência com a mesma cena recolhida. Devido ao facto do ruído introduzido ser aleatório, os modelos obtidos não possuem nuvens de pontos iguais. O valor de ruído seleccionado foi o de 2 milímetros, dado ter apresentado anteriormente o melhor alinhamento para os valores mais elevados de percentagem de correspondência. As correspondências estão ilustradas na figura 7.6.

A análise visual das correspondências revela que os alinhamentos tiveram pouca variação

de modelo para modelo. No entanto é perceptível que ocorrem pequenas mudanças na posição calculada.

Recorrendo a métrica anterior, de percentagem de correspondência, foi desenhado o gráfico presente na figura 7.7.

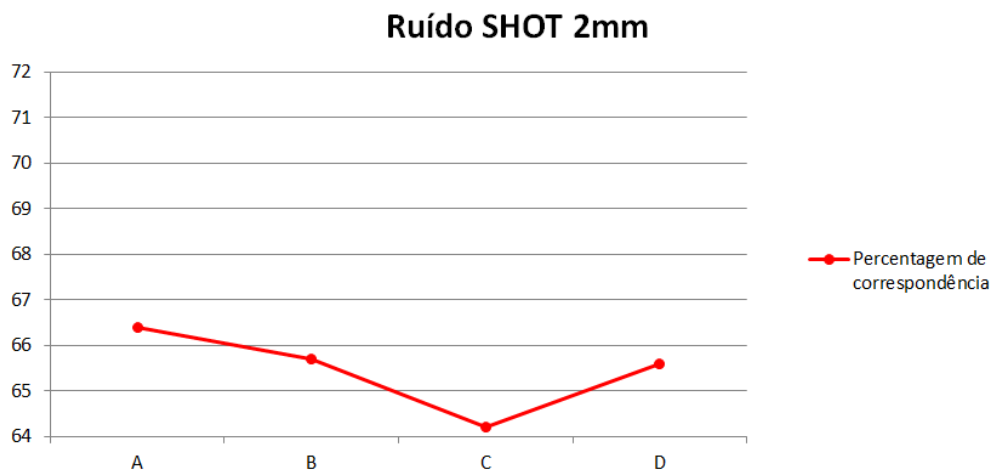


Figura 7.7: Gráfico dos valores de percentagem de correspondência para o ensaio de sensibilidade ao ruído, com o descritor SHOT.

Neste podemos observar que existe uma pequena variação dos valores, entre o máximo de 66,4 e o mínimo de 64,2. A variação de 2,2 não é muito significativa. No entanto de salientar que os modelos foram obtidos pelo mesmo modo, com a mesma amplitude de ruído. O que demonstra que o descritor SHOT é sensível ao ruído. E pequenas alterações na posição dos pontos afetam a correspondência de descritores. Na análise estatística o valor médio foi de 65,6 e o desvio padrão 0,798.

7.2 Descritor FPFH

Os ensaios com o descritor FPFH seguiram a mesma metodologia que os realizados com o descritor SHOT. O primeiro modelo foi o sem ruído, obtido diretamente da conversão do modelo CAD. O resultado obtido ilustra na figura 7.8, como podemos observar apesar de o objeto ter sido identificado, a sua localização é incorreta. Ao calcular a percentagem de correspondência obteve-se um valor de 98,9 o que não condiz com o resultado de análise visual.

O uso do modelo com 1 milímetro de ruído revelou melhorias no alinhamento das nuvens, como ilustra a figura 7.9. Apesar de o alinhamento obtido não ser perfeito é melhor que o anteriormente obtido com o modelo sem ruído. O resultado da percentagem de correspondência foi de 77,9 valor inferior ao obtido anteriormente com o outro modelo.

O ensaio seguinte, realizado com o modelo de 2 milímetros, revelou um resultado visivelmente pior que o obtido anteriormente, a figura 7.10 revela o mesmo. O alinhamento entre o modelo e a cena não foi satisfatório, para a localização do objeto. O resultado da percentagem de correspondência não sofreu grande variação em relação ao anterior tendo sido de 71,2.

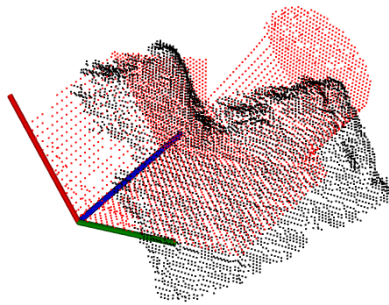


Figura 7.8: Ensaio de correspondência com o descritor FPFH e modelo sem ruído.

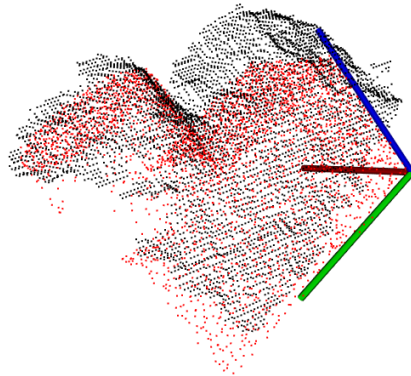


Figura 7.9: Ensaio de correspondência com o descritor FPFH e modelo com 1 milímetro de ruído.

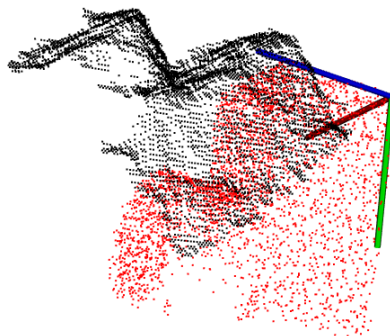


Figura 7.10: Ensaio de correspondência com o descritor FPFH e modelo com 2 milímetros de ruído.

Os testes tiveram continuidade até ao valor máximo de 10 milímetros de ruído, com um intervalo de ruído de 1 milímetro. A análise comparativa dos resultados foi executada por via do gráfico que figura em 7.11. Os limites do gráfico são definidos entre o máximo de 98,9 obtido com o modelo sem ruído e o mínimo de 38,4 gerado pelo modelo com 10 milímetros de ruído. Os resultados de percentagem de correspondência obtidos, têm um comportamento semelhante aos do descritor SHOT. Atinge-se o valor máximo sem o uso de ruído, e com a introdução do mesmo a percentagem de correspondência baixa. A análise visual do alinhamento, apesar de subjetiva, não retrata a mesma tendência.

O que se observa nas nuvens de pontos analisadas, é que o melhor resultado se obtém com a introdução de ruído de um 1 milímetro. Dado que o alinhamento melhora em relação ao modelo sem ruído. O aumento do valor de ruído, para 2 milímetros, leva ao retrocesso na precisão da localização do objeto. Embora esta observação não seja suportada pelos resultados de percentagem de correspondência. Nos restantes testes aos alinhamentos são instáveis, com estes a alterarem bastante a cada teste. De salientar também que não alcançando um resultado equivalente ao obtido com o modelo de 1 milímetro de ruído. Os resultados dos alinhamentos obtidos para cada modelo, estão representados no apêndice B.

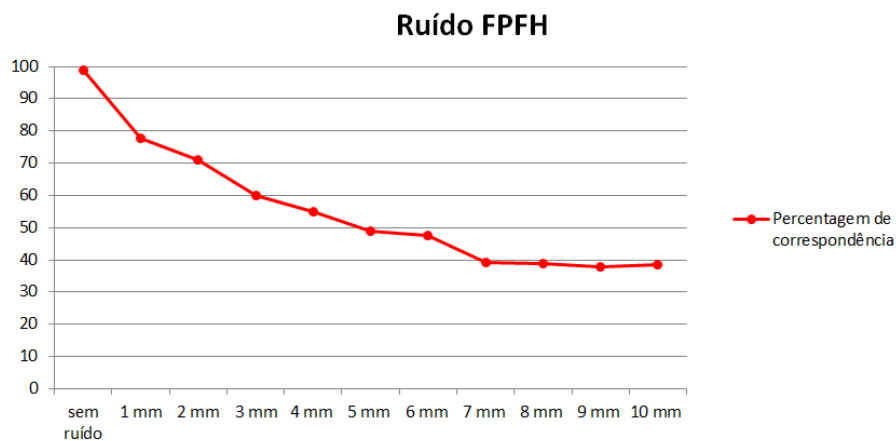
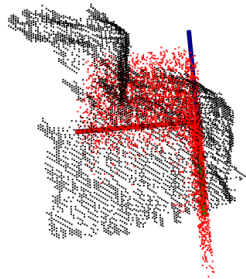


Figura 7.11: Gráfico da percentagem de correspondência em função do ruído imposto ao modelo.

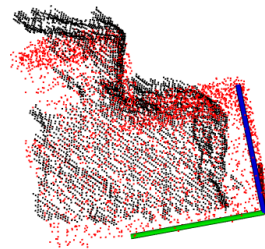
A sensibilidade do descritor FPFH ao ruído, foi testada pelo mesmo método que o descritor SHOT. Foram gerados quatro modelos com o mesmo nível de ruído, com 2 milímetros, e de seguida utilizados consecutivamente num teste de correspondência com a mesma cena. Os resultados dos alinhamentos obtidos estão ilustrados na figura 7.12.

As figuras ilustram uma forte influência do ruído nos resultados obtidos com o descritor FPFH. Apesar do nível de ruído ser idêntico para todos os modelos, os alinhamentos obtidos variam bastante a cada ensaio. Este comportamento do descritor, é bastante distinto do obtido com o descritor SHOT, que não demonstrou alterações tão prenunciadas de modelo para modelo.

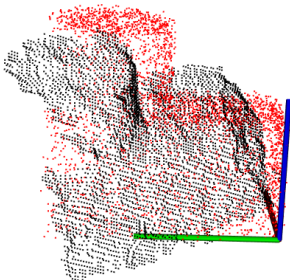
Os valores de percentagem de correspondência foram calculados, permitindo a construção do gráfico da figura 7.13. Os valores máximo e mínimo, de percentagem de correspondência, estão definidos respetivamente em 71,5 e 69,0. A amplitude de 2,5 é semelhante a obtida com o descritor SHOT, logo não justifica a variabilidade de alinhamentos obtidos para o mesmo ruído.



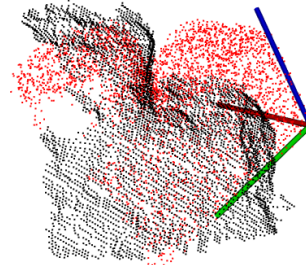
(a) Resultado de correspondência da nuvem de pontos A.



(b) Resultado de correspondência da nuvem de pontos B.



(c) Resultado de correspondência da nuvem de pontos C.



(d) Resultado de correspondência da nuvem de pontos D.

Figura 7.12: Resultados obtidos com vários modelos de ruído 2 milímetros, para o descritor FPFH.

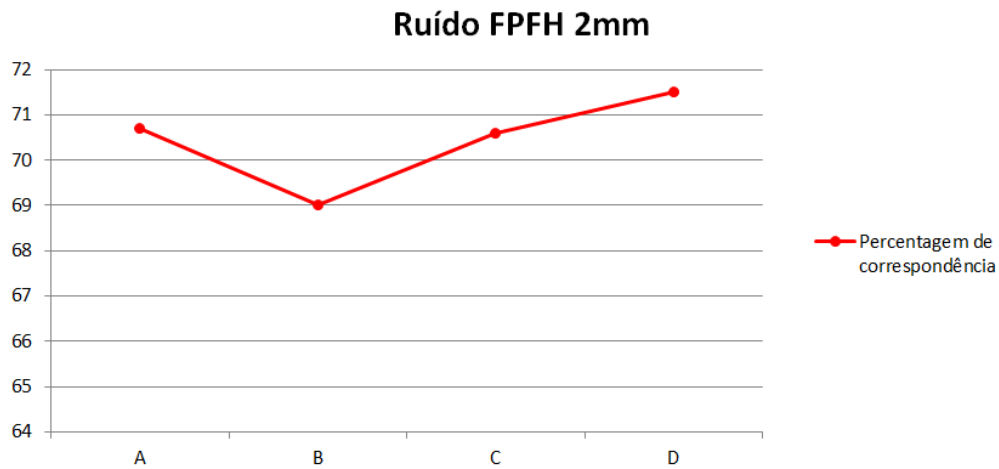


Figura 7.13: Gráfico dos valores de percentagem de correspondência para o ensaio de sensibilidade ao ruído, com o descritor FPFH.

Os resultados obtidos em ambos os descritores fundamentam, pela análise do alinhamento das nuvens, que a introdução de ruído nos modelos tem influência na correspondência entre cena e modelo. Contudo a métrica utilizada não justificou tal afirmação. Esta descreveu uma diminuição da percentagem de correspondência com o aumento do ruído, o que contraria os resultados de correspondência observados. O facto do valor máximo de percentagem de correspondência, para ambos os descritores, ser obtido com o modelo sem ruído não nos permite validar por esta métrica que o melhor alinhamento é obtido com penas perturbações de ruído nos modelos utilizados.

Capítulo 8

Conclusões e Trabalho Futuro

O trabalho descrito neste documento foi um desafio ambicioso desde a definição dos seus objetivos. Contudo veio dar continuidade a uma linha de investigação do Laboratório de Automação e Robótica, o Bin-Picking, com uma nova metodologia. A nova abordagem baseou-se na utilização de modelos CAD 3D dos objetos como referência para a sua identificação e localização, estratégia que aumenta a versatilidade do sistema.

Os desenvolvimentos conseguidos permitiram o estabelecer de uma metodologia que suporta os objetivos iniciais. O uso da referência do modelo CAD, todo o processamento associado a informação recolhida pelo sensor, e a correspondência de ambos culminando na localização do objeto.

A estrutura experimental desenvolvida possibilitou a realização do trabalho com sucesso, superando algumas falhas que haviam sido detetadas. Este conjunto equipamentos permanecerá no laboratório, e servirá de apoio a futuras atividades e trabalhos que possam beneficiar com a sua utilização.

O procedimento para a conversão do modelo CAD em nuvem de pontos foi estabelecido e conseguido. As nuvens de pontos obtidas têm rigor dimensional e são adaptáveis as solicitações necessárias, exemplo disso é a possibilidade de gerar vista parciais e inclusão de ruído. A apontar ficou a necessidade de utilizar uma primeira conversão com um software e posteriormente utilizar a programação, esta é uma oportunidade de melhoria futura.

A recolha e tratamento de informação fornecida pelo sensor foi realizada com sucesso, tendo-se atingido os objetivos. O processo de deteção do contentor foi bem estruturado, permitindo a localização do mesmo em múltiplas posições. O isolamento da informação é também conseguido, separando os pontos dos objetos dos pontos do contentor. Esta ação divide a informação de correspondência da informação de condicionamento de trajetória.

A sequência de correspondência foi implementada e mostrou-se funcional. Alguns dos algoritmos que a constituem foram revistos e alterados, como também a estrutura utilizada, contudo após a evolução os foram satisfatórios. Os descritores SHOT e FPFH possibilitaram a execução da localização dos objetos no entanto mostram muitas fragilidades. A pequenas alterações de modelo ou cena geravam a não deteção, o que demonstrava instabilidade do sistema.

O uso de ruído não teve uma fundamentação válida, de forma analisar a sua influencia utilizou-se uma métrica contudo com resultados inconclusivos. A métrica utilizada, percentagem de correspondência, não demonstrou uma tendência concordante com a análise feita visualmente e como tal não suportou uma justificação. O argumento que pode ser utilizado

para o efeito do ruído é a necessidade de variabilidade, para a operacionalidade dos descritores, contudo esta afirmação não tem suporte teórico.

O resultado final do trabalho foi de encontro aos objetivos delineados. Este apresentou soluções para os problemas que se colocaram na sua génese, e levantou outros para posterior resolução em trabalhos futuros.

Trabalho Futuro As conclusões que se obtiveram, e as questões que ficaram em aberto neste trabalho, tornam pertinentes um conjunto de trabalhos futuros. No contexto do hardware utilizado sugere-se duas oportunidades de estudo, a alteração do sensor e a diversificação dos objetos. O sensor como foi referido anteriormente é limitativo em termos de resolução, logo seria interessante testar uma alternativa mais capaz. As peças testadas foram interessantes e representativas para testar o sistema, contudo o passo que se segue é o aumento da diversidade, para o qual o sistema está pensado.

A conversão do modelo CAD apesar do bom desempenho apresenta ainda uma oportunidade de melhoria, na implementação da conversão num só passo. O módulo de deteção do contentor já recolhe toda a informação de condicionamento da trajetória, sendo o próximo passo a sua introdução num software de simulação de trajetórias como o MoveIt.

A prestação dos descritores não apresentou um desempenho pleno para justificar a sua utilização em todos os futuros trabalhos. Baseado nessa constatação sugere-se a procura de alternativas mais adequadas.

A análise das correspondências foi limitada a uma avaliação visual e a métrica sugerida neste trabalho, como tal seria interessante o desenvolvimento de uma outra que pudesse complementar as utilizadas.

Bibliografia

- [1] I. F. of Robotics, *248,000 industrial robots revolutionising the global economy*, <http://www.ifr.org/news/ifr-press-release/world-record-816/>, Accessed: 2016-10-8.
- [2] Photoneo, *Bin picking*, <http://www.photoneo.com/product-detail/bin-picking-solution/>, Accessed: 2016-10-11.
- [3] D. Buchholz, *Bin-picking new approaches for a classical problem*, Ph.D. dissertation, 2015.
- [4] L. G. Roberts, *Machine perception of three-dimensional solids*, Ph.D. dissertation, 1963.
- [5] Y. Shirai e M. Suwa, “Recognition of polyhedrons with a range finder”, *International Joint Conference on Artificial Intelligence*, pp. 80–87, 1971.
- [6] W. A. Perkins, “A model-based vision system for industrial parts”, *IEEE Transactions on Computers*, pp. 126–143, 1978.
- [7] S. N. e. a. K. Ikeuchi B. K. Horn, “Picking up an object from a pile of objects”, *First International Symposium on Robotics Research*, pp. 139–166, 1983.
- [8] F. Spenrath, M. Palzkill, A. Pott e A. Verl, “Object recognition: bin-picking for industrial use”, em *Robotics (ISR), 2013 44th International Symposium on*, out. de 2013, pp. 1–3. DOI: 10.1109/ISR.2013.6695743.
- [9] S. Akizuki e M. Hashimoto, “Position and pose recognition of randomly stacked objects using highly observable 3d vector pairs”, em *IECON 2014 - 40th Annual Conference of the IEEE Industrial Electronics Society*, out. de 2014, pp. 5266–5271. DOI: 10.1109/IECON.2014.7049303.
- [10] infaimon, *Inpicker*, <http://www.infaimon.com/pt/inpicker>, Accessed: 2016-10-18.
- [11] MVTec, *Halcon - the power of machine vision*, <http://www.mvtec.com/products/halcon/>, Accessed: 2016-10-18.
- [12] HALCON, *Solution guide iii-c 3D Vision*, Muchen, 2015.
- [13] MVTec, *Surface-based 3d matching*, <http://www.mvtec.com/services-solutions/technologies/matching/>, Accessed: 2016-10-18.
- [14] Beltech, *3d random bin picking randomly placed pipes*, <http://www.beltech.nl/portfolio/random-bin-picking-pipes/>, Accessed: 2016-10-18.
- [15] F. America, *Fanuc america demonstrates a variety of 3d bin picking applications with the next generation irvision 3d area sensor at imts 2014*, <http://www.fanucamerica.com/FanucAmerica-news/press-releases/PressReleaseDetails.aspx?id=42>, Accessed: 2016-10-21.

- [16] FANUC, *Vision functions for robots*, <http://www.fanuc.eu/es/en/robots/accessories/robot-vision>, Accessed: 2016-10-20.
- [17] BBC, *Microsoft kinect 'fastest-selling device on record'*, <http://www.bbc.com/news/business-12697975>, Accessed: 2016-09-08.
- [18] Microsoft, *Kinect for windows sensor components and specifications*, <https://msdn.microsoft.com/en-us/library/jj131033.aspx>, Accessed: 2016-09-08.
- [19] M. A. T. J. et al, *Kinect depth sensor evaluation for computer vision applications*, Denmark, 2012.
- [20] OpenNI, *What is openni?*, <http://openni.ru/index.html>, Accessed: 2016-09-09.
- [21] S. Menon, *How to run the kinect on ubuntu 14.04*, <https://bitbucket.org/samirmenon/scl-manips-v2/wiki/vision/kinect>, Accessed: 2016-09-09.
- [22] ROS, *Opennicamera*, http://wiki.ros.org/openni_camera, Accessed: 2016-09-09.
- [23] —, *How to calibrate a monocular camera*, http://wiki.ros.org/camera_calibration/Tutorials/MonocularCalibration, Accessed: 2016-09-11.
- [24] —, *Intrinsic calibration of the kinect cameras*, http://wiki.ros.org/openni_launch/Tutorials/IntrinsicCalibration, Accessed: 2016-09-11.
- [25] N. Santos, *Bin Picking de Objectos Polimórficos Convexos usando Perceção 3D*, M.S. thesis, Portugal, 2014.
- [26] *Meshlab*, <http://meshlab.sourceforge.net/>, Accessed: 2016-09-16.
- [27] *Blender reference manual*, <https://www.blender.org/manual/contents.html>, Accessed: 2016-09-14.
- [28] R. Rusu e S.Cousins, “3d is here: point cloud library (pcl)”, *IEEE International Conference on Robotics and Automation*, pp. 1–4, 2011.
- [29] S. S. F. Tombari e L. D. Stefano, “Unique signatures of histograms for local surface description”, *11th European conference on computer vision conference on Computer vision*, pp. 356–369, 2010.
- [30] N. B. R. Rusu e M. Beetz, “Fast point feature histograms (fpfh) for 3-d registration”, *IEEE Int. Conf. Robotics and Auto-mation (ICRA)*, 2009.
- [31] PCL, *Mesh2pcd.cpp*, <https://github.com/otherlab/pcl/blob/master/tools/mesh2pcd.cpp>, Accessed: 2016-11-14.
- [32] —, *Ply2pcd.cpp*, <https://github.com/otherlab/pcl/blob/master/tools/ply2pcd.cpp>, Accessed: 2016-11-14.
- [33] —, *Obj2pcd.cpp*, <https://github.com/PointCloudLibrary/pcl/blob/master/tools/obj2pcd.cpp>, Accessed: 2016-11-14.
- [34] A. G. X. Lin e A. Wagen, “3d part identification based on local shape descriptors”, *PerMIS '08 Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems*, pp. 162–166, 2008.
- [35] M. V. A. Aldoma e N. B. et al, “Cad-model recognition and 6dof pose estimation using 3d cues”, *IEEE International Conference on Computer Vision Workshops*, pp. 585–592, 2011.

- [36] C. Khan, *Development and Evaluation of a Kinect based Bin-Picking System*, M.S. thesis, Sweden, 2015.
- [37] *Point Cloud Library Documentation pclvisualizer class reference*, <http://docs.pointclouds.org/1.7.2/a00979.html>, Accessed: 2016-07-18.
- [38] *github PCL sourcecode virtual_scanner.cpp*, https://github.com/PointCloudLibrary/pcl/blob/master/tools/virtual_scanner.cpp, Accessed: 2016-07-27.
- [39] K. K. S. Elberink, *Accuracy and resolution of kinect depth data for indoor mapping applications*, Netherlands, 2012.
- [40] T. F. Tombari, *3d object recognition based on correspondence grouping*, http://pointclouds.org/documentation/tutorials/correspondence_grouping.php#{#}correspondence-grouping, Accessed: 2016-03-13.
- [41] A. Z. F. et al, “Point Cloud Library three-dimensional object recognition and 6 dof pose estimation”, *IEEE Robot. Automat. Mag.*, vol. 19, n° 3, pp. 80–91, 2012.
- [42] Y. Zhong, “Intrinsic shape signatures: a shape descriptor for 3d object recognition”, *ICCV Workshops*, pp. 689–696, 2009.

Apêndices

Apêndice A

Testes de correspondência SHOT

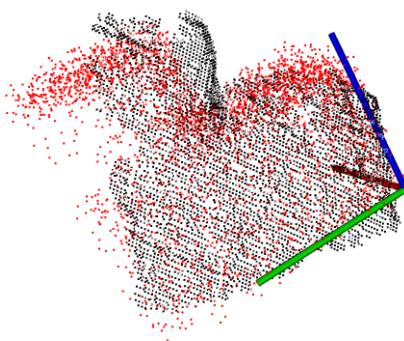


Figura A.1: Ensaio de correspondência com o descritor SHOT e modelo com 4 milímetros de ruído.

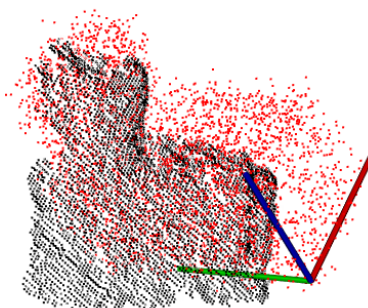


Figura A.2: Ensaio de correspondência com o descritor SHOT e modelo com 5 milímetros de ruído.

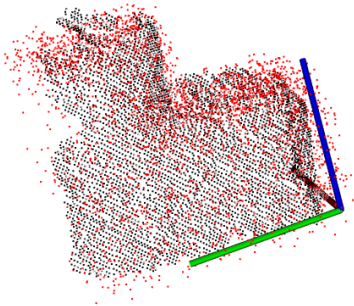


Figura A.3: Ensaio de correspondência com o descritor SHOT e modelo com 6 milímetros de ruído.

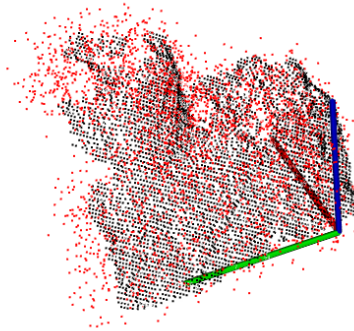


Figura A.4: Ensaio de correspondência com o descritor SHOT e modelo com 7 milímetros de ruído.

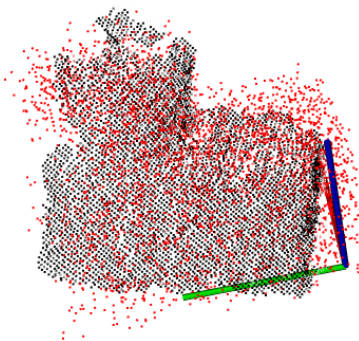


Figura A.5: Ensaio de correspondência com o descritor SHOT e modelo com 8 milímetros de ruído.

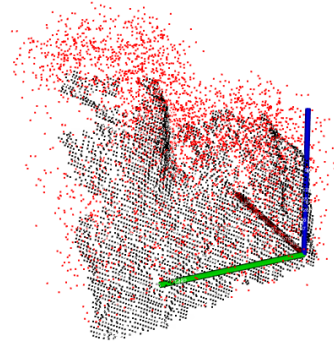


Figura A.6: Ensaio de correspondência com o descritor SHOT e modelo com 9 milímetros de ruído.

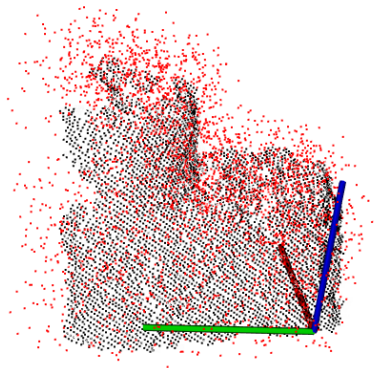


Figura A.7: Ensaio de correspondência com o descritor SHOT e modelo com 10 milímetros de ruído.

Apêndice B

Testes de correspondência FPFH

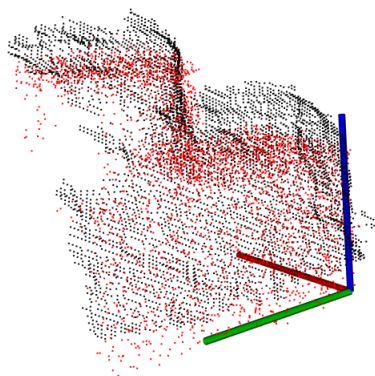


Figura B.1: Ensaio de correspondência com o descritor FPFH e modelo com 3 milímetros de ruído.

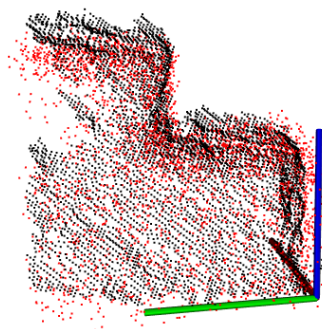


Figura B.2: Ensaio de correspondência com o descritor FPFH e modelo com 4 milímetros de ruído.

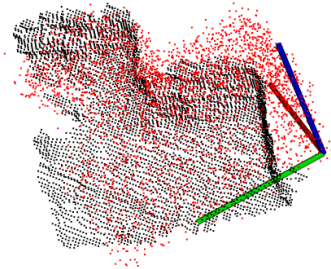


Figura B.3: Ensaio de correspondência com o descritor FPFH e modelo com 5 milímetros de ruído.

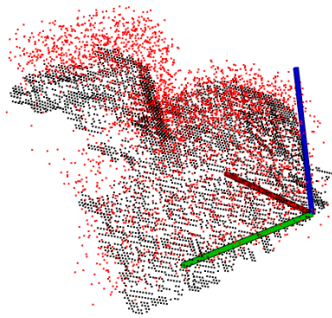


Figura B.4: Ensaio de correspondência com o descritor FPFH e modelo com 6 milímetros de ruído.

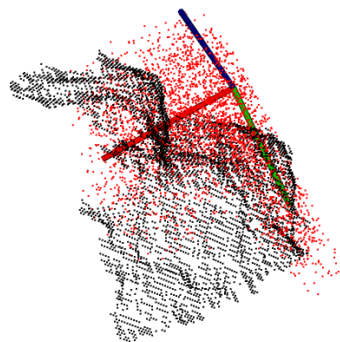


Figura B.5: Ensaio de correspondência com o descritor FPFH e modelo com 7 milímetros de ruído.

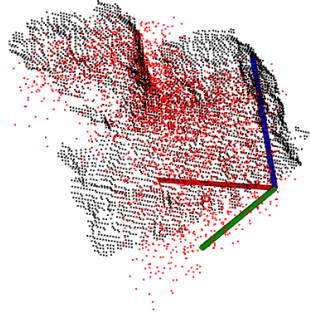


Figura B.6: Ensaio de correspondência com o descritor FPFH e modelo com 8 milímetros de ruído.

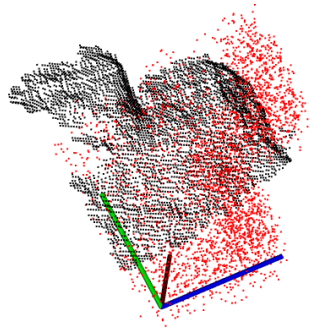


Figura B.7: Ensaio de correspondência com o descritor FPFH e modelo com 9 milímetros de ruído.

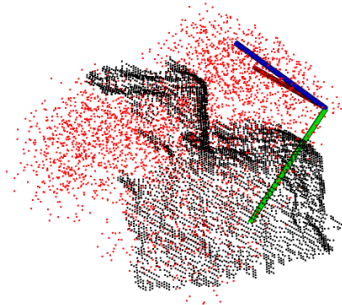


Figura B.8: Ensaio de correspondência com o descritor FPFH e modelo com 10 milímetros de ruído.