



**Diogo Barbosa  
Fernandes**

**Sistema automático de marcação a laser de peças 3D  
Automatic 3D parts laser marking system**





Universidade de Aveiro  
2022

**Diogo Barbosa  
Fernandes**

**Sistema automático de marcação a laser de peças 3D**  
**Automatic 3D parts laser marking system**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Eletrónica e Telecomunicações, realizada sob a orientação científica do Professor Pedro Nicolau Faria da Fonseca e do Professor José Nuno Panelas Nunes Lau, Professores do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro.

Dissertação desenvolvida no âmbito do projeto AdaptMark - Sistema inteligente, autónomo e flexível de marcação robotizada de componentes, POCI-01-0247-FEDER-046982, projeto de I&DT em co-promoção promovido pela JPM Industry e pela Universidade de Aveiro



**o júri / the jury**

presidente / president

Prof. Doutor José Luís Costa Pinto de Azevedo  
Professor auxiliar da Universidade de Aveiro

vogais / examiners committee

Prof. Doutor Luís Paulo Gonçalves dos Reis  
Professor associado da Faculdade de Engenharia da Universidade do Porto

Prof. Doutor Pedro Nicolau Faria da Fonseca  
Professor auxiliar da Universidade de Aveiro



## agradecimentos / acknowledgements

Começo por agradecer aos meus orientadores Professor Pedro Fonseca e Professor Nuno Lau não só por toda a ajuda e apoio prestados durante a realização desta dissertação, mas também pelo conhecimento que me foi transmitido durante a realização da mesma. Agradeço ainda ao Reza Javanmard, investigador participante no projeto, toda a ajuda prestada e também ao Professor José Luís Azevedo que, apesar de não estar diretamente ligado ao projeto, dedicou muito do seu tempo a ajudar a resolver os problemas de *hardware* que apareceram durante a realização deste projeto.

Aproveito também para agradecer à JPM por me proporcionar a possibilidade de desenvolver a dissertação num projeto tão inovador e que certamente será de grande valor para a empresa e, em especial, ao Engenheiro Fillipe Ribeiro por toda a ajuda prestada durante a realização desta dissertação.

Aproveito ainda para agradecer a todos aqueles que contribuíram para a conclusão do meu percurso académico e, especialmente, àqueles que sempre estiveram ao meu lado nos momentos de maior aperto.

Por último, e o mais especial dos agradecimentos, é dedicado à minha família, especialmente ao meu pai, mãe e irmã, por toda a ajuda, apoio e confiança depositados em mim ao longo de todo o meu percurso.



## Palavras Chave

Visão por computador, *point cloud*, reconhecimento de objetos, descritores 3D, marcação a laser, robótica

## Resumo

Com o aumento cada vez mais acentuado da população mundial, cresce também o desafio de continuar a responder às suas necessidades. Desta forma, o mundo industrial é colocado mais uma vez à prova, levando assim ao surgimento da Indústria 4.0.

É nesse sentido que surge o projeto AdaptMark, lançado como desafio pela empresa JPM à Universidade de Aveiro, onde o objetivo deste consiste na investigação e desenvolvimento de um sistema robótico de marcação a laser de peças de diferentes geometrias e materiais, dotado de elevada inteligência, autonomia de operação, flexibilidade e adaptabilidade. O projeto pode ser dividido em três subsistemas: subsistema de identificação, subsistema de manipulação e marcação laser e subsistema de verificação de qualidade, sendo nesta dissertação abordado o subsistema de identificação, que tem como objetivo a correta identificação das peças que chegam aleatoriamente num tapete transportador, bem como a correta estimação da posição de marcação nas mesmas.

Relativamente ao subsistema de identificação, este pode ser dividido em duas partes: uma delas relativa à identificação das peças e a outra relativa à estimativa da pose destas. Na parte da identificação das peças, foram testados vários descritores 3D e, no final, é possível verificar que todas as peças são corretamente identificadas. Relativamente à parte da estimativa da pose das peças, acabou por se efetuar a classificação das peças em relação à sua forma e, na presente dissertação, apenas são abordadas as peças retangulares.



**Keywords**

Computer vision, point cloud, object recognition, 3D descriptors, laser marking, robotics

**Abstract**

As the world's population increases, the challenge of continuing to respond to their needs is also growing. In this way, the industrial world is once again tested, leading to the emergence of Industry 4.0.

It is in this sense that the AdaptMark project emerges, launched as a challenge by the JPM company to the University of Aveiro, where the objective of this is the research and development of a robotic laser marking system of parts of different geometries and materials, endowed with high intelligence, autonomy of operation, flexibility and adaptability.

The project can be divided into three subsystems: identification subsystem, laser manipulation and marking subsystem and quality verification subsystem, and this dissertation addresses the identification subsystem, which aims to correctly identify the parts that arrive randomly on a conveyor belt, as well as the correct estimation of the position of marking on them.

Regarding the identification subsystem, it can be divided into two parts: one related to the identification of the parts and the other related to the estimation of their pose. In the part of the pieces identification, several 3D descriptors were tested and, in the end, it is possible to verify that all the pieces are correctly identified. Regarding the part about estimating the pose of the pieces, we ended up with the classification of the pieces about their shape and, in this dissertation, only rectangular pieces are addressed.



# Conteúdo

<b>Conteúdo</b>	<b>i</b>
<b>Lista de Figuras</b>	<b>iii</b>
<b>Lista de Tabelas</b>	<b>v</b>
<b>Siglas</b>	<b>vii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Enquadramento . . . . .	1
1.2 A empresa . . . . .	2
1.3 Objetivos . . . . .	2
1.4 Estrutura do documento . . . . .	4
<b>2 Reconhecimento e estimativa da posição de objetos 3D</b>	<b>5</b>
2.1 Formatos e representação 3D . . . . .	5
2.2 Bibliotecas 3D . . . . .	7
2.2.1 Point Cloud Library (PCL) . . . . .	7
2.2.2 Open3D . . . . .	8
2.2.3 HALCON . . . . .	8
2.3 Transformações geométricas 3D . . . . .	9
2.4 Descritores 3D . . . . .	10
2.5 Técnicas de registo . . . . .	15
2.5.1 RANSAC . . . . .	16
2.5.2 Iterative Closest Point (ICP) . . . . .	17
2.6 ROS . . . . .	17
2.6.1 Ferramentas de desenvolvimento . . . . .	18
2.6.2 Transformações ROS . . . . .	19
2.6.3 Simulador Gazebo . . . . .	20
2.7 Trabalhos relacionados . . . . .	20
2.8 Conclusão . . . . .	21
<b>3 Metodologia e resultados no reconhecimento de peças 3D</b>	<b>23</b>
3.1 Arquitetura proposta . . . . .	23
3.2 Simulação do subsistema de identificação . . . . .	23
3.3 Sensorização . . . . .	25
3.3.1 Sensor utilizado na dissertação . . . . .	26
3.3.2 Encoder utilizado na dissertação . . . . .	28
3.4 Peças de teste utilizadas na dissertação . . . . .	28
3.5 Aquisição da <i>point cloud</i> . . . . .	28
3.5.1 Extração do perfil . . . . .	30
3.5.2 Extração da superfície . . . . .	30
3.5.3 Configurações finais do sensor . . . . .	32

3.6	Identificação das peças . . . . .	34
3.6.1	Identificação de peças com o sensor . . . . .	35
3.6.2	Descritores 3D . . . . .	35
3.7	Conclusões . . . . .	43
<b>4</b>	<b>Metodologia e resultados na estimativa da pose das peças</b>	<b>45</b>
4.1	Estimativa da pose das peças . . . . .	45
4.2	Estimativa da pose através do ICP . . . . .	45
4.3	Estimativa da rotação das peças . . . . .	47
4.4	Estimativa da distância entre peças . . . . .	49
4.5	Estimativa da posição das peças . . . . .	50
4.6	Marcação das peças . . . . .	51
4.7	Conclusão . . . . .	52
<b>5</b>	<b>Conclusão</b>	<b>55</b>
5.1	Considerações finais . . . . .	55
5.2	Trabalho futuro . . . . .	55
	<b>Bibliografia</b>	<b>57</b>
	<b>Apêndices</b>	
	<b>A Resultados completos dos <i>models</i></b>	
	<b>B Matrizes de discriminação completas</b>	
	<b>C Histogramas completos com cotas das peças</b>	

# Lista de Figuras

1.1	Logótipo da JPM. . . . .	2
1.2	Sistema geral do projeto AdaptMark. . . . .	3
2.1	Formatos 3D. [1] . . . . .	6
2.2	Módulos da PCL. [5] . . . . .	8
2.3	Interface do HALCON. . . . .	9
2.4	Representação gráfica do <i>Darboux frame</i> e as <i>features</i> angulares do PFH. [9]	11
2.5	Cálculo de componente <i>viewpoint</i> . [9] . . . . .	12
2.6	Cálculo de componente estendida do FPFH. [9] . . . . .	12
2.7	Componentes do descritor ESF. [15] . . . . .	13
2.8	Histograma resultante do descritor OUR-CVFH. [17] . . . . .	14
2.9	Exemplo de categorização das superfícies nas várias classes. [19] . . . . .	14
2.10	Exemplo ilustrativo do descritor GOOD. [21] . . . . .	16
2.11	Ilustração do RANSAC com os <i>inliers</i> a serem representados pelos pontos verdes e os <i>outliers</i> pelos pontos vermelhos. . . . .	16
2.12	Visualização de todos os <i>frames</i> <i>tf</i> do robô UR10. . . . .	19
2.13	Interface gráfica do simulador Gazebo. . . . .	20
3.1	Esquema geral do subsistema de identificação. . . . .	24
3.2	Ambiente de simulação desenvolvido. . . . .	24
3.3	Sensores 3D. . . . .	25
3.4	Conexões da Gocator. [36] . . . . .	27
3.5	Interface Web. [36] . . . . .	28
3.6	Disco utilizado para a calibração do encoder. . . . .	28
3.7	Peças usadas durante o desenvolvimento da dissertação. . . . .	29
3.7	Peças usadas durante o desenvolvimento da dissertação. . . . .	30
3.8	Exemplificação da aquisição do perfil. . . . .	31
3.9	Resultados da extração do perfil. . . . .	31
3.10	Exposição simples. . . . .	33
3.11	Exposição dinâmica. . . . .	33
3.12	Exposição múltipla. . . . .	34
3.13	Diagrama de fluxo da identificação das peças. . . . .	36
3.14	Simplificação das matrizes de discriminação. . . . .	37
3.15	Matrizes de discriminação obtidas pelos descritores. . . . .	38
3.16	Resultados obtidos através da softmax. . . . .	40
3.17	Histogramas e vista frontal das peças n.º 2 e n.º 8. . . . .	41
3.18	Matrizes de discriminação e softmax obtidas. . . . .	42
4.1	Diagrama de fluxo para obter a estimativa da pose. . . . .	45
4.2	<i>Point cloud</i> antes e após aplicar o RANSAC . . . . .	46
4.3	Teste realizado com peças circulares e retangulares. . . . .	46
4.4	Resultados obtidos através do ICP. . . . .	47

4.5	Classificação das peças com base na sua forma. . . . .	48
4.6	Resultados obtidos na detecção do retângulo de menor área . . . . .	48
4.7	Estimativa da distância entre peças. . . . .	49
4.8	Scan da peça n.º 13 . . . . .	50
4.9	Etapas de scan . . . . .	50
4.10	Estimativa da posição da peça . . . . .	50
4.11	Diagrama de fluxo para efetuar a marcação das peças. . . . .	51
4.12	Simulação do marcador a laser . . . . .	52
4.13	Função distribuição acumulada . . . . .	52

# Lista de Tabelas

2.1	Formatos 3D. . . . .	6
3.1	Parâmetros para a escolha do sensor. . . . .	26
3.2	Configurações do sensor. . . . .	34
3.3	Dimensões dos descritores 3D. . . . .	36
3.4	Valores obtidos após a softmax. . . . .	39
3.5	Valores obtidos na soma do descritor GOOD com as <i>features</i> altura e área após a softmax. . . . .	43
4.1	Rotações obtidas através do ICP. . . . .	46
4.2	Translações obtidas através do ICP. . . . .	47
4.3	Rotações obtidas através do método do retângulo de menor área. . . . .	48
4.4	Média da estimativa da posição da peça n.º 13. . . . .	51



# Siglas

**CVFH** Clustered Viewpoint Feature Histogram.

**DART** Dynamic Animation and Robots Toolkit.

**ESF** Ensemble of Shape Functions.

**FLANN** Fast Library for Approximate Nearest Neighbors.

**FPFH** Fast Point Feature Histograms.

**GASD** Globally Aligned Spatial Distribution.

**GOOD** Global Orthographic Object Descriptor.

**GRSD** Global Radius-Based Surface Descriptor.

**ICP** Iterative Closest Point.

**IoT** Internet of Things.

**IRIS** Intelligent Robotics and Systems.

**ODE** Open Dynamics Engine.

**OGRE** Open Source 3D Graphics Engine.

**OUR-CVFH** Oriented, Unique and Repeatable Clustered Viewpoint Feature Histogram.

**PCA** Principal Component Analysis.

**PCL** Point Cloud Library.

**PFH** Point Feature Histograms.

**RANSAC** RANdom SAmple Consensus.

**ROS** Robotic Operating System.

**RSD** Radius-based Surface Descriptor.

**SAC** SAmple Consensus.

**SGURF** Semi-Global Unique Reference Frames.

**VFH** Viewpoint Feature Histogram.



# Capítulo 1

## Introdução

A presente dissertação está enquadrada no projeto AdaptMark, projeto de I&DT em co-promoção, desenvolvido pela JPM Industry e pela Universidade de Aveiro. Este projeto tem como objetivo o desenvolvimento de um sistema robótico com a capacidade de executar a marcação a laser de peças com geometrias e materiais diferentes, bem como a verificação da qualidade da marcação nas mesmas.

Neste capítulo começa-se por fazer um breve enquadramento do projeto em questão, seguido da apresentação da empresa, objetivos do projeto e, por fim, é realizada uma breve descrição da estrutura e organização dos conteúdos deste documento.

### 1.1 Enquadramento

Com o aumento cada vez mais acentuado da população mundial, cresce também o desafio de continuar a responder às suas necessidades. Desta forma, o mundo industrial é colocado mais uma vez à prova, levando assim ao surgimento da Indústria 4.0, por muitos conhecida como a Quarta Revolução Industrial.

Tendo como principais considerações as mudanças a nível industrial, a Primeira Revolução Industrial, ocorrida nos finais do século XVIII, ficou marcada principalmente pela utilização do vapor na produção mecânica e pelo aparecimento do primeiro tear mecânico. A Segunda Revolução Industrial, iniciada a finais do século XIX, ficou marcada pela produção em larga escala baseada na eletricidade e pela invenção da primeira linha de montagem, o que acabaria por impulsionar o setor industrial. A Terceira Revolução Industrial, iniciada no início dos anos 1970, produz grandes avanços tecnológicos em diversas áreas, sendo a robótica marcada pela progressiva automatização das linhas de produção.

*“What if existing manufacturing and large technological innovations came together to create the next big manufacturing reinvention? Bingo. This is the fourth manufacturing revolution, and it’s happening right now.”* - Olivier Scalabre, Senior Partner and Managing Director, BCG Paris, 2016

A Quarta Revolução Industrial é representada pela integração de diferentes tecnologias, como por exemplo, sensores, *Internet of Things*(IoT), inteligência artificial, automação, aprendizagem automática, entre outros. A conjugação destas tecnologias possibilita a realização de processos mais rápidos, mais flexíveis e mais eficientes, permitindo desta forma a produção de produtos em grandes quantidades a custos reduzidos, garantindo elevados padrões de qualidade.

É nesse sentido que surge o projeto AdaptMark, lançado como desafio pela empresa JPM à Universidade de Aveiro.

## 1.2 A empresa



**Figura 1.1:** Logótipo da JPM.

A JPM *Industry*<sup>1</sup> foi fundada em 1994, em Vale de Cambra. A empresa teve vários marcos importantes, mas foi em 2009 que se criou o Grupo JPM atualmente constituído pela JPM Industry, pela JPM Renováveis e pela *Jointsteel Process Technologies*.

O grupo atua em diversas áreas ligadas ao desenvolvimento e implementação de equipamentos industriais, soluções de engenharia e projetos industriais, tendo como principais clientes várias companhias de escala global.

Atualmente, o grupo possui uma forte presença na indústria alimentar, química e farmacêutica.

## 1.3 Objetivos

O projeto tem como objetivo a investigação e o desenvolvimento de um sistema robótico de marcação a laser de peças de diferentes geometrias e materiais, dotado de elevada inteligência, autonomia de operação, flexibilidade e adaptabilidade. Tanto as configurações dos parâmetros do laser, como a posição de marcação das peças, deverão ficar armazenados numa base de dados que será consultada à medida que vão aparecendo peças para sofrerem a respetiva marcação. Após a marcação, todas as peças serão inspecionadas por um sistema de visão que irá determinar se a marcação foi bem feita, ou não. Caso a marcação esteja de acordo com o esperado, a peça é aceite. Caso contrário, a peça é rejeitada e o sistema irá fazer sugestões de ajustes nos parâmetros de marcação da peça, de modo a corrigir o problema na próxima iteração. No final do projeto ter-se-á uma solução inovadora quando comparada com os sistemas atuais, os quais necessitam de intervenção humana para se conseguir garantir uma maior flexibilidade e desta forma, o sistema irá permitir a identificação e gravação de diferentes peças, adaptando os parâmetros de funcionamento a cada peça em tempo real, sem intervenção humana e com alta cadência.

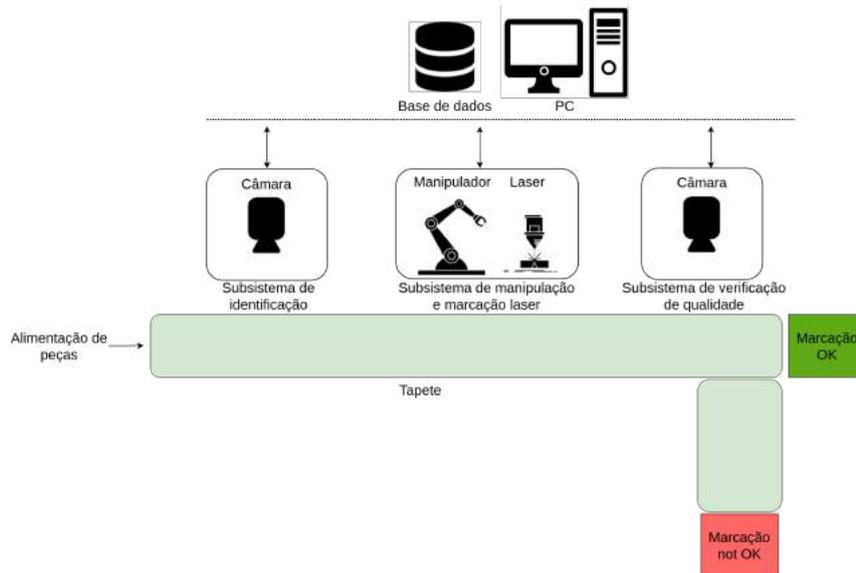
O projeto AdaptMark pode ser dividido em três subsistemas:

- **Subsistema de identificação** - tem como objetivo a correta identificação das peças que chegam aleatoriamente num tapete transportador, bem como a correta estimação da posição de marcação nas mesmas. Cada tipo de peça receberá uma determinada marcação numa localização específica e, por esta razão, é importante que o sistema consiga identificar de forma autónoma o tipo de peça presente, bem como estimar corretamente a sua posição, para que seja possível proceder à marcação conforme o pretendido para os diferentes tipos de peças;
- **Subsistema de manipulação e marcação laser** - tem como objetivo a marcação das peças que chegam no tapete transportador;
- **Subsistema de verificação de qualidade** - tem como objetivo a verificação da qualidade das peças marcadas.

---

<sup>1</sup><https://jpm.pt/pt/inicio/>

Na Figura 1.2 apresenta-se uma ilustração do sistema descrito anteriormente.



**Figura 1.2:** Sistema geral do projeto AdaptMark.

Tendo em consideração a fase atual do projeto, ainda existem os seguintes pontos por concluir:

- Estudo de soluções para o subsistema de identificação 3D para que este seja capaz de identificar as peças no tapete transportador;
- Estudo de soluções para o subsistema de identificação 3D para que este seja capaz de fornecer ao subsistema de manipulação robótica a identificação, localização e orientação das peças no tapete, bem como as coordenadas para a gravação das mesmas;
- Criação de uma base de dados para armazenamento dos dados necessários ao correto funcionamento do sistema em geral;
- Desenvolvimento de uma interface que permita inserir uma nova peça no sistema;
- Melhoria dos parâmetros de marcação a laser e controlo de qualidade;
- Gestão do robô e do tapete, bem como a sua sincronização;

Tendo em atenção os pontos mencionados acima, a presente dissertação tem como objetivos:

- Estudo de soluções para o subsistema de identificação 3D, que seja capaz de identificar as peças no tapete transportador com uma distância mínima de 50 mm entre si;
- Estudo de soluções para o subsistema de identificação 3D capazes de fornecerem ao subsistema de manipulação robótica a identificação, localização e orientação das peças no tapete, bem como as coordenadas para a gravação das mesmas;

No final do projeto, este sistema será baseado no *Robotic Operating System*(ROS)<sup>2</sup>, bem como na *Point Cloud Library*(PCL)<sup>3</sup> uma vez que conseguem reunir as ferramentas necessárias para o desenvolvimento do mesmo.

<sup>2</sup><https://www.ros.org>

<sup>3</sup><https://pointclouds.org>

Da presente dissertação resultaram contribuições para os entregáveis do projeto Adapt-Mark, nomeadamente para os entregáveis E3.1 – Desenho da arquitetura técnica do sistema e E4.3 – Resumo da verificação e validação do protótipo num ambiente de laboratório. Esta dissertação está ainda a contribuir para para o entregável E5.1 – Avaliação global do desempenho da solução e ainda para o artigo que está a ser desenvolvido para a revista *IEEE Transactions on Automation Science and Engineering*.

## 1.4 Estrutura do documento

Este documento divide-se nos seguintes capítulos:

- Capítulo 1 - Introdução: Neste capítulo é feita uma breve descrição da empresa, a motivação para o desenvolvimento deste projeto, o enquadramento, os objetivos do mesmo e, finalmente, a presente secção relativa à estrutura do documento.
- Capítulo 2 - Reconhecimento e estimativa da posição de objetos 3D: Neste capítulo é apresentado o estado de arte do reconhecimento de objetos 3D. Para além disso, faz-se também uma breve descrição do que é o ROS, bem como algumas ferramentas por ele disponibilizadas.
- Capítulo 3 - Metodologia e resultados no reconhecimento de peças 3D: Neste capítulo é apresentado o trabalho prático desenvolvido para o subsistema de identificação. Ao longo do capítulo são apresentadas todas as experiências efetuadas, e respetivos resultados obtidos.
- Capítulo 4 - Metodologia e resultados na estimativa da pose de peças 3D: Neste capítulo é apresentado todo o trabalho desenvolvido para uma correta identificação da posição e rotação das peças, bem como uma correta estimação da posição de marcação sobre estas.
- Capítulo 5 - Conclusão e trabalho futuro: Neste capítulo faz-se uma análise geral sobre o trabalho realizado. Para além disso, tiram-se algumas conclusões sobre o resultado final e apresentam-se algumas sugestões de trabalho futuro.

## Capítulo 2

# Reconhecimento e estimativa da posição de objetos 3D

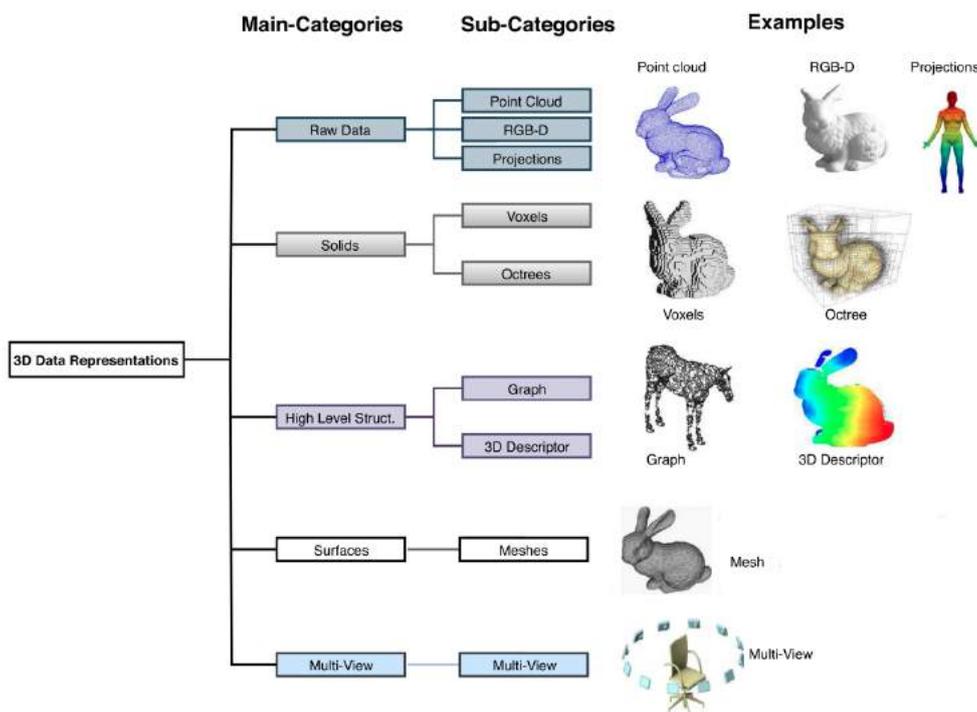
Este capítulo tem como objetivo a apresentação de várias tecnologias e metodologias na área de visão por computador que visam o reconhecimento de objetos. Começa-se por fazer uma breve apresentação dos formatos e representação de dados 3D, seguida da apresentação de algumas bibliotecas 3D. Posteriormente é feito um estudo de alguns descritores 3D, sendo estes os responsáveis por extrair as características de objetos 3D para que, desta forma, sejam corretamente identificados. Em seguida, é realizada a apresentação de outros métodos de correspondência de *features* que serão importantes para a identificação da pose dos objetos em questão. A seguir, efetua-se uma breve apresentação do ROS, bem como algumas das ferramentas por ele disponibilizadas e, por fim, são apresentados alguns trabalhos efetuados com objetivos idênticos ao da presente dissertação, ou seja, a identificação de objetos 3D.

### 2.1 Formatos e representação 3D

A representação de dados 3D [1] pode ser feita das seguintes formas (Figura 2.1):

- **Dados em bruto** - Podem ser obtidos através de diferentes dispositivos e podem ser representados de várias maneiras, como por exemplo, *point cloud*, dados RGB-D e projeções 3D;
  - *Point cloud* - é um conjunto de pontos que pode representar uma forma ou um objeto 3D e é utilizada para várias tarefas de visão por computador, como por exemplo, segmentação e classificação, reconhecimento de objetos, entre outras;
  - RGB-D - faz-se a conjugação de dados RGB com a sua profundidade na imagem;
  - Projeções de dados 3D - o mapeamento de pontos 3D para planos 2D é feito através de projeções 3D e é implementado através de projeções fictícias que fornecem aos dados projetados características importantes do objeto 3D real. O modelo 3D é frequentemente transformado usando vários métodos de projeção em uma grade 2D com recursos úteis.
- **Representações sólidas de modelos 3D** - A informação de controlo do espaço virtual para um objeto específico pode ser encontrada em representações sólidas de modelos 3D. Normalmente a informação é binária, ou seja, o espaço ou está ou não está ocupado e as duas principais representações desta categoria são *octree* e *voxels*;
- **Superfícies** - Normalmente, os polígonos de superfície são utilizados para representar os limites de objetos 3D que rodeiam a parte interior do objeto.

- **Estruturas de alto nível** - Para descrever um objeto como sendo representativo de uma determinada categoria, um objeto 3D deve ser representado de forma concisa. Os descritores de forma 3D de alto nível são uma representação condensada que contém as propriedades geométricas do objeto e podem ser usados para expressar formas 3D;
- **Vistas múltiplas** - Outro método de representação de dados 3D consiste na aquisição de um conjunto de imagens retiradas em vários ângulos sobre o objeto e na utilização desse conjunto de imagens resultante como entrada para uma rede neural, que pode ser usada para tarefas de análise de forma.



**Figura 2.1:** Formatos 3D. [1]

Os vários formatos 3D podem ser guardados em ficheiros com diversas extensões, sendo algumas delas apresentadas na Tabela 2.1.

Formato	Nome	Extensão
STL	Stereo Lithography	.stl
SAT	Standard ACIS Text	.sat
PLY	Stanford Polygon Format	.ply
PCD	Point Cloud Data	.pcd
STEP	Standard for the Exchange of Product Data	.stp
XYZ	Point Cloud	.xyz
AMF	Additive Manufacturing	.amf
3MF	3D Manufacturing Format	.3mf
DXF	AutoCAD Drawing Exchange Format	.dxf
CSG	Constructive solid geometry	.csg
3DS	3D Studio	.3ds

**Tabela 2.1:** Formatos 3D.

## 2.2 Bibliotecas 3D

Nesta secção são apresentadas algumas das bibliotecas tidas em consideração para a realização desta dissertação.

### 2.2.1 Point Cloud Library (PCL)

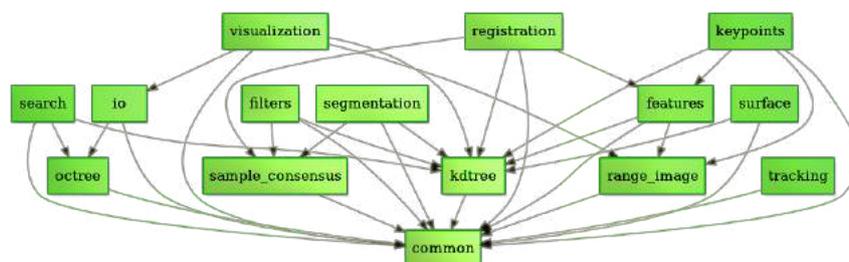
Com o aumento da utilização de dados 3D, é necessário o desenvolvimento de ferramentas de processamento e algoritmos que permitam tirar o máximo proveito destes dados. A PCL [2], introduzida em 2011, é uma biblioteca de C++ para o processamento de *point cloud* e oferece um grande número de algoritmos de processamento 3D, nomeadamente:

- filtragem
- estimação de características
- reconstrução de superfícies
- ajuste de modelos
- segmentação
- *registration* [3]
- entre outros...

De modo a simplificar tanto a utilização como o desenvolvimento, esta biblioteca foi dividida em vários módulos [4]. O conjunto mais importante de módulos PCL são os seguintes:

- *Features* - A biblioteca *pcl\_features* contém estruturas de dados e mecanismos para estimar *features* 3D de dados de *point cloud*, como por exemplo, descritores 3D;
- *Filters* - A biblioteca *pcl\_filters* contém mecanismos de remoção de ruído para dados 3D, como por exemplo, redução do número de amostras, remoção de *outliers*, entre outros;
- *I/O* - A biblioteca *pcl\_io* implementa operações de leitura e escrita de *point clouds*, como por exemplo a leitura e escrita de e para ficheiros “.*pcd*” e “.*ply*”, bem como a aquisição de dados de uma grande variedade de dispositivos capazes de detetar *point clouds*;
- *KdTree* - A biblioteca *pcl\_kdtree* fornece a estrutura de dados kd-tree, utilizando o *Fast Library for Approximate Nearest Neighbors*(FLANN), que permite pesquisas rápidas de vizinhos mais próximos;
- *Keypoints* - A biblioteca *pcl\_keypoints* contém implementações de algoritmos de deteção de pontos de interesse de *point clouds* e pode ser utilizada como um passo de pré-processamento para a extração dos descritores de *features*;
- *Octree* - A biblioteca *pcl\_octree* fornece um conjunto de métodos eficientes que permitem a criação de uma estrutura de dados hierárquica em árvore a partir dos dados da *point cloud*;
- *Range Image* - A biblioteca *range\_image* contém 2 classes para representar e trabalhar com *pcl\_range\_images*. Uma *range\_image* é uma imagem onde os valores de cada pixel representam uma distância;

- *Registration* - A biblioteca *pcl\_registration* implementa uma grande quantidade de algoritmos de *registration*, cujo objetivo consiste no alinhamento de *point clouds*. Um exemplo destes algoritmos é o *Iterative Closest Point*(ICP);
- *Sample Consensus* - A biblioteca *pcl\_sample\_consensus* contém métodos *Sample Consensus* (SAC), como por exemplo, o *RANdom SAMple Consensus* (RANSAC) e modelos de planos, cilindros, esferas, entre outros;
- *Segmentation* - A biblioteca *pcl\_segmentation* contém algoritmos para a segmentação de *point clouds* em diferentes *clusters*. Este tipo de algoritmos são muito úteis para os casos de *point clouds*;
- *Surface* - A biblioteca *pcl\_surface* permite-nos fazer a reconstrução de superfícies completas com base em *scans* 3D;
- *Visualization* - A biblioteca *pcl\_visualization* foi desenvolvida com o intuito de permitir visualizar os resultados obtidos através dos algoritmos que tratam os dados das *point clouds*.



**Figura 2.2:** Módulos da PCL. [5]

### 2.2.2 Open3D

O Open3D [6] foi apresentado por Qian-Yi Zhou, Jaesik Park e Vladlen Koltun, investigadores dos Laboratórios Intel, e trata-se de uma biblioteca *open-source* que suporta o rápido desenvolvimento de software que lida com dados 3D e as suas principais características são:

- Suporte para *machine learning* 3D com *PyTorch* e *TensorFlow*;
- Aceleração do GPU para as principais operações 3D;
- Disponível em C++ e Python.

Esta biblioteca também disponibiliza estruturas de dados para a representação de *point clouds*, *meshes*, e imagens RGB-D. Para além disto, também é possível efetuar a reconstrução de cenários, alinhamento de superfícies 3D e visualização de dados 3D.

Durante a dissertação, esta biblioteca não foi muito utilizada uma vez a PCL continha material suficiente para a sua realização.

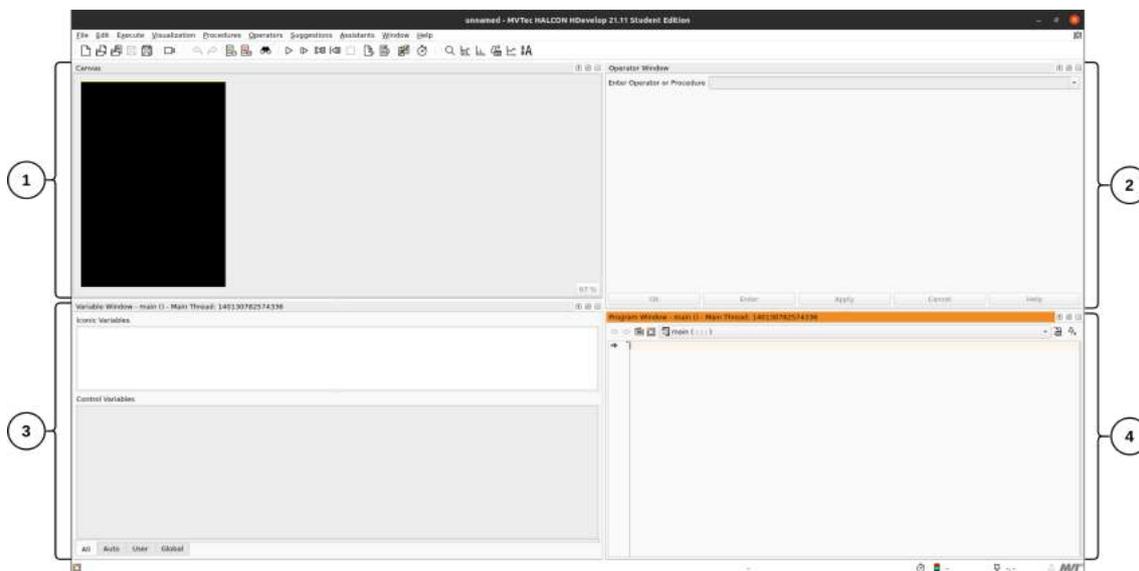
### 2.2.3 HALCON

O HALCON é um software de visão por computador desenvolvido pela MVTec Software GmbH (Figura 2.3). É utilizado em diferentes indústrias, como por exemplo, médica, automóvel, vigilância e é apropriado para processamento de imagem de baixo e alto nível.

Este software é capaz de analisar imagens 1D, 2D, 3D e é compatível com vários sistemas operativos e com várias linguagens de programação.

Para desenvolver algoritmos neste software [7] é necessário utilizar uma linguagem *script* suportada pelo seu próprio IDE gráfico denominado de HDevelop, sendo depois possível exportar o código desenvolvido para um ficheiro “.txt”, ou até mesmo para outras linguagens, como por exemplo, C e C#.

A interface do HDevelop pode ser dividida em quatro janelas principais, como apresentado na Figura 2.3.



**Figura 2.3:** Interface do HALCON.

1. *Canvas window* com *graphics window* - A janela *graphics* é utilizada para apresentar dados como imagens, regiões; A janela *Canvas* permite-nos organizar várias janelas *Graphics*;
2. *Operator window* - Nesta janela, pode-se editar e apresentar um operador bem como o número de parâmetros, o tipo e valor destes;
3. *Variable window* - É aqui que são apresentadas as variáveis e respetivos valores do programa que está a ser desenvolvido;
4. *Program window* - É nesta janela que são desenvolvidos todos os programas.

## 2.3 Transformações geométricas 3D

Uma transformação geométrica é uma operação que pode ser utilizada para alterar algumas características de um objeto, como por exemplo, a sua posição, orientação, forma e até mesmo o seu tamanho, com base numa matriz transformação. Existem vários tipos de transformações geométricas, no entanto, neste documento são apresentadas as mais usuais:

- Similaridade: baseiam-se numa matriz transformação que é constituída por rotação, escala uniforme e translação nos eixos x, y e z;
- Afim: baseiam-se numa matriz transformação composta rotação, translação nos eixos x, y e z, escala não uniforme e enviesamento;

- Homografia: baseiam-se numa matriz transformação composta por rotação, translação nos eixos  $x$ ,  $y$  e  $z$ , escala não uniforme, enviesamento e a perspetiva.

Dado que neste sistema irão existir transformações geométricas apenas de translação e rotação, também designadas por isometrias, são apresentadas em seguida as matrizes de rotação em torno dos 3 eixos principais e a matriz de translação.

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta & 0 \\ 0 & \sin \theta & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

$$\begin{bmatrix} \cos \theta & 0 & \sin \theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \theta & 0 & \cos \theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

$$\begin{bmatrix} \cos \theta & -\sin \theta & 0 & 0 \\ \sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.3)$$

$$\begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

sendo 2.1, 2.2 e 2.3 as matrizes rotação em torno do eixo  $x$ ,  $y$  e  $z$ , respetivamente, e 2.4 a matriz de translação.

É importante referir que, dado o problema em questão, neste caso apenas se terão translações e rotações em torno do eixo do  $z$ .

## 2.4 Descritores 3D

As *point clouds* podem ter milhares de pontos e, por esta razão, o processo de reconhecimento de objetos pode ser moroso à medida que são adicionadas mais *point clouds* ao sistema. Assim sendo, é importante ter ferramentas que permitam efetuar este reconhecimento mais rapidamente. É nesse sentido que surgem os descritores 3D.

Um determinado objeto, ou cenário, representado por uma *point cloud* pode ser caracterizado através de um conjunto de pontos característicos, normalmente conhecidos por pontos-chave ou *features*. O objetivo de um descritor 3D consiste na identificação inequívoca de um ponto em várias *point clouds*, independentemente do ruído, resolução ou transformações. Este vetor deve conter a mesma informação independentemente da variação da densidade das amostras de dados e deve ser capaz de apresentar os mesmos resultados, ou resultados muito próximos na presença de algum ruído.

Alguns dos descritores 3D conseguem adquirir dados adicionais dos objetos, como por exemplo, o ângulo a que se está a ver o objeto (*viewpoint*), o que permite a extração da pose do objeto.

A grande vantagem dos descritores 3D consiste no facto de se obter um vetor de dimensões bastante reduzidas, quando comparado com a *point cloud* original, e ainda assim ser capaz de conter os pontos-chave que permitam a identificação do objeto.

Estes descritores podem ser classificados em 2 grupos distintos [8]:

- **Descritores locais** - Os descritores locais não têm noção do que é um objeto e são calculados para pontos individuais que são enviados como *input*, ou seja, apenas descrevem como é a geometria local à volta desse ponto.

Estes descritores são utilizados para reconhecimento de objetos e *registration*, ou seja, no alinhamento de duas ou mais *point clouds*.

- **Descritores globais** - Os descritores globais são calculados para um conjunto de pontos que representam o objeto como um todo e não para pontos individuais. Assim sendo, é necessário realizar um passo intermédio chamado de segmentação. A ideia da segmentação consiste dividir a *point cloud* em diferentes partes e, idealmente, cada uma dessas partes corresponderá a um objeto.

Para a resolução do problema em questão, tiveram-se em conta vários descritores 3D, sendo a maioria deles disponibilizados pela biblioteca PCL.

Nesta secção são apresentados os descritores utilizados e testados na identificação das peças.

### Viewpoint Feature Histogram (VFH)

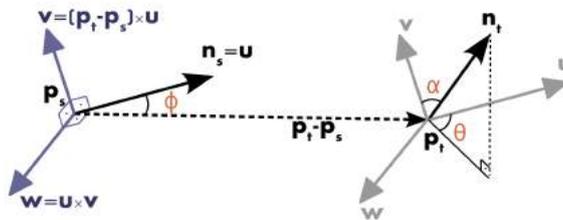
O VFH [9] é um descritor global que codifica a geometria e *viewpoint* da *point cloud*, e baseia-se no descritor local Fast Point Feature Histograms (FPFH) [10] que, por sua vez, é baseado no descritor local Point Feature Histograms (PFH) [11].

De um modo geral, o PFH tenta capturar informações sobre a geometria em torno dos pontos e, para isso, analisa a diferença entre as direções das suas normais e a sua vizinhança.

Considerando um par de pontos  $\langle \mathbf{p}_i, \mathbf{p}_j \rangle$  e as suas normais de superfície estimadas  $\langle \mathbf{n}_i, \mathbf{n}_j \rangle$ , pode-se estimar o conjunto de características dos desvios angulares através das equações definidas em (2.5).

$$\begin{aligned} \cos(\alpha) &= v \cdot \mathbf{n}_j \\ \sin(\phi) &= v \cdot \frac{\mathbf{p}_j - \mathbf{p}_i}{d} \\ \theta &= \arctan(w \cdot \mathbf{n}_j, u \cdot \mathbf{n}_j) \end{aligned} \quad (2.5)$$

onde  $d$  corresponde à distância Euclidiana entre os pontos  $p_i$  e  $p_j$  e  $u$ ,  $v$  e  $w$  representam o sistema de coordenadas do *Darboux frame* [12] que é escolhido para o ponto  $p_i$ , como se pode verificar pela Figura 2.4.

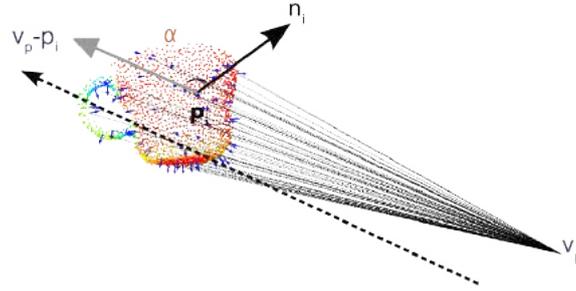


**Figura 2.4:** Representação gráfica do *Darboux frame* e as *features* angulares do PFH. [9]

O VFH pode ser dividido em 2 componentes: [13]:

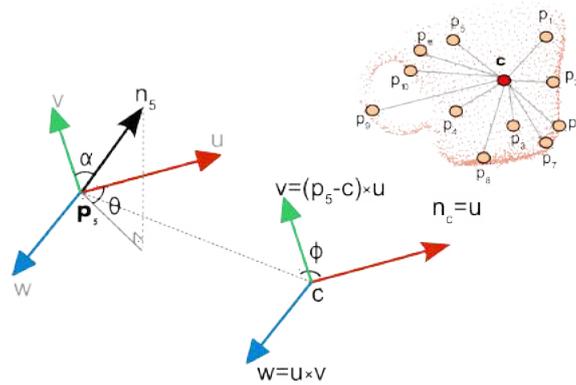
- componente com a direção do *viewpoint*
- componente do FPFH estendido

Para a primeira componente ser obtida, encontra-se o centróide do objeto e, posteriormente, calcula-se o vetor entre a posição do sensor e o centróide do objeto, seguido da sua normalização. No final, é calculado o ângulo entre este vetor e a sua normal para todos os pontos do objeto, como representado na Figura 2.5.



**Figura 2.5:** Cálculo de componente *viewpoint*. [9]

A segunda componente, componente do FPFH estendido, mede os ângulos relativos *roll*, *pitch* e *yaw* entre a direção do ponto vista do ponto central e cada uma das normais da superfície, como apresentado na Figura 2.6.



**Figura 2.6:** Cálculo de componente estendida do FPFH. [9]

Desta forma, o vetor do descriptor VFH é construído através da concatenação do histograma da componente do *viewpoint* e dos 3 histogramas resultantes da componente do FPFH estendido. No final, obtém-se um vetor com 263 posições, sendo 45 para cada  $(\alpha, \theta, \phi)$  e 128 para a componente com a direção do *viewpoint*.

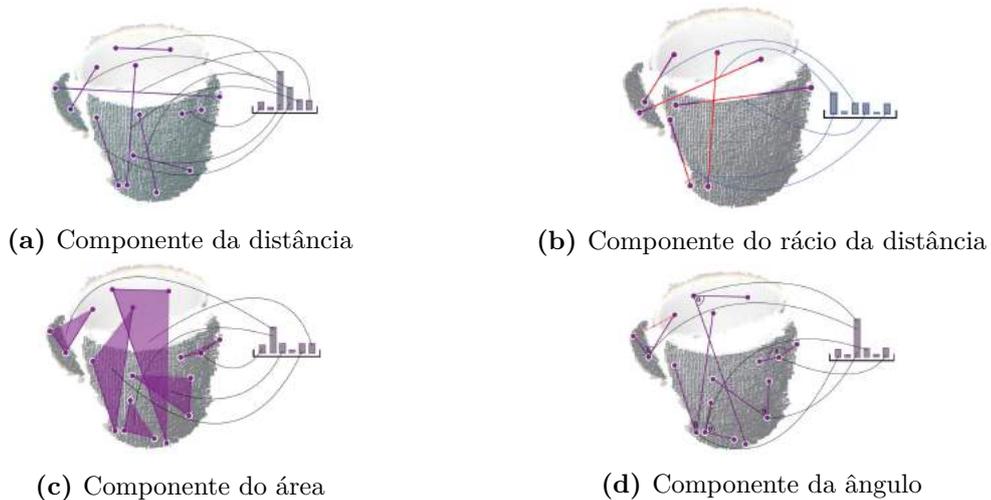
### Ensemble of Shape Functions (ESF)

Este descriptor foi introduzido por Robert Osada *et al.* [14] que sugeriram a caracterização de qualquer modelo 3D poligonal através da utilização de uma função de forma geométrica baseada em 5 medidas:

- **A3** - ângulo entre 3 pontos aleatórios na superfície do modelo 3D;
- **D1** - distância entre um ponto fixo e um ponto aleatório da superfície;
- **D2** - distância entre dois pontos aleatórios da superfície;
- **D3** - raiz quadrada de um triângulo composto por três pontos aleatórios da superfície;
- **D4** - raiz cúbica do volume de um tetraedro composto por quatro pontos aleatórios da superfície.

Alguns anos mais tarde, Walter Wohlkinger e Markus Vincze [15] reduziram o número de medidas para 3, mais concretamente, A3, D2 e D3.

Desta forma, obtém-se um vetor descritor com 640 posições dividido em dez secções distintas onde três delas correspondem à componente do ângulo (D3), três para a componente da área, três para a componente da distância e, finalmente, uma para o rácio da componente da distância, como pode ser observado na Figura 2.7.



**Figura 2.7:** Componentes do descritor ESF. [15]

### Clustered Viewpoint Feature Histogram (CVFH)

O descritor global CVFH [16] foi desenvolvido com o intuito de resolver as limitações do descritor VFH, nomeadamente, a sensibilidade ao ruído e às oclusões, ou seja, partes do objeto em falta.

Desta forma, os autores propuseram dividir o objeto em regiões estáveis através de um algoritmo de segmentação por crescimento de região e, posteriormente, aplicar o VFH a todas as regiões, em vez de aplicar o VFH ao objeto como um todo.

Para além disso, foi adicionada outra componente para efetuar a computação do histograma, mais concretamente, *Shape Distribution Component (SDC)* calculada por (2.6) e apresentada em [16].

$$SDC = \frac{(p_c - p_i)^2}{\max((p_c - p_i)^2)} \quad (2.6)$$

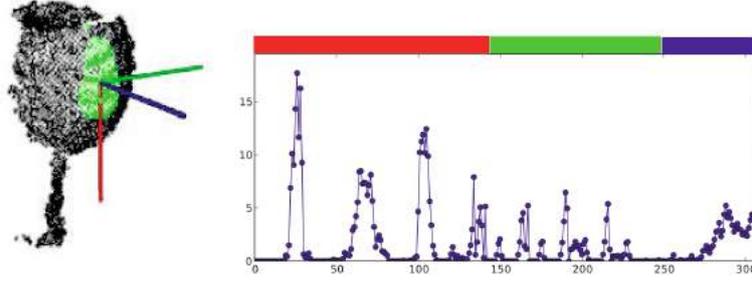
No final, o vetor descritor é dado pela concatenação dos histogramas  $(\alpha, \phi, \theta, SDC, \beta)$ , o que resulta num vetor de 308 posições.

### Oriented, Unique and Repeatable Clustered Viewpoint Feature Histogram (OUR-CVFH)

O descritor OUR-CVFH [17] aparece como uma continuação do CVFH ao qual se acrescenta a computação de um *frame* de referência único.

Este descritor baseia-se na utilização de sistemas de coordenadas repetíveis denominados por Semi-Global Unique Reference Frames (SGURF) que permitem não só remover a invariância à rotação da câmara em torno do ângulo *roll*, como também permitem a extração da pose com 6 graus de liberdade diretamente sem passos adicionais.

Inicialmente, a computação é muito semelhante ao CVFH, no entanto, após a segmentação, os pontos de cada região são novamente filtrados de acordo com a diferença entre as suas normais e a média da normal da região.



**Figura 2.8:** Histograma resultante do descritor OUR-CVFH. [17]

Posteriormente, calcula-se o SGURF para cada uma das regiões e, finalmente, é calculado o vetor descritor OUR-CVFH.

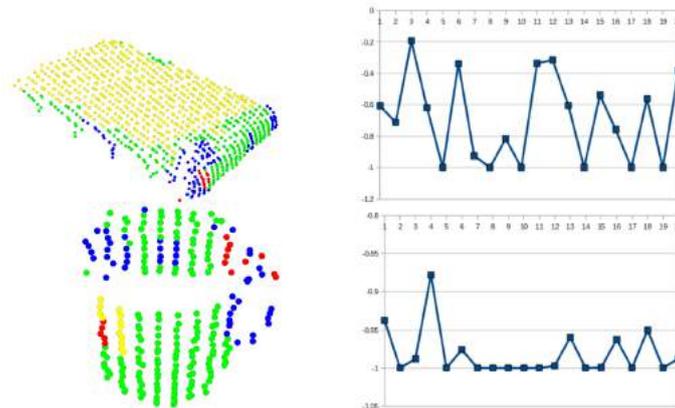
### Global Radius-Based Surface Descriptor (GRSD)

O descritor GRSD [18] é considerado a versão global do descritor local RSD [19] que codifica a relação radial entre cada par de pontos situados numa superfície, definida por um raio  $r$ , onde para cada ponto  $p$  e os seus vizinhos  $p_i$ . A distância  $d_{(\alpha)}$  e o ângulo  $\alpha$  a calcular, são dados por (2.7) [19].

$$d_{(\alpha)} = \sqrt{2r} \sqrt{1 - \cos(\alpha)} \quad (2.7)$$

$$r = \frac{d_{(\alpha)}}{\alpha}$$

Para efetuar o cálculo do descritor GRSD, começa-se por “voxelizar” a *point cloud* e, uma vez que a vizinhança esteja definida em *voxels*, calcula-se o descritor RSD de forma a categorizar as várias superfícies de acordo com a sua categoria geométrica, mais concretamente, planos, cilindros, arestas e esferas de acordo com as regras definidas em [19], como se pode verificar na Figura 2.9.



**Figura 2.9:** Exemplo de categorização das superfícies nas várias classes. [19]

Posteriormente, todo o conjunto é classificado numa das categorias mencionadas anteriormente e é efetuada a computação do descritor GRSD.

### Globally Aligned Spatial Distribution (GASD)

O descritor GASD [20] começa por estimar o *frame* de referência através do método Principal Component Analysis (PCA).

Através de (2.8) [20], começa-se por estimar a origem do *frame* de referência que corresponde ao centróide  $\bar{\mathbf{P}}$  de um determinado conjunto de pontos  $\mathbf{P}_i$  que representa uma vista parcial do objeto em questão.

$$\bar{\mathbf{P}} = \frac{1}{n} \sum_{i=1}^n \mathbf{P}_i \quad (2.8)$$

Posteriormente, é calculada a matriz covariância  $\mathbf{C}$  por (2.9) [20].

$$\mathbf{C} = \frac{1}{n} \sum_{i=1}^n (\mathbf{P}_i - \bar{\mathbf{P}})(\mathbf{P}_i - \bar{\mathbf{P}})^T \quad (2.9)$$

Para se obter o descritor GASD é ainda necessário efetuar o alinhamento do *frame* de referência com o sistema de coordenadas canônicas, sendo este alinhamento efetuado a partir de (2.10) [20].

$$\begin{bmatrix} \mathbf{R} & \mathbf{T} \\ \mathbf{0} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{v}_3^T & -\mathbf{v}_3^T \bar{\mathbf{P}} \\ \mathbf{v}_2^T & -\mathbf{v}_2^T \bar{\mathbf{P}} \\ \mathbf{v}_1^T & -\mathbf{v}_1^T \bar{\mathbf{P}} \\ 0 & 1 \end{bmatrix} \quad (2.10)$$

### Global Orthographic Object Descriptor (GOOD)

O descritor GOOD [21] foi desenvolvido para ser invariante à escala e à pose de modo a obter um reconhecimento de objetos 3D preciso. Para isto foi utilizado um método baseado nos *eigenvalues* e na Principal Component Analysis (PCA), obtendo desta forma um *frame* de referência.

Após a obtenção dos frames de referência, são criadas 3 projeções com base na projeção ortográfica (*XoZ*, *YoZ* e *XoY*), como se pode verificar nas Figura 2.10b. Em seguida, cada uma destas projeções é dividida em *bins*, obtendo assim 3 matrizes de distribuição que representam as diferentes projeções. Posteriormente, calcula-se a entropia e a variância através de (2.11) e (2.12) [21], respetivamente, de cada uma das matrizes. Estes valores são utilizados para definir a ordem de concatenação das matrizes de distribuição num único vetor descritor, como apresentado na Figura 2.10d.

$$H(\mathbf{m}) = - \sum_{i=1}^{n^2} \mathbf{m}_i \log_2 \mathbf{m}_i \quad (2.11)$$

$$\sigma^2(\mathbf{m}) = \sum_{i=1}^{n^2} (i - \mu_{\mathbf{m}})^2 \mathbf{m}_i \quad (2.12)$$

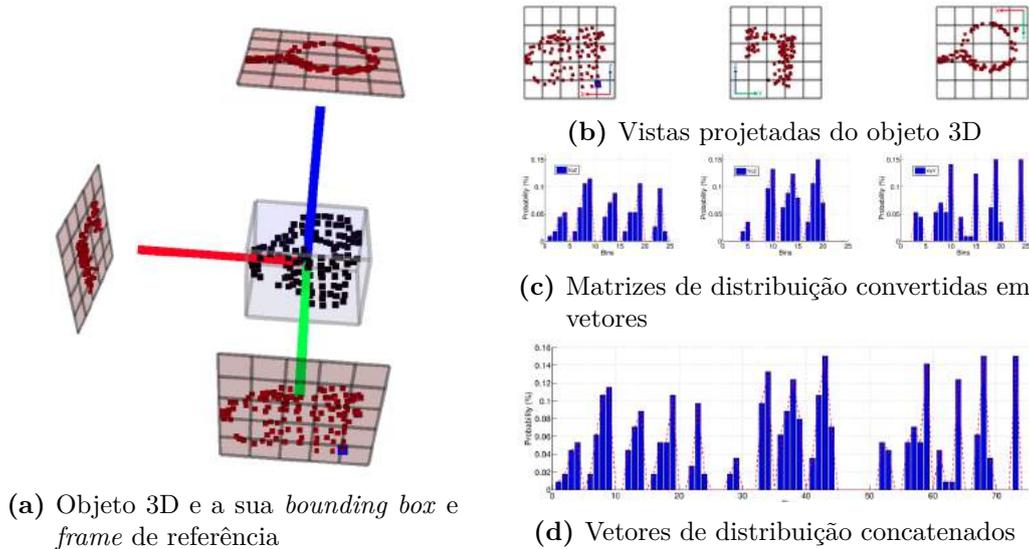
## 2.5 Técnicas de registo

Quando uma *point cloud* é adquirida, muitas vezes é necessário passar por uma etapa fundamental relativa ao seu processamento, denominada por segmentação.

A segmentação de uma *point cloud* pode ser vista como a divisão desta em vários grupos onde, idealmente, cada um deles corresponde a um objeto. Muitas vezes estes grupos são denominados de *clusters*.

Existem várias técnicas de segmentação que podem ser divididas em diferentes classes [22]:

- Métodos baseados nas *edges* dos objetos;
- Métodos baseados em regiões;



**Figura 2.10:** Exemplo ilustrativo do descritor GOOD. [21]

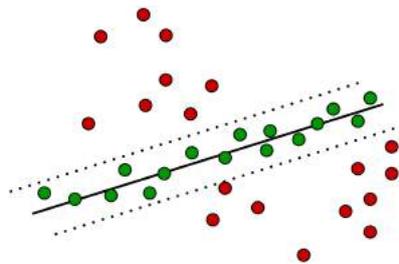
- Métodos baseados em atributos;
- Métodos baseados em modelos;
- Métodos baseados em gráficos.

### 2.5.1 RANSAC

Na presente dissertação dá-se especial atenção a um método baseado em modelos, mais concretamente, ao RANdom SAMple Consensus [23]. Este método, proposto em 1981 por Fischler e Booles, é um método iterativo que permite estimar o modelo matemático de um determinado conjunto de dados que contenha *outliers*. Consideram-se que os *outliers* são os pontos que não se encaixam no modelo, enquanto que os *inliers* são os pontos que encaixam neste.

Contrariamente a muitas das técnicas utilizadas em visão por computador para efetuar estimativas robustas, como por exemplo o método dos mínimos quadrados, o RANSAC foi desenvolvido dentro da comunidade de visão por computador. Algumas das vantagens deste método em relação aos métodos convencionais são o facto de que, por um lado, não existe a necessidade de uma boa estimativa inicial para garantir a convergência do método e, por outro lado, atinge-se a convergência num conjunto de dados onde está presente uma grande percentagem de *outliers*.

A Figura 2.11 representa o caso em que se pretende estimar o modelo de uma reta com o RANSAC.



**Figura 2.11:** Ilustração do RANSAC com os *inliers* a serem representados pelos pontos verdes e os *outliers* pelos pontos vermelhos.

Segundo Konstantinos [24], este método pode-se sintetizar através do Algoritmo 1.

---

**Algoritmo 1** RANSAC

---

- 1: Seleção aleatória do número de pontos necessários para determinar os parâmetros do modelo;
  - 2: Cálculo dos parâmetros do modelo;
  - 3: Determinação do número de *inliers*, ou seja, do número de pontos do conjunto de todos os pontos que se encaixam com uma tolerância  $\epsilon$  predefinida, representada pela linha a tracejado na Figura 2.11;
  - 4: Armazenamento do número de *inliers* estimados para o modelo estimado e, em cada ciclo, estes são comparados com os modelos já armazenados de modo a determinar o modelo com o maior número de *inliers*;
  - 5: Repetição dos passos 1 a 4 até um máximo de N iterações predefinidas;
  - 6: Obtenção dos parâmetros do modelo com maior número de *inliers* no final das N iterações.
- 

### 2.5.2 Iterative Closest Point (ICP)

Para além da segmentação, existe ainda uma técnica também muito importante denominada de *registration* que consiste no alinhamento de *point clouds*.

O Iterative Closest Point [25] é um dos algoritmos mais populares para efetuar este processo e tem como objetivo tentar encontrar a melhor transformação que minimize a distância entre 2 *point clouds*, através da minimização dos erros quadráticos entre estas. Uma vez que este método é iterativo, as correspondências das transformações são reconsideradas à medida que a solução se aproxima do mínimo local do erro.

A principal desvantagem deste algoritmo é o facto de poder convergir para um ótimo local, ou seja, caso a transformação entre as duas *point clouds* não seja suficientemente pequena, o resultado do cálculo da transformação poderá estar incorreto.

Segundo [26], o ICP pode ser sintetizado através do Algoritmo 2.

---

**Algoritmo 2** ICP

---

- 1: Cálculo do centro de massa das *point clouds target* (*point cloud* de referência) e *source* (*point cloud* transformada), dados por  $T_x$  e  $S_y$ ;
  - 2: Translação da *point cloud* transformada para a *point cloud* original através de  $T_x - S_y$ ;
  - 3: Para cada um dos pontos do primeiro conjunto de pontos, determinar o ponto mais próximo do segundo conjunto de pontos;
  - 4: Utilizando os pontos de correspondência obtidos no passo anterior, determinar a transformação relativa entre as duas *point clouds*;
  - 5: Transformação da *point cloud* transformada em coordenadas locais da *point cloud* de referência;
  - 6: Através da soma das distâncias quadráticas entre os pontos correspondentes, é calculado o erro quadrático médio;
  - 7: Repetir passos 3 a 6 até que o valor do erro quadrático seja inferior a um determinado valor de *threshold* ou até que as N iterações máximas sejam atingidas.
- 

## 2.6 ROS

Como foi referido anteriormente, no final do projeto ir-se-á utilizar o ROS para efetuar a interligação e sincronização de todo o sistema e, desta forma, a presente secção pretende

efetuar uma introdução a esta ferramenta. Apesar de já existir a versão 2 do ROS, nesta secção apenas se irá descrever a versão 1 uma vez que será nesta versão que se irá fazer toda a interligação do sistema.

O ROS é um *framework* para desenvolvimento de software para o mundo da robótica e, apesar de já terem sido desenvolvidas várias distribuições [27], as únicas que se encontram disponíveis atualmente, são:

- **Melodic Morenia** lançada em 2018 e desenvolvida, principalmente, para Ubuntu 18.04, apesar de ser suportada por outros sistemas operativos, como por exemplo, Windows, Mac OS e Android;
- **Noetic Ninjemys** lançada em 2020 e desenvolvida, principalmente, para Ubuntu 20.04.

Devem ser tidos em conta alguns conceitos base relativos à estrutura de comunicação ROS:

- *node* - Um nó é um processo/programa com uma determinada função que comunica com outros nós através de tópicos e/ou serviços;
- *master* - O ROS *master* fornece serviços de nomenclatura e registo para os restantes nós do sistema. O papel do *master* é permitir que os restantes nós se localizem e, uma vez que o façam, comecem a fazer comunicação ponto a ponto. Os nós comunicam com o *master* para reportar a sua informação de registo e, como estes comunicam com o *master*, podem receber informação sobre outros nós que estejam registados e, desta forma, fazem as conexões apropriadas. O *master* também disponibiliza um servidor de parâmetros;
- *topic* - Os tópicos podem ser vistos como barramentos sobre os quais os nós trocam mensagens e permitem o fluxo unidirecional entre os vários nós;
- *message* - As mensagens são estruturas de dados trocadas entre nós através de tópicos, serviços e ações e a descrição destas é armazenada em ficheiros com extensão “.msg”. Estas mensagens podem conter um tipo de dados simples, como por exemplo um inteiro ou uma *string*, assim como tipos de dados mais complexos, como por exemplo estruturas de dados.
- *service* - Os serviços são o mecanismo para interações do tipo *request/reply* e por isso, são definidos por um par de estruturas de mensagens: uma para *request* e outro para *response*. Os serviços são definidos em ficheiros com a extensão “.srv”;
- *parameter server* - Este servidor é utilizado pelos *nodes* para armazenar e devolver parâmetros durante a execução dos vários *nodes* ROS e é executado dentro do ROS *master*. O *parameter server* pode ser visto como um dicionário compartilhado que pode ser acessado através de APIs de rede e que contém variáveis globais que são acessíveis de qualquer lado no ambiente ROS atual. Essas variáveis, são designadas por parâmetros ROS;
- *bags* - Os *bags* são ficheiros ROS com extensão “.bag” que permitem guardar e reproduzir dados de mensagens ROS. Estes ficheiros podem ser editados e reproduzidos através da ferramenta *rosvbag*.

### 2.6.1 Ferramentas de desenvolvimento

De modo a facilitar a sua utilização, o ROS permite a utilização de várias ferramentas de desenvolvimento, sendo apresentadas em seguida duas delas.

## RViz

O RViz<sup>1</sup> é uma ferramenta de visualização 3D do ROS e o seu principal objetivo é a visualização de dados, quer sejam 2D ou 3D, provenientes de tópicos e parâmetros ROS. Esta ferramenta permite ainda visualizar dados de modelos de robôs, como por exemplo, a posição das juntas e as suas transformações 3D, a visualização de dados provenientes de diversos sensores, como por exemplo, uma câmara RGB, uma câmara RGBD, e ainda a visualização de trajetórias determinadas pelos planeadores para os manipuladores.

## rqt

O rqt<sup>2</sup> é um *framework* do ROS, baseado em Qt<sup>3</sup>, utilizado para desenvolver interfaces gráficas, desenvolvidas como *plugins*, que permitem a visualização de tópicos, ligações entre nós, entre outros.

Alguns dos *plugins* disponíveis atualmente permitem para visualização de tópicos, ações e serviços (`rqt_topic`, o `rqt_action` e `rqt_graphics`, respetivamente), bem como o `rqt_bag` e o `rqt_image_view` que permitem a visualização de ficheiros “.bag” e a visualização de imagens, respetivamente.

### 2.6.2 Transformações ROS

Tipicamente, num sistema robótico existem vários *frames* de coordenadas 3D que podem variar ao longo do tempo como, por exemplo, a *frame* do “mundo” em que o sistema se encontra (*world*), a *frame* da base do robô (*base\_link*), a *frame* da garra do robô, entre outras (Figura 2.12). A biblioteca `tf` foi projetada para fornecer aos utilizadores um procedimento padrão que permita controlar todos os sistemas de coordenadas dos *frames* ao longo do tempo [28], permitindo desta forma transformar coordenadas entre os diferentes referenciais. Isto é possível uma vez que existe uma relação bem explícita entre os *frames* do sistema.

Na Figura 2.12 é possível visualizar todos os *frames* presentes no robô UR10, sendo os eixos do X, Y e Z de cada *frame* representados pelos cilindros vermelhos, verdes e azuis, respetivamente.

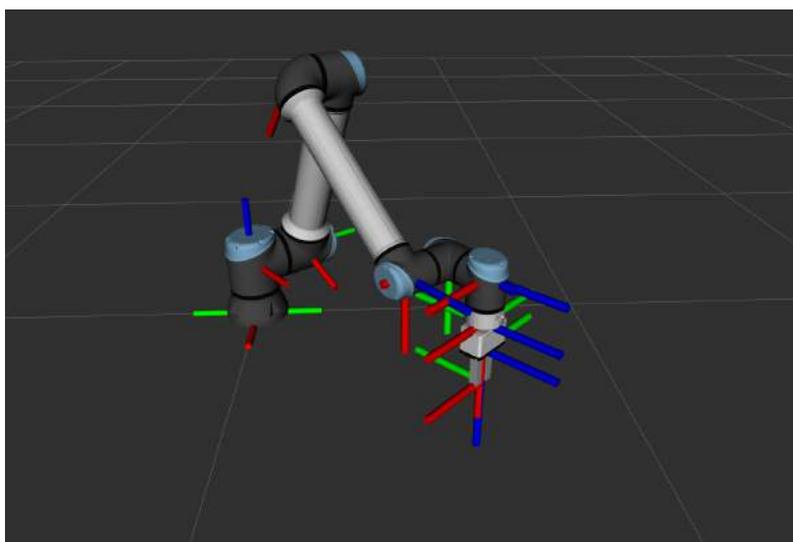


Figura 2.12: Visualização de todos os *frames* tf do robô UR10.

---

<sup>1</sup><http://wiki.ros.org/rviz>

<sup>2</sup><http://wiki.ros.org/rqt>

<sup>3</sup><https://www.qt.io/>

### 2.6.3 Simulador Gazebo

O Gazebo é um simulador 3D *open-source* com uma grande capacidade para simular robôs, quer em ambientes interiores, quer em ambientes exteriores e foi desenvolvido pela *Open Robotics*.

Este simulador utiliza mecanismos externos de renderização *Open Source 3D Graphics Engine* (OGRE) o que leva a uma boa fidelidade gráfica nas simulações. Para além disso, o Gazebo apresenta vários mecanismos de física de elevado desempenho, nomeadamente, *Open Dynamics Engine* (ODE), *Dynamic Animation and Robots Toolkit* (DART), Bullet e Simbody. Na Figura 2.13 é apresentada a janela inicial deste simulador.

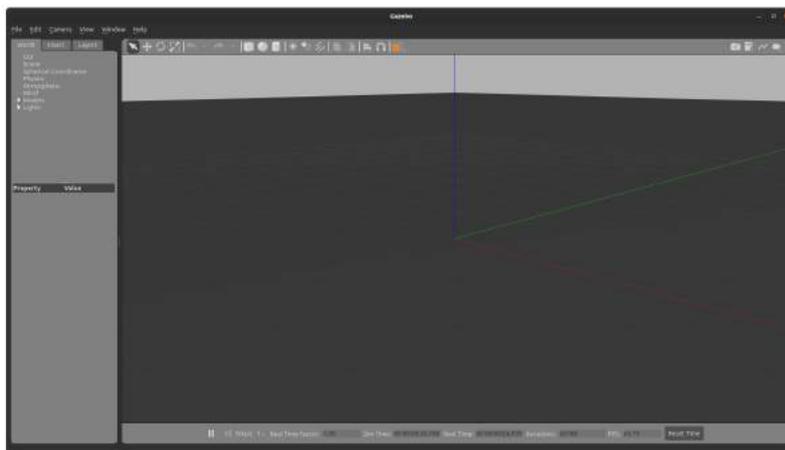


Figura 2.13: Interface gráfica do simulador Gazebo.

## 2.7 Trabalhos relacionados

Nesta secção são apresentados alguns trabalhos desenvolvidos e relacionados com a identificação de objetos 3D a partir de uma *point cloud*.

Hamidreza Kasaei *et al.* [29] apresentaram uma abordagem de aprendizagem e reconhecimento 3D baseado em instâncias, para um robô de serviço autónomo num ambiente de restaurante. Este método depende de um descritor de formas local conhecido como *spin image* [30]. Neste trabalho tiveram-se em atenção duas questões importantes:

1. Como efetuar a deteção, conceção e reconhecimento de objetos em cenas 3D não estruturadas?
2. Como efetuar a aquisição e a utilização de conhecimentos de alto nível obtidos pelo utilizador para melhorar o desempenho do sistema?

No desenvolvimento do trabalho, começaram por efetuar pré-processamento para eliminar informação desnecessária e, desta forma, preparar uma *point cloud* adequada. Em seguida, detetaram candidatos a objetos, através de *clustering* e depois, todos os candidatos são descritos com base no descritor *spin image*. Por fim, utilizaram a regra de classificação da vizinhança mais próxima para atribuir *labels* aos objetos detetados e os resultados mostram que esta abordagem tem um bom desempenho em diferentes tipos de objetos.

Luís Alexandre [31] apresentou um trabalho onde se fazia comparação e avaliação de alguns descritores disponibilizados pela PCL. No final do trabalho desenvolvido, as principais conclusões retiradas foram: é possível melhorar os resultados do reconhecimento através do aumento do número de pontos chave em detrimento do tamanho e do tempo; devem ser utilizados descritores que usem a informação da cor em vez de utilizar apenas informação sobre a forma dos objetos; e, finalmente, o descritor SHOTCOLOR [32]

apresenta um bom balanço entre os resultados do reconhecimento e a sua complexidade temporal;

Jay Tang *et al.* [33] propôs um sistema para reconhecimento e localização de objetos em cenas desorganizadas, utilizando para tal dados RGB-D. Este sistema permite lidar com oclusões, alterações de iluminação, múltiplos objetos e múltiplas instâncias dos mesmo objeto;

Kevin Lai *et al.* [34] apresentou um *dataset* e propôs um método onde sugere *kernels* baseados em informação de profundidade, sendo este método denominado por *Hierarchical Kernel Descriptors*. O *dataset* contém 300 objetos organizados em 51 categorias. No trabalho desenvolvido, são introduzidas técnicas de reconhecimento e detecção de objetos baseados em dados RGB-D e, no final, é possível verificar que a combinação de informações de cor e profundidade melhoram a qualidade dos resultados.

## 2.8 Conclusão

Ao longo deste capítulo foram apresentadas várias tecnologias com vista à resolução do problema em estudo. No que diz respeito às bibliotecas 3D, todas as bibliotecas foram utilizadas sendo a PCL a mais utilizada ao longo do desenvolvimento do projeto. Quanto à identificação das peças 3D, foram abordados e estudados vários descritores 3D, sendo os resultados obtidos por eles apresentados no capítulo seguinte. Relativamente ao problema de estimar a pose das peças, abordaram-se métodos como o RANSAC e o ICP. Finalmente, foi apresentado o ROS, bem como algumas das suas ferramentas, uma vez que, no final, o projeto será baseado neste *framework*.



## Capítulo 3

# Metodologia e resultados no reconhecimento de peças 3D

Neste capítulo é apresentado o trabalho desenvolvido para o subsistema de identificação, desde a parte mais embrionária, como por exemplo a escolha dos sensores, até à fase final onde é feita uma correta identificação das peças 3D. Este capítulo inicia-se com a apresentação da arquitetura proposta para o subsistema em questão, seguida do desenvolvimento de um protótipo do mesmo no simulador Gazebo. De seguida, são apresentados os cálculos efetuados para a escolha do sensor 3D. Uma vez que o sensor escolhido não iria chegar a tempo para a realização da dissertação, a JPM emprestou o sensor 3D previamente adquirido por eles. Assim sendo, é apresentado todo o estudo e testes realizados para o correto funcionamento do mesmo para que fosse possível realizar uma boa aquisição de dados. Posteriormente é apresentado todo o processo de identificação das peças 3D, bem como os resultados obtidos.

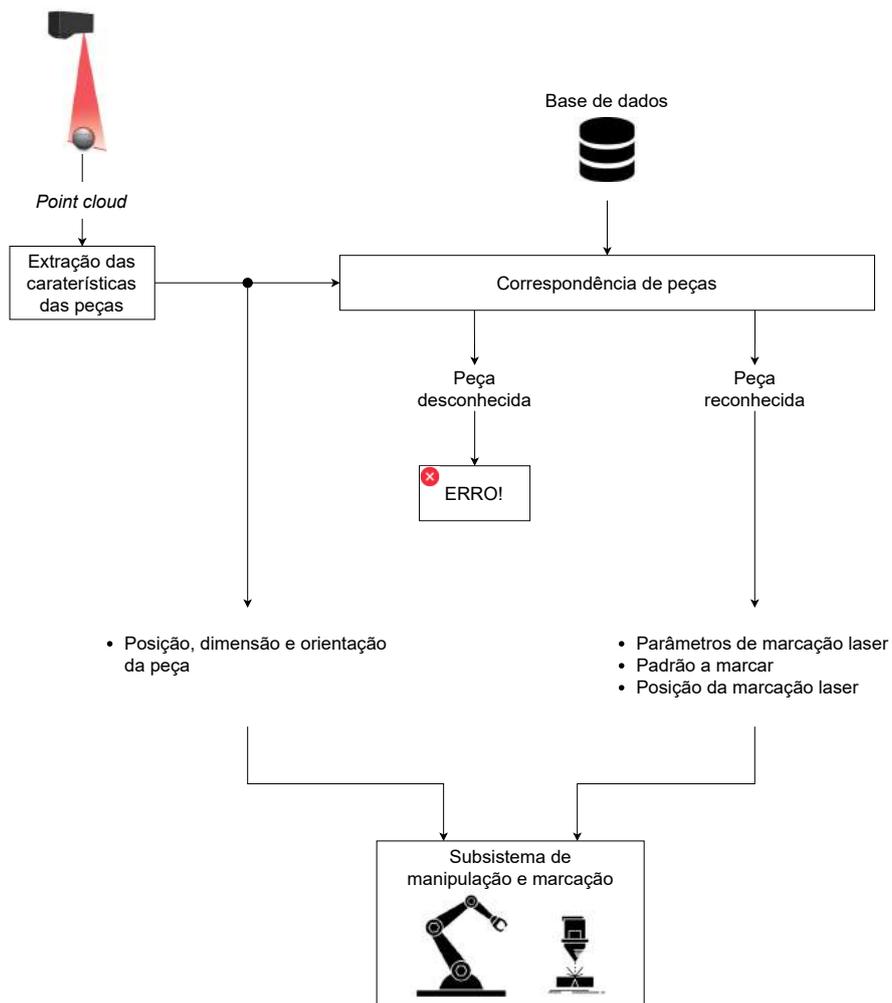
### 3.1 Arquitetura proposta

Na Figura 3.1 é apresentado o esquema geral do subsistema de identificação. Neste subsistema, adquire-se a *point cloud* através do sensor e, posteriormente, os dados são passados por um conjunto de algoritmos onde são extraídas as características das peças. De seguida, é feita a comparação dessas características com o que foi previamente guardado na base de dados e, caso a peça em questão se encontre na base de dados, são enviados os dados para o subsistema de manipulação e marcação laser, nomeadamente, o ID da peça, a sua localização e orientação no tapete, as coordenadas para a gravação da mesma e os parâmetros de marcação do laser. Caso contrário, ou seja, caso a peça não se encontre na base de dados e, desta forma, não seja reconhecida, é gerado um erro.

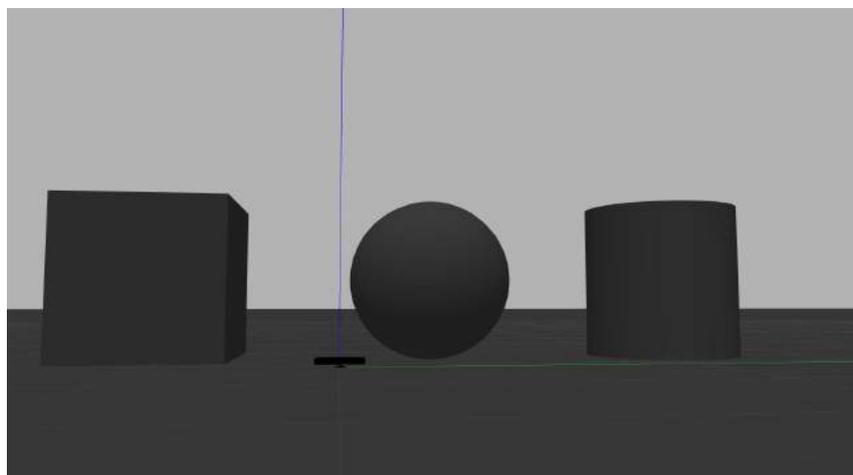
### 3.2 Simulação do subsistema de identificação

Com o intuito de progredir no desenvolvimento do projeto enquanto não se estava na posse do sensor, utilizou-se o simulador Gazebo para criar um ambiente de simulação e, desta forma, simular parte do subsistema de visão. Este ambiente criado (Figura 3.2) inclui um *plugin* de uma câmara de profundidade ROS e o objetivo passa por conseguir distinguir três objetos a partir da sua representação em *point cloud*:

- cubo
- cilindro
- esfera



**Figura 3.1:** Esquema geral do subsistema de identificação.



**Figura 3.2:** Ambiente de simulação desenvolvido.

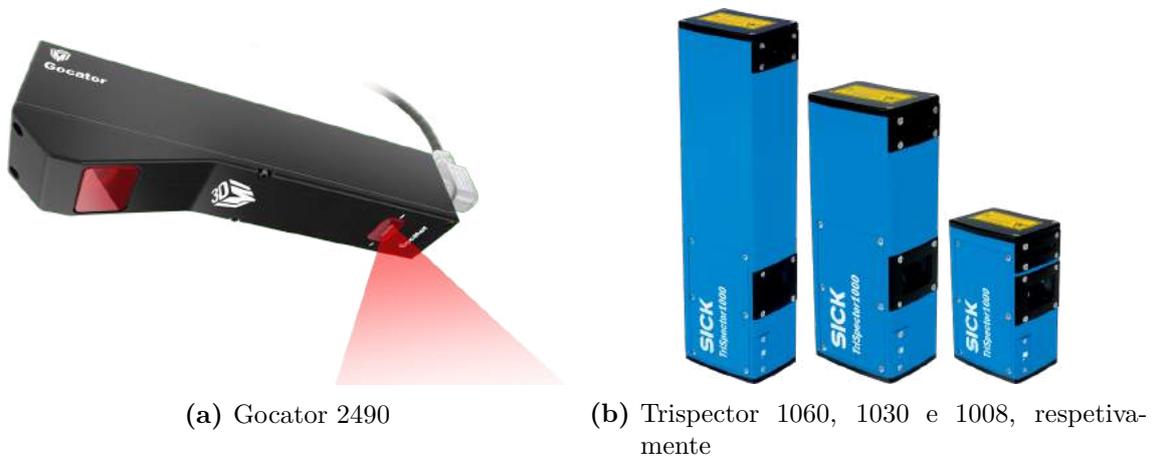
É importante referir ainda que estas simulações trouxeram grandes vantagens ao desenvolvimento do projeto, uma vez que foi possível ganhar mais experiência com o ROS, bem como com a PCL.

### 3.3 Sensorização

De modo a escolher o sensor mais adequado para a realização deste projeto, fez-se um estudo para a escolha do mesmo tendo em atenção os seguintes requisitos definidos pela JPM:

- Tipos de geometria de peças: chapas, peças trabalhadas quadradas e redondas
- Dimensões mínimas das peças: 5 mm x 5 mm x 15 mm (largura x comprimento x altura)
- Dimensões máximas das peças: 400 mm x 400 mm x 150 mm (largura x comprimento x altura)
- Velocidade máxima de marcação: 1200 mm/sec
- Afastamento mínimo entre as peças: 50 mm
- Cadência mínima requerida: 7,5 produtos por minuto

Com o objetivo de cumprir os objetivos definidos anteriormente, consideraram-se os sensores presentes na Figura 3.3, visto apresentarem características capazes de resolver o problema em mãos.



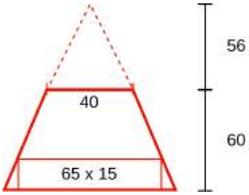
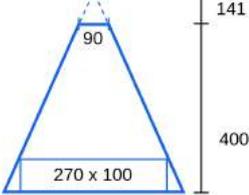
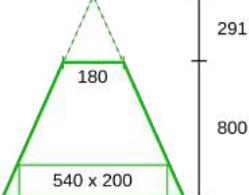
**Figura 3.3:** Sensores 3D.

É importante também referir que o sensor apresentado na Figura 3.3a já tinha sido escolhido pela empresa e, desta forma, na presente dissertação apenas foram efetuados cálculos para os restantes 3 sensores.

Na Tabela 3.1 são apresentados alguns dos parâmetros tidos em atenção para a aquisição do sensor, sendo os valores de (a) e (b), obtidos através das equações 3.1 e 3.2, respetivamente.

$$a = \frac{2 \text{ mm}}{\text{Height resolution}} \quad (3.1)$$

$$b = \frac{2 \text{ mm}}{\text{3D Profile resolution}} \quad (3.2)$$

	TriSpector1008	TriSpector1030	TriSpector1060
Typical FoV (mm)			
Height resolution	[20 ... 50] $\mu\text{m}$	[40 ... 280] $\mu\text{m}$	[80 ... 670] $\mu\text{m}$
3D profile resolution	0.049 mm/px	0.215 mm/px	0.43 mm/px
For a tolerance of 2 mm (a)	[100 ... 40] $\mu\text{m}/\text{px}$	[50 ... 7] $\mu\text{m}/\text{px}$	[25 ... 3] $\mu\text{m}/\text{px}$
Number of pixels corresponding to 2 mm (b)	40 px	9 px	4px

**Tabela 3.1:** Parâmetros para a escolha do sensor.

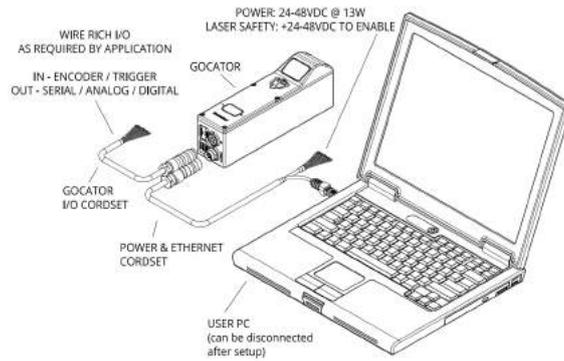
Através da análise da Tabela 3.1, é fácil perceber que, devido às especificações da empresa, o TriSpector1008 é logo descartado uma vez que apenas é adequado para peças de, no máximo, 60 mm de altura. O TriSpector1030 seria uma escolha mais aceitável uma vez que já se poderiam ter peças de maior dimensão. No entanto, devido aos requisitos impostos pela JPM, continuaríamos com o problema de não conseguirmos trabalhar com as peças de maior dimensão. Assim sendo, o sensor mais adequado é o TriSpector1060, uma vez que se consegue cumprir o requisito da altura máxima das peças. No entanto, é importante ter em atenção que com a escolha deste sensor, perde-se resolução. Por exemplo, comparativamente com o TriSpector1030 fica-se com apenas metade da resolução. Desta forma foram efetuados cálculos para determinar qual a distância mínima a que se podia colocar o TriSpector1060 do tapete, sendo esta de aproximadamente 646 mm.

### 3.3.1 Sensor utilizado na dissertação

Uma vez que o sensor escolhido anteriormente iria demorar a chegar ao IRIS, a JPM decidiu ceder o sensor por eles adquirido durante a fase de desenvolvimento da presente dissertação e, desta forma, possibilitar o avanço na realização do projeto em questão. Na presente secção irá fazer-se uma apresentação do funcionamento da Gocator 2490, sensor adquirido pela JPM, bem como algumas das suas funcionalidades [35].

#### Ligações do sensor

Antes de o sensor ser configurado, deve-se garantir que as ligações apresentadas na Figura 3.4 são corretamente efetuadas.



**Figura 3.4:** Conexões da Gocator. [36]

Como se pode verificar na Figura 3.4, o sensor liga-se ao computador através da porta Ethernet. Este sensor pode ser ligado a dispositivos, tais como fotocélulas, PLCs ou encoders, sendo este último utilizado para o desenvolvimento da presente dissertação.

### Interface Web

Os sensores Gocator tanto podem ser configurados através de kits de desenvolvimento fornecidos, como através de um browser por meio de uma ligação Ethernet. Para isso, é necessário alterar algumas definições de rede do PC, nomeadamente, desativar a ligação DHCP e estabelecer um endereço IP estático, bem como a máscara da sub-rede. Todos os sensores Gocator são configurados com o endereço IP *default* 192.168.1.10 e introduzindo este endereço no browser, é apresentada a interface Web, presente na Figura 3.5, que apresenta as seguintes funcionalidades:

1. *Manage*: contém configurações para o *layout* do sistema de sensores, rede, movimento e alinhamento, manipulação de *jobs* e manutenção do sensor;
2. *Scan*: contém configurações para o modo de scan (*video*, *profile*, *surface*), o *trigger*, configurações detalhadas do sensor e respetivo alinhamento;
3. *Model*: permite fazer o *matching* de objetos;
4. *Measure*: contém ferramentas de medição integradas;
5. *Output*: contém definições para a configuração de protocolos de saída para comunicação entre o sensor e dispositivos externos;
6. *Dashboard*: fornece monitorização do estado do sensor, bem como estatísticas de medição;
7. *CPU Load and Speed*: fornece informações importantes sobre o sensor, nomeadamente, a carga do CPU e a velocidade do sensor;
8. *Toolbar*: permite controlar a operação do sensor, gerir os ficheiros “.job” e filtrar e reproduzir dados previamente gravados;
9. *Configuration Area*: permite fazer a configuração dos modos de scan;
10. *Data viewer*: Permite fazer a visualização dos dados recolhidos pelo sensor;
11. *Status bar*: Apresenta mensagens de erros, avisos e outras informações que sejam relevantes durante a utilização do sensor, bem como informações sobre o *frame*, quer o sensor esteja a fazer leituras ou a apresentar dados guardados. Para além disto, também permite alterar a linguagem da interface e entrar em modo *Quick Edit*.

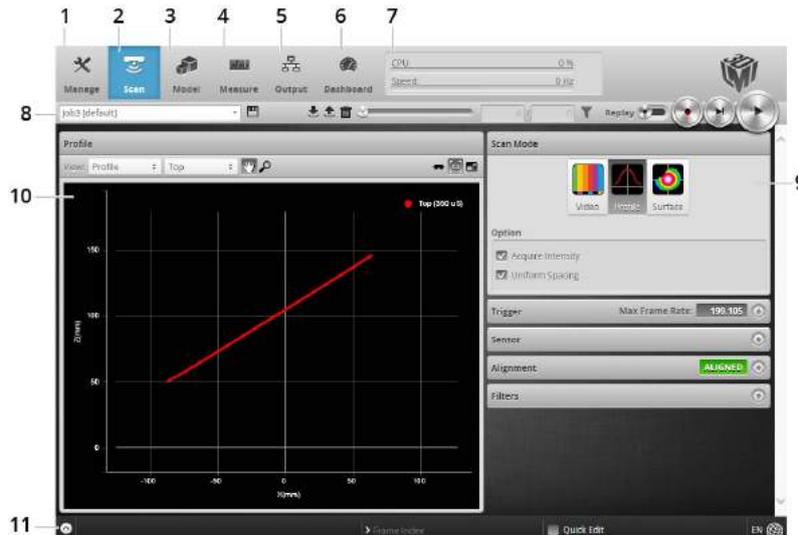


Figura 3.5: Interface Web. [36]

### 3.3.2 Encoder utilizado na dissertação

De modo a conseguir-se saber o quanto a peça se movimentou em cima do tapete, recorreu-se à utilização de um encoder incremental. Para a presente dissertação, foi utilizado o encoder E6B2-CWZ1X [37] da Omron com uma resolução de 2000 pulsos/rotação.

Para se obter a resolução do encoder com maior precisão, optou-se por utilizar a propriedade disponibilizada pelo sensor que consiste no alinhamento deste através de um disco, de dimensões conhecidas, disponibilizado pelo fabricante (Figura 3.6).

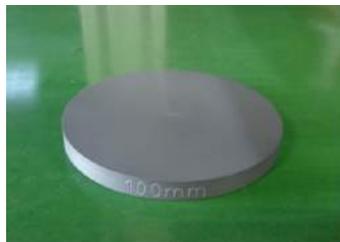


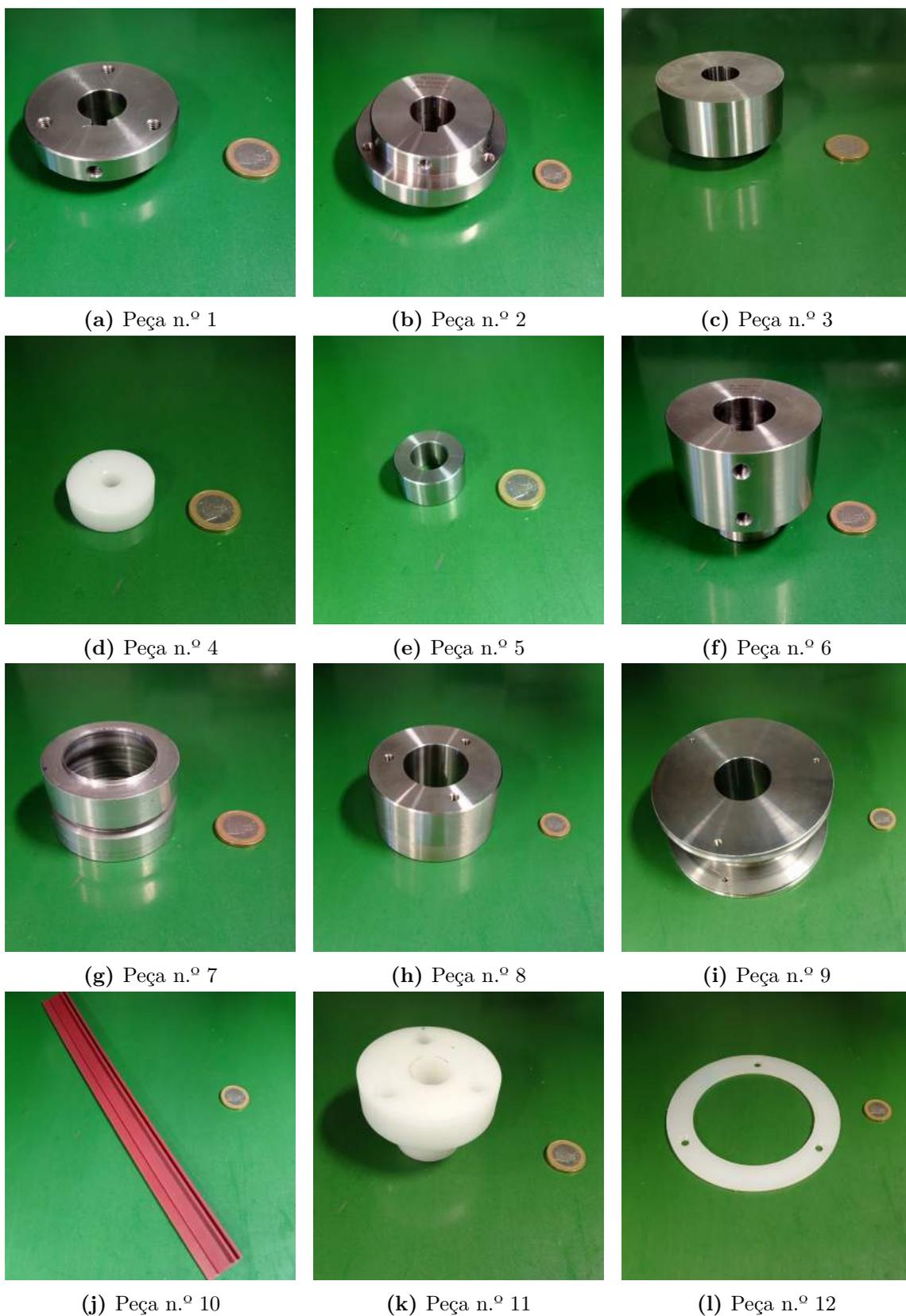
Figura 3.6: Disco utilizado para a calibração do encoder.

## 3.4 Peças de teste utilizadas na dissertação

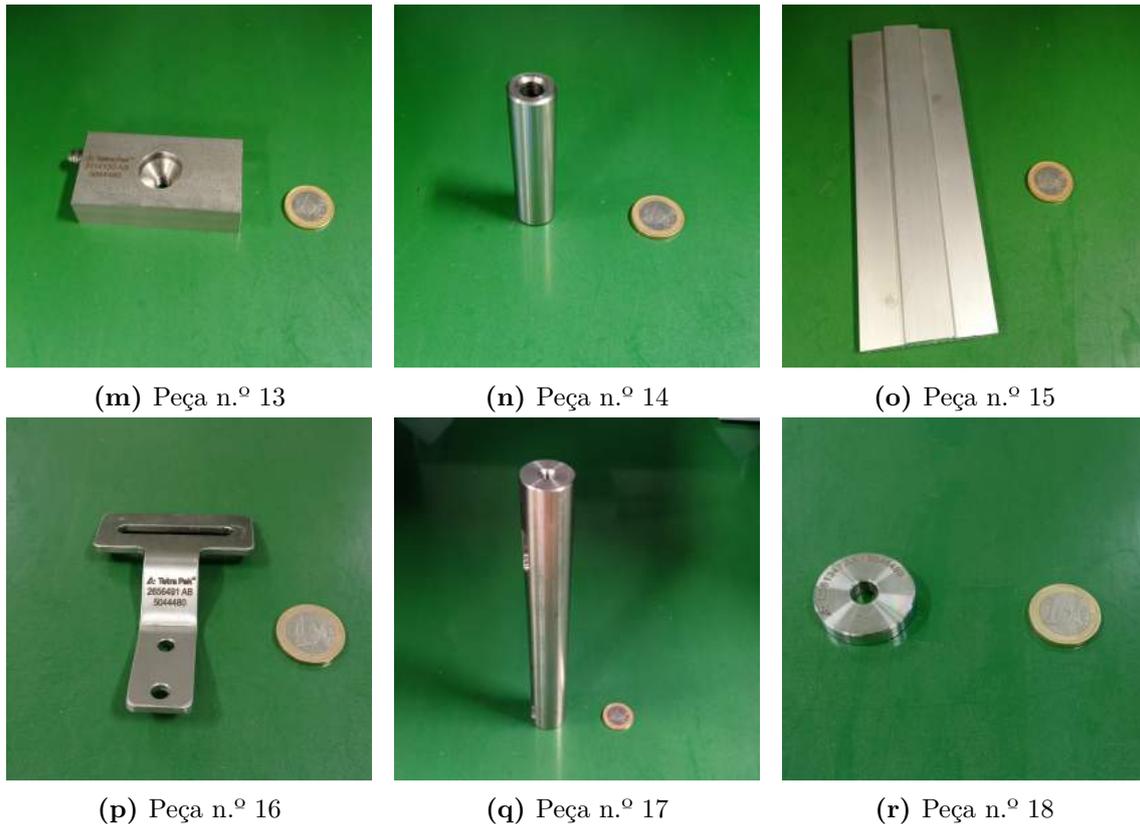
Relativamente às peças que se pretendem identificar, estas apresentam diferentes formatos, dimensões e materiais. Para facilitar as referências às peças, atribuiu-se um identificador numérico de 1 a 18 e, ao longo deste capítulo, todas as referências feitas às mesmas seguirão a nomenclatura apresentada na Figura 3.7, onde são apresentadas algumas peças de teste utilizadas nesta dissertação.

## 3.5 Aquisição da *point cloud*

Depois de se receber o sensor 3D emprestado pela JPM no IRIS, foram feitas várias experiências com o intuito de ajudar na familiarização com o mesmo e, desta forma, conseguir-se extrair corretamente a *point cloud* de todas as peças. Começou-se por extrair o perfil das peças detetadas pelo sensor e, posteriormente, extraiu-se a superfície das peças que são detetadas pelo mesmo.



**Figura 3.7:** Peças usadas durante o desenvolvimento da dissertação.



**Figura 3.7:** Peças usadas durante o desenvolvimento da dissertação.

Para efeitos de estudo, começaram-se por fazer todas as configurações necessárias ao correto funcionamento do sensor através da Interface Web, bem como o estudo dos tempos de exposição da câmara do sensor para os diferentes tipos de material disponíveis. Para além disso, foram ainda testados vários posicionamentos do sensor ao longo do suporte, bem como diferentes inclinações, de modo a extrair os dados com a melhor qualidade possível. É de salientar que cada vez que se alterava o posicionamento do sensor, todas as configurações tinham que ser refeitas, tal como todos os testes de exposição para os diferentes tipos de material.

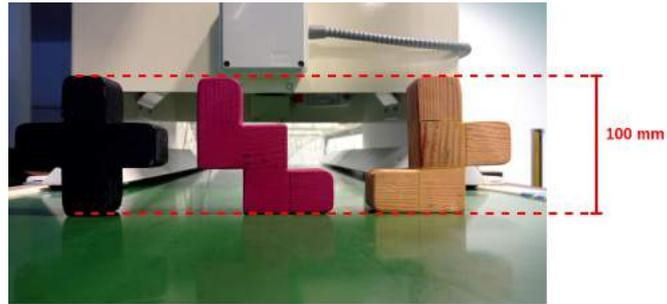
### 3.5.1 Extração do perfil

Enquanto não se estava na posse das peças apresentadas na Figura 3.7, foram utilizadas as peças apresentadas na Figura 3.8 como prova de conceito, sendo os resultados obtidos através da interface Web e do programa desenvolvido para extração do perfil são apresentados na Figura 3.9.

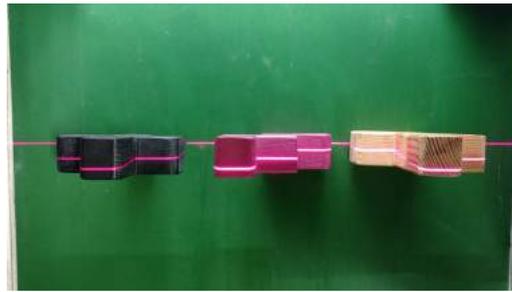
A altura das peças apresentadas na Figura 3.8a é de 100 mm. Nos resultados obtidos através da interface Web e do programa desenvolvido, é possível verificar que os dados adquiridos medem aproximadamente 101 mm, ou seja, existe um erro relativo percentual de 1%.

### 3.5.2 Extração da superfície

No final da extração do perfil, receberam-se as peças de teste apresentadas na Figura 3.7, enviadas pela JPM. Com base nesta figura, é possível verificar que os materiais das peças a identificar são muito diferentes e, como se verá em seguida, isto condiciona bastante o trabalho a ser desenvolvido.

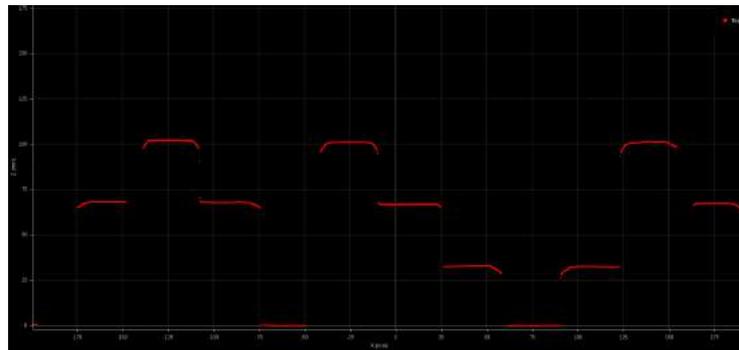


(a) Vista frontal

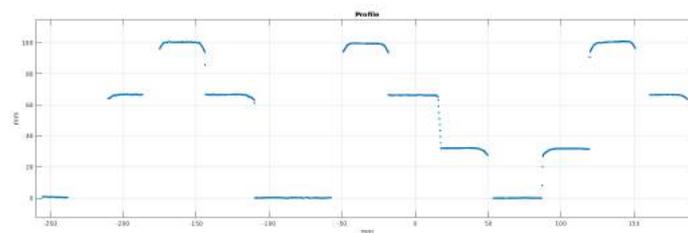


(b) Vista superior

**Figura 3.8:** Exemplificação da aquisição do perfil.



(a) Perfil obtido através da interface web



(b) Perfil obtido através do programa desenvolvido

**Figura 3.9:** Resultados da extração do perfil.

As diferentes características dos materiais em termos da reflexão do feixe na sua superfície, levaram a que uma parte significativa do trabalho desta dissertação tenha sido encontrar um tipo de exposição, entre aqueles disponibilizados pelo sensor, capaz de adquirir as *point clouds* dos diferentes materiais e também a realização de vários testes de posicionamento do sensor para tentar minimizar as reflexões e, com isto, extrair *point clouds* o mais completas possível.

Uma vez que se está na presença de vários tipos de materiais, começou-se por fazer um estudo sobre os vários tipos de exposição, apresentados a seguir, para os diferentes tipos de material, bem como diferentes tempos de exposição.

Tipos de exposição:

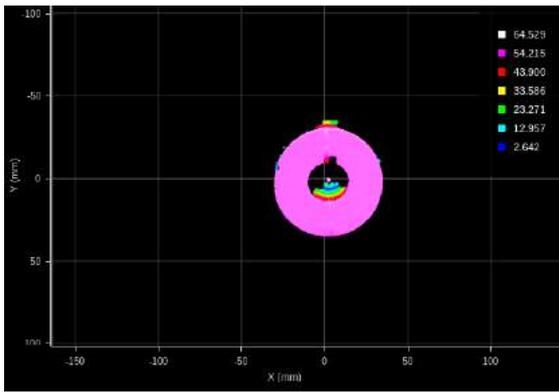
- Simples - como o próprio nome indica, é utilizado um único tempo de exposição para todos os objetos. Normalmente este tipo de exposição é utilizado quando as superfícies a serem adquiridas têm uma superfície uniforme e é igual para todos os objetos a serem scaneados;
- Dinâmica - neste tipo de exposição, o sensor utiliza automaticamente informações de perfis extraídos anteriormente para ajustar a exposição para exposições posteriores e, desta forma, conseguir o melhor perfil. Normalmente, este tipo de exposição é utilizado quando a superfície a ser lida apresenta variações de scan para scan;
- Múltipla - com a exposição múltipla o sensor combina dados de múltiplas exposições, criando desta forma um único perfil laser. Este tipo de exposição pode ser utilizado com o intuito de aumentar a capacidade de scan de materiais claros e escuros que estão simultaneamente no campo de visão do sensor e, para cada exposição, será feito um scan completo ao *frame rate* atual. Desta forma, o *frame rate* efetivo será inferior. Nesta configuração podem ser adicionados até cinco tempos de exposição diferentes e, neste caso, a velocidade de processamento dos dados será um quinto do *frame rate* da exposição simples.

Numa primeira fase, concluiu-se que para se obter uma boa aquisição das *point clouds*, os tempos de exposição teriam que ser diferentes para os vários tipos de material. É necessário ter em atenção que, para esta abordagem funcionar, seria necessário adicionar outra câmara ao sistema de modo a identificar o tipo de material e, posteriormente, teria que se configurar o sensor com o correto tempo e modo de exposição. No entanto, apesar de ser possível identificar rapidamente o tipo de material, o mesmo não acontece com a configuração do sensor e visto que estamos na presença de um sistema de tempo real, o sistema iria ficar demasiado lento e iria contra um dos objetivos do projeto que consiste em tornar o sistema de alta cadência.

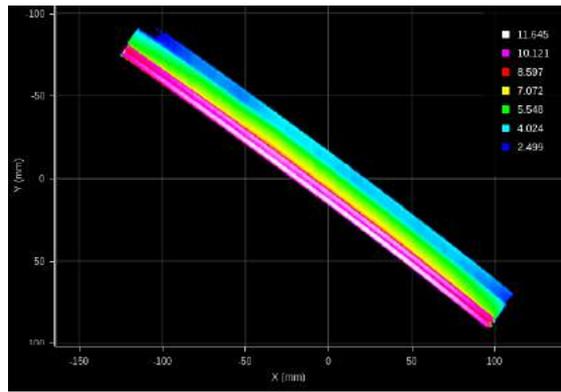
Nas figuras 3.10, 3.11 e 3.12 são apresentados alguns dos testes efetuados para o tipo de exposição simples, dinâmica e múltipla, respetivamente, já com o sensor no seu posicionamento final, ou seja, a 10°. Através destas figuras, é possível visualizar alguns dos problemas que apareceram para identificação das peças, nomeadamente a aquisição de *point clouds* incompletas para peças do tipo n.º 12 (3.10c, 3.11c e 3.12c). Uma possível causa para estes problemas acontecerem, pode ser o tipo de material da peça em questão e também o facto da peça ser muito fina, o que leva a que o laser do sensor consiga atravessar o material.

### 3.5.3 Configurações finais do sensor

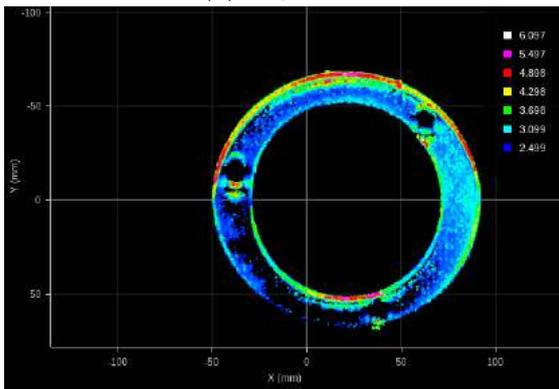
Como se pôde verificar anteriormente, foram efetuados vários testes para a configuração do sensor, tendo-se verificado que os melhores resultados são obtidos com as configurações presentes na Tabela 3.2.



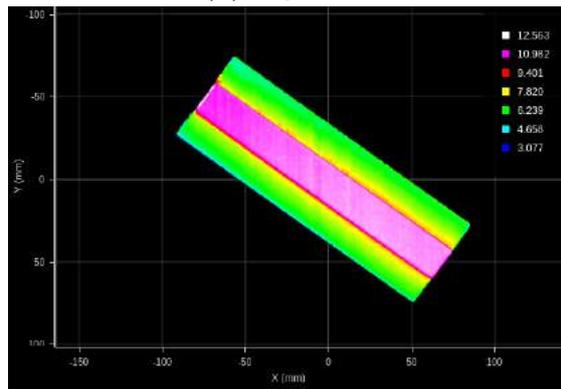
(a) Peça n.º 1



(b) Peça n.º 2

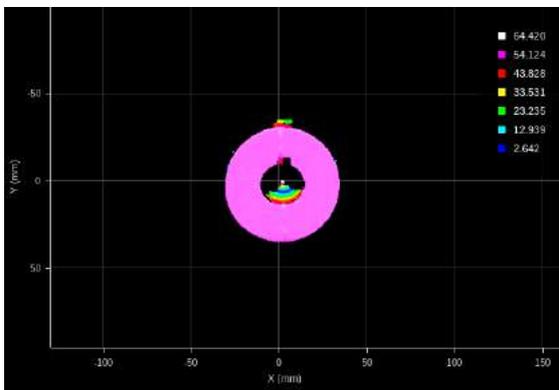


(c) Peça n.º 3

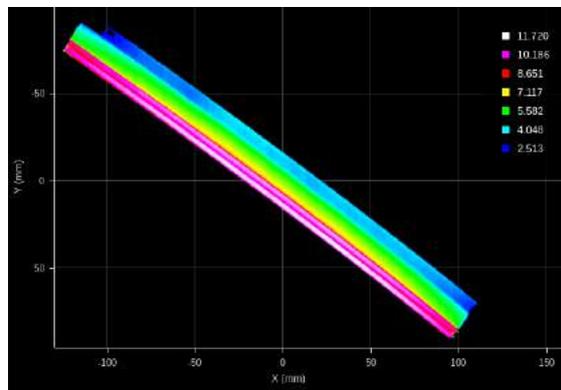


(d) Peça n.º 4

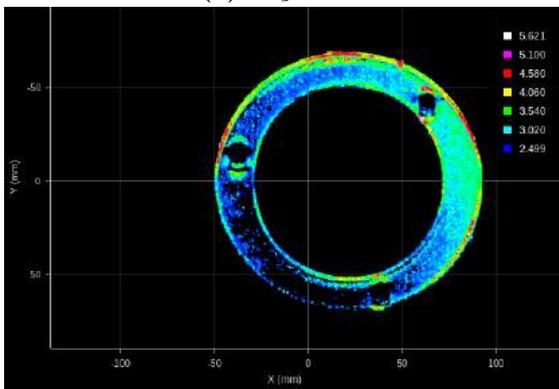
Figura 3.10: Exposição simples.



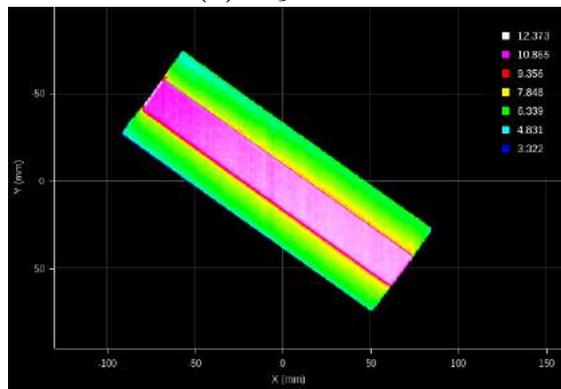
(a) Peça n.º 6



(b) Peça n.º 10

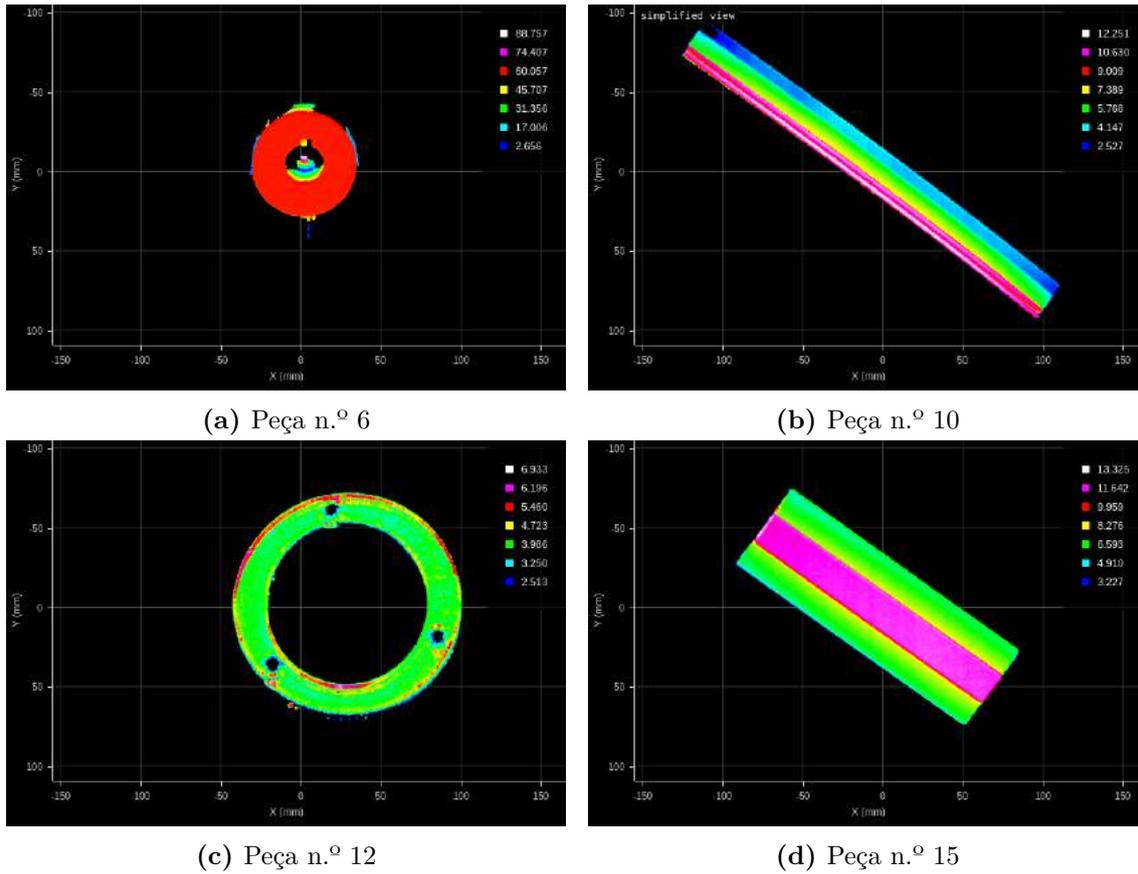


(c) Peça n.º 12



(d) Peça n.º 15

Figura 3.11: Exposição dinâmica.



**Figura 3.12:** Exposição múltipla.

		Configurações utilizadas
Modo de scan		Surface
Fonte de trigger		Encoder
Área ativa	X Field of View	332 mm
	Measurement Range	-155 mm
	X Start	515 mm
	Z Start	250 mm
Alinhamento	Tipo	Stationary
	Target	Flat Surface
Modo de exposição		Multiple
Surface generation		Continuous
Part detection	Frame of reference	Sensor
	Altura de threshold	2.5 mm
	Distância entre peças	5 cm
	Comprimento máximo das peças	30 cm

**Tabela 3.2:** Configurações do sensor.

### 3.6 Identificação das peças

O objetivo desta secção consiste na apresentação do trabalho realizado para a identificação das peças 3D. Começou-se por testar a funcionalidade disponibilizada pelo sensor denominada por *Models*. Após se verificar que esta funcionalidade não iria servir para a resolução do problema em questão, partiu-se para outra solução que consistiu na utilização

dos descritores 3D. Após a aquisição das *point clouds*, passam-se estes dados aos descritores 3D, obtendo desta forma um vetor de menor dimensão que iria caracterizar cada uma das peças e, posteriormente, calcula-se a distância euclidiana entre os vetores obtidos para cada uma delas. É com base nesta distância que se faz a identificação das peças.

### 3.6.1 Identificação de peças com o sensor

O sensor disponibiliza uma funcionalidade de *matching* de *point clouds* que será apresentada nesta secção. Para a utilização desta funcionalidade, devem ter-se em conta os seguintes aspetos:

- Modo de scan deve estar em *Surface*;
- Frame de referência do sensor, configurado através da *Part Detection*, deve ser colocado em *Part*;
- Ativação da *Part Detection*.

Cumprindo os requisitos acima descritos, começa-se por fazer a aquisição de dados para a criação dos modelos das peças a identificar, efetuando de seguida o *matching*, calculado pelo sensor, entre os modelos criados e as *point clouds* que iam sendo adquiridas.

A Gocator disponibiliza três algoritmos de *matching* distintos, nomeadamente, *Bounding Box*, *Ellipse* e *Edges*. Para usar os algoritmos da *bounding box* e da *ellipse*, é necessário definir as dimensões das mesmas para que a peça seja aceite como *matched*. No entanto, para este caso pretende-se que os algoritmos sejam capazes de identificar os vários tipos de peças e não os que fazem *match* com as dimensões de uma *bounding box* ou *ellipse* uma vez que grande parte das peças têm dimensões extremamente próximas. Deste modo, foi utilizado o algoritmo *edges* para efetuar os testes de *matching* com as peças n.º 1, n.º 2, n.º 6 e n.º 8.

Quando se utiliza o algoritmo de deteção de *edges*, o sensor compara um modelo, que deverá ser criado inicialmente, com um *target* que corresponde à peça a que se irá fazer a correspondência com o modelo selecionado. Os resultados de *matching* são apresentados em forma de percentagem, ou seja, o resultado obtido corresponde à percentagem de *matching* entre a peça modelo e a peça de teste. É importante referir que as peças de teste apenas serão consideradas pertencentes a um determinado modelo se a percentagem de *matching* obtida for superior a um determinado valor, ou seja, superior à *acceptance criteria*. Neste caso, considerou-se que este critério teria o valor de 90% e, tanto os resultados como as peças utilizadas para exemplo de *matching*, podem ser encontrados no Apêndice A.

Apesar de este método apresentar bons resultados, não é capaz de solucionar o problema em questão, uma vez que apenas nos garante o grau de certeza entre uma peça pertencer ou não a um determinado modelo, sendo que apenas se pode considerar um único modelo de referência. Isto não resolve o problema uma vez que o objetivo passa por identificar exatamente o tipo de peça em questão, ou seja, seria necessário ter todos os modelos selecionados, e esta funcionalidade não o permite. Assim sendo, passou-se para a utilização de descritores 3D.

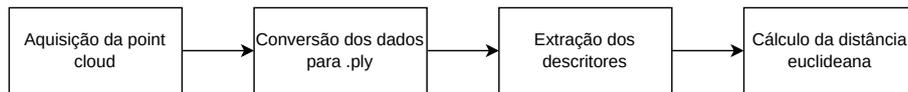
### 3.6.2 Descritores 3D

Como referido no capítulo anterior, existem dois tipos de descritores 3D: descritores locais e descritores globais. Como o objetivo foi tratar cada peça como um todo, decidiu-se utilizar os descritores globais. Apesar de na maioria dos casos ser necessário o passo intermédio da segmentação, neste caso isso não foi necessário uma vez que o sensor permite adquirir as peças individualmente, desde que afastadas 5 cm. De notar que o espaçamento

de 5 cm foi um dos requisitos da empresa, ou seja, este espaçamento pode variar desde que devidamente especificado na configuração do sensor.

Para resolver o problema da identificação das peças seguiu-se a metodologia descrita pela Figura 3.13, explicada a seguir, e tiveram-se em conta os seguintes descritores 3D:

- VFH;
- ESF;
- CVFH;
- OUR-CVFH;
- GRSD;
- GASD;
- GOOD;



**Figura 3.13:** Diagrama de fluxo da identificação das peças.

De modo a garantir uma correta identificação das peças, independentemente da posição a que estas passam pelo sensor, fez-se a distribuição das mesmas em 3 posições distintas no tapete, mais concretamente, numa zona à esquerda, numa zona central e numa zona mais à direita do mesmo, obtendo assim resultados que nos dão uma maior certeza de que as peças serão lidas corretamente qualquer que seja a posição em que estas se encontrem no tapete.

Após a extração das *point clouds*, são calculados os descritores 3D, obtendo desta forma vetores de dimensão  $1 \times N$ , onde o valor de  $N$  varia de descritor para descritor, como pode ser observado na Tabela 3.3.

Descriptor	Dimension of the descriptor vector
VFH	1 x 308
ESF	1 x 54
CVFH	1 x 308
OUR-CVFH	1 x 308
GRSD	1 x 21
GASD	1 x 512
GOOD	1 x 675

**Tabela 3.3:** Dimensões dos descritores 3D.

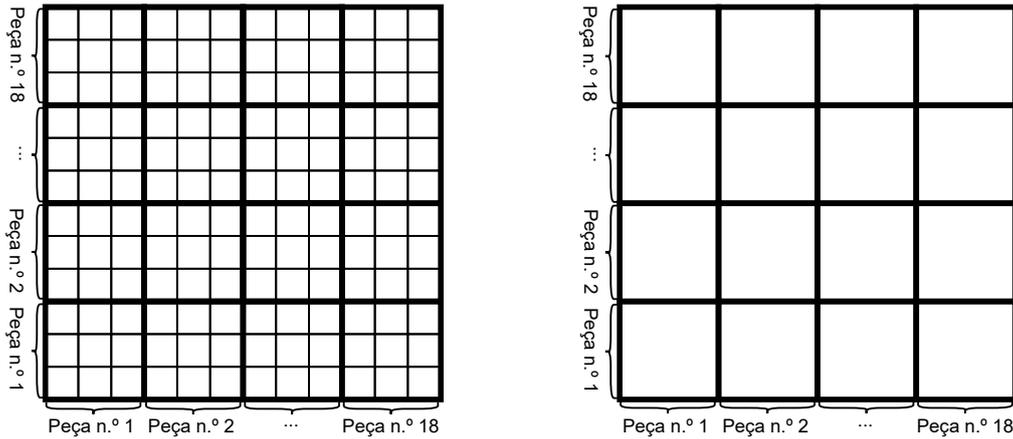
Posteriormente, para cada um dos descritores utilizados, é calculada a distância euclidiana entre cada um dos vetores descritores obtidos para cada uma das peças. Desta forma, são construídas matrizes de 54x54 posições denominadas, a partir deste momento, por matrizes de discriminação. Através de (3.3) os valores destas matrizes são normalizados e compreendidos entre 0 e 1, onde  $X$  é o valor original da distância,  $X_{\min}$  e  $X_{\max}$  os valores mínimo e máximo sobre toda a matriz, respetivamente, e  $X_{\text{norm}}$  o valor normalizado.

$$X_{\text{norm}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \quad (3.3)$$

Os valores muito próximos de 0 representam os casos em que a distância euclidiana calculada entre os vetores obtidos pelos descritores entre a peça de teste e a peça modelo é muito próxima, enquanto que os valores muito próximos de 1 indicam o oposto, isto é, a distância euclidiana calculada entre os vetores dos descritores da peça de teste e da

peça modelo é elevada, ou seja, a peça de teste é muito diferente da peça modelo. Nestas matrizes foi considerado que as peças presentes nas linhas são as peças de teste e as peças das colunas são as peças modelo.

Uma vez que foram testadas 18 peças diferentes em 3 posições distintas no tapete (esquerda, centro e direita), obtiveram-se matrizes de 54x54. Cada matriz de 54x54 é dividida em 18x18 submatrizes (disjuntas) de 3x3, e cada uma destas submatrizes corresponde a uma peça nas 3 posições no tapete. De modo a simplificar a interpretação dos resultados, decidiu-se dividir as matrizes de 54x54 em matrizes de 18x18, obtendo desta forma apenas uma distância entre as 18 peças distintas, tal como exemplificado na Figura 3.14.



**Figura 3.14:** Simplificação das matrizes de discriminação.

Este valor foi obtido para o pior dos cenários, ou seja, para os blocos 3x3 na diagonal da matriz, considerou-se o valor da distância mais elevada, enquanto que para os restantes foi considerada a distância mais baixa. Este é o pior dos cenários uma vez que é onde as peças irão ter maior probabilidade de serem confundidas. Na presente secção apenas serão apresentadas as matrizes de discriminação simplificadas, ou seja, as matrizes 18x18. Os valores completos dos dados para as matrizes de 54x54 são apresentados no Apêndice B.

Finalmente, considera-se que a identificação é bem feita se estas matrizes tiverem os valores da diagonal muito próximos de zero (a azul escuro na matriz) e os restantes o mais afastados do valor 0 e, idealmente, com o valor 1 ou muito próximos disso.

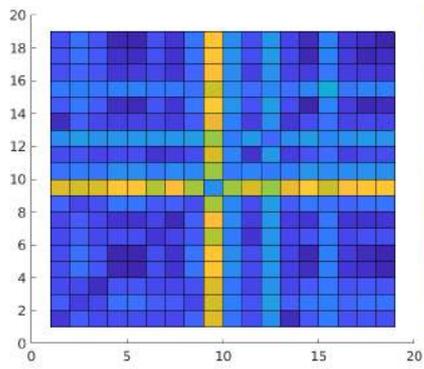
## Resultados

Na Figura 3.15 são apresentadas as matrizes de discriminação obtidas pelos descritores e, após a análise desta figura, é possível verificar que a identificação das peças não é efetuada corretamente. Isto verifica-se porque, por exemplo, nos descritores VFH, ESF, CVFH, OUR-CVFH, GRSD e GASD, praticamente todos os valores próximos de 0 se encontram fora da diagonal e, dentro desta, existem valores próximos de 1. Apesar de isto não acontecer tanto no descritor GOOD, ainda existem peças mal identificadas.

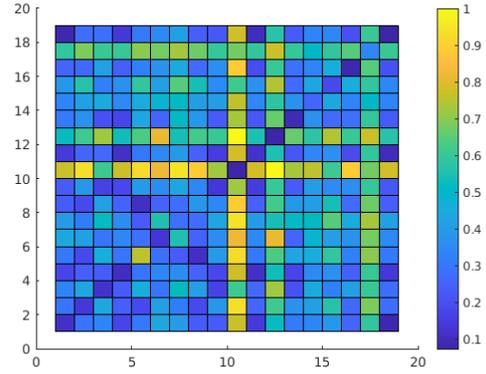
De modo a comprovar o que foi dito anteriormente, calculou-se o inverso da distância euclidiana entre os vetores das peças modelo e das peças de teste e, posteriormente aplicou-se a função softmax. Esta função recebe como entrada um vetor de  $K$  números reais e converte-o numa distribuição de probabilidades, normalmente definida por (3.4).

$$\text{softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^k e^{z_j}} \quad (3.4)$$

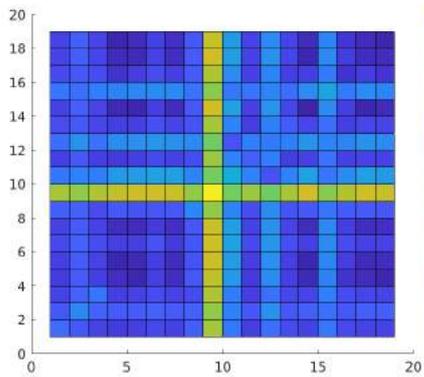
O vetor de saída, obtido por (3.4), representa a distribuição de confiança de cada classe e facilmente se percebe que os seus valores estarão compreendidos entre 0 e 1 e a sua soma será 1.



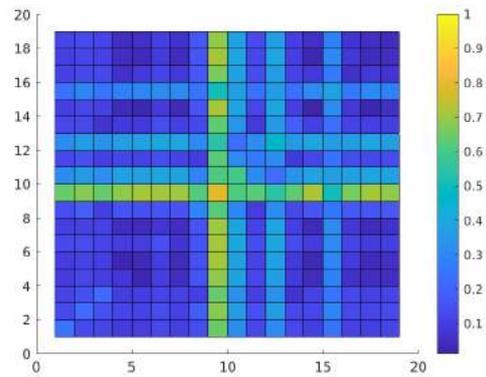
(a) VFH



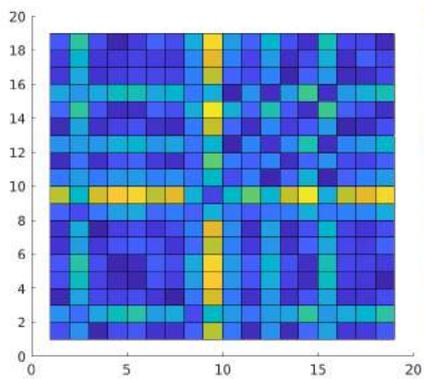
(b) ESF



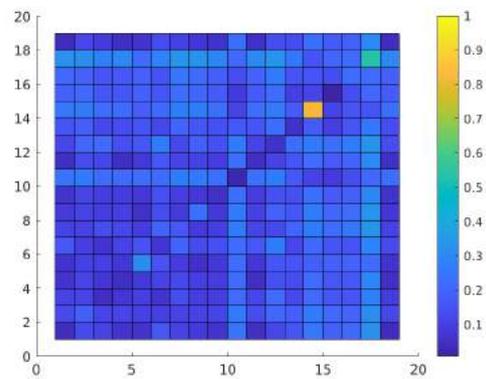
(c) CVFH



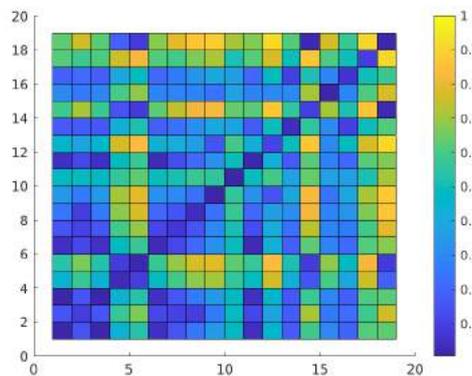
(d) OUR-CVFH



(e) GRSD



(f) GASD



(g) GOOD

**Figura 3.15:** Matrizes de discriminação obtidas pelos descritores.

Para o problema em questão, o vetor de saída irá representar o grau de certeza de uma determinada peça de teste corresponder à peça modelo. Tal como anteriormente, os valores obtidos variam entre 0 e 1, no entanto, como foi aplicado o inverso das distâncias euclidianas, os valores mais próximos de 1 indicam que a peça de teste corresponde à peça modelo em questão e os valores mais próximos de 0 indicam o oposto.

Na Tabela 3.4 são apresentados os valores obtidos após a soma da diagonal principal (*Diagonal*), a soma dos restantes valores (*Others*) de cada um dos descritores e o resultado da divisão entre os valores *Diagonal* e *Others*. Estes valores são calculados após a aplicação da softmax, onde os resultados desta são apresentados na Figura 3.16.

Descriptor	Diagonal	Others	Ratio
VFH	7.6921	10.2625	0.7495
ESF	9.3311	8.6115	1.0836
CVFH	2.7481	15.2055	0.1807
OUR-CVFH	2.3938	15.5546	0.1539
GRSD	1.1694	16.8267	0.0695
GASD	6.5360	11.4615	0.5703
GOOD	16.0149	1.9848	8.0687

**Tabela 3.4:** Valores obtidos após a softmax.

Tendo em consideração os resultados obtidos na Figura 3.16 e na Tabela 3.4, facilmente se verifica que a identificação das peças através dos descritores VFH, ESF, CVFH, OUR-CVFH, GRSD e GASD está muito longe de ser efetuada corretamente. Por exemplo, na Figura 3.16a a peça n.º 1 está a ser identificada como sendo a peça n.º 13 e a peça n.º 13 como peça n.º 1 e na Figura 3.16b, a peça n.º 1 está a ser identificada como sendo a peça n.º 18 e a peça n.º 5 como a peça n.º 8. Para além disso, sabendo que os valores das matrizes da Figura 3.16 estão compreendidos entre 0 e 1, é fácil perceber que a soma dos valores das diagonais principais seria 18, ou muito próximo disso, e a soma dos valores fora destas seria 0, ou muito próximo disso. No entanto, pode-se verificar que através do descritor GOOD já são obtidos valores mais corretos.

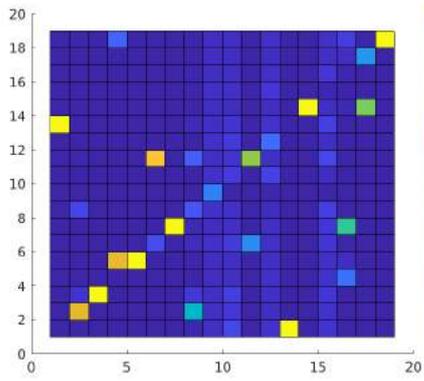
Estes resultados são confirmados através da Tabela 3.4 onde se pode verificar que com o descritor GOOD é o que obtém o valor *Diagonal* mais próximo de 18 e o valor *Others* mais próximo de 0.

Tendo em consideração as conclusões obtidas anteriormente, e com o intuito de tornar o vetor descritor do GOOD mais completo para conseguir identificar corretamente todas as peças, foram extraídas mais características, nomeadamente, a altura e a área que são obtidas através dos histogramas, e também o perímetro e dimensões da *bounding box* das peças, sendo estas últimas dadas por W1 e W2.

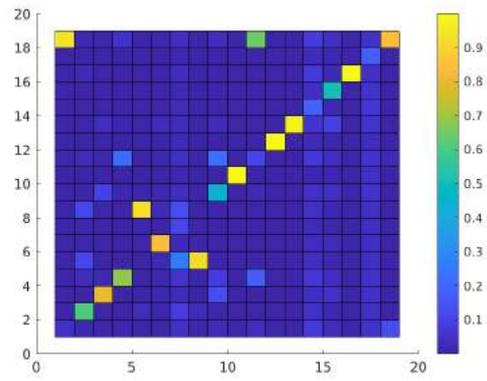
A título de exemplo, na Figura 3.17 apenas são apresentados dois dos histogramas obtidos, mais concretamente, os histogramas das peças n.º 2 e n.º 8, bem como a vista frontal das mesmas. Para observar os histogramas completos, pode-se consultar o Apêndice C.

Na Figura 3.18 são apresentadas as matrizes de discriminação e softmax obtidas através da concatenação dos vetores das *features* altura e área (Figura 3.18a) e dos vetores das *features* altura, área, perímetro e dimensões da *bounding box* das peças (Figura 3.18b). Nesta figura também são apresentadas as matrizes de discriminação e softmax através da soma entre os vetores normalizados do descritor GOOD e das *features* altura e área (Figura 3.18c) e entre as *features* altura, área, perímetro e dimensões da *bounding box* das peças (Figura 3.18d).

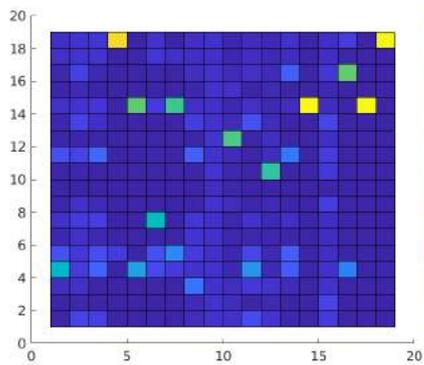
Após a análise dos resultados obtidos pela Figura 3.18 é possível verificar que a identificação das peças apenas é feita corretamente através da soma entre o descritor GOOD e as *features* altura e área.



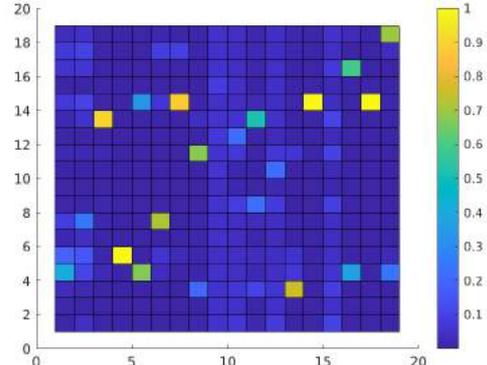
(a) VFH



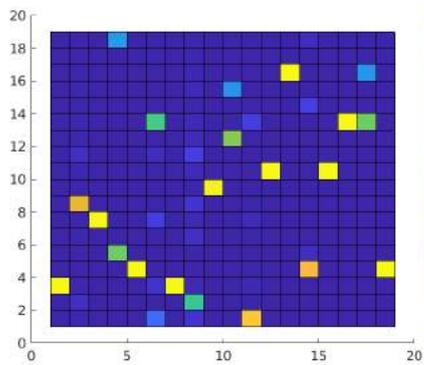
(b) ESF



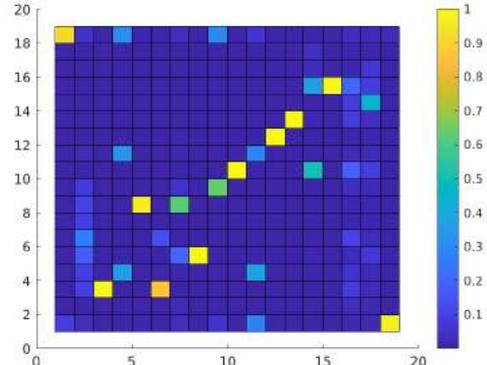
(c) CVFH



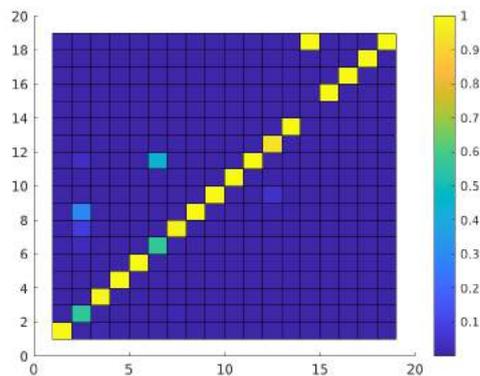
(d) OUR-CVFH



(e) GRSD

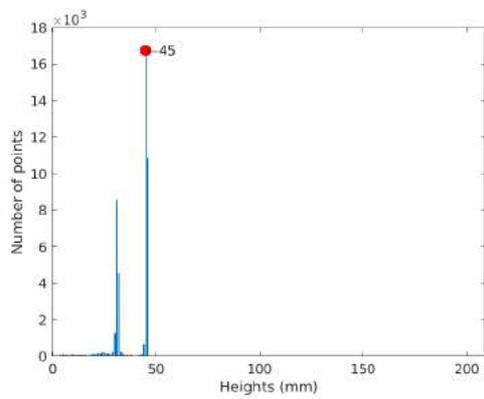


(f) GASD

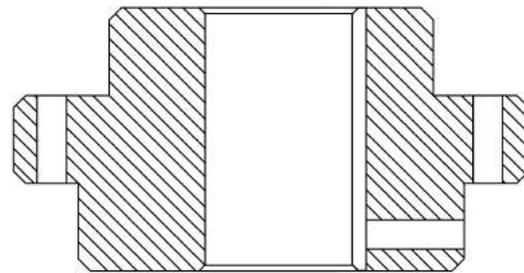


(g) GOOD

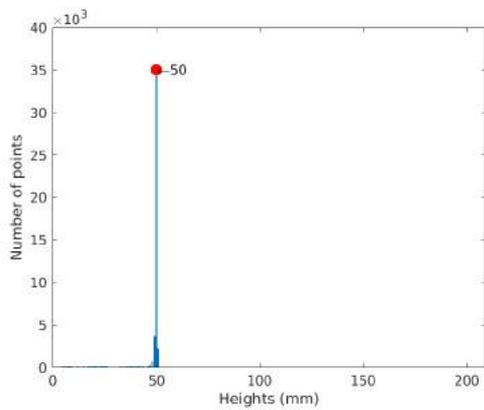
Figura 3.16: Resultados obtidos através da softmax.



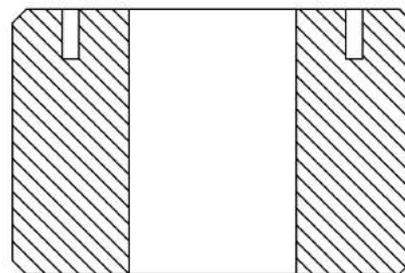
(a) Histograma da cota da peça n.º 2



(b) Corte da peça n.º 2

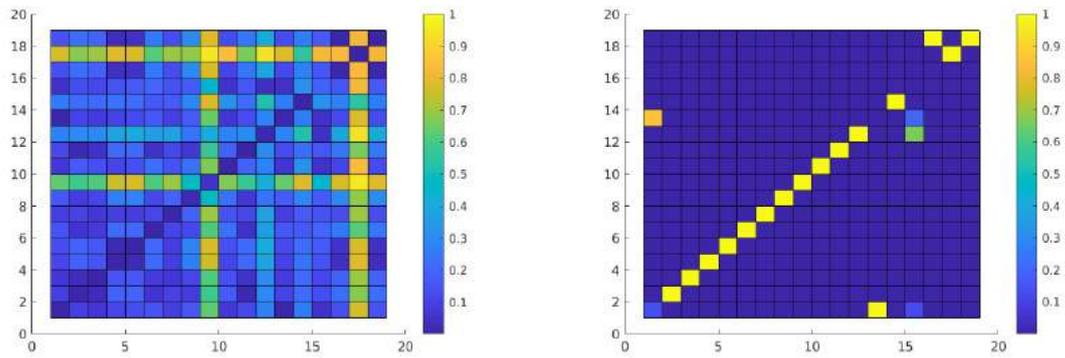


(c) Histograma da cota da peça n.º 8

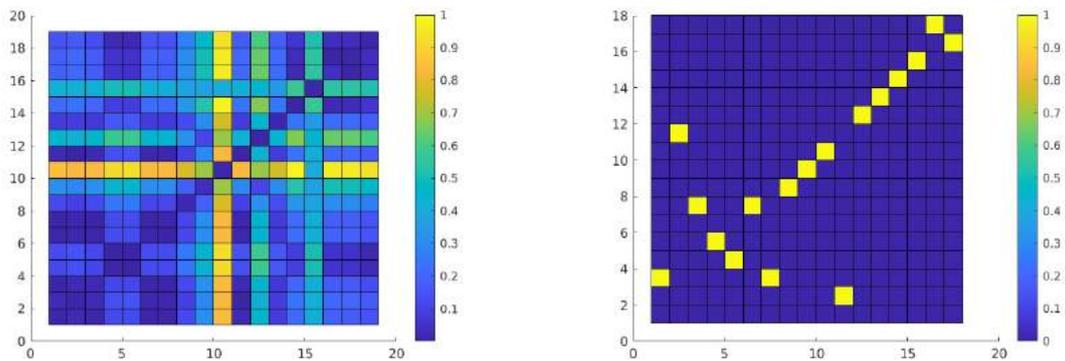


(d) Corte da peça n.º 8

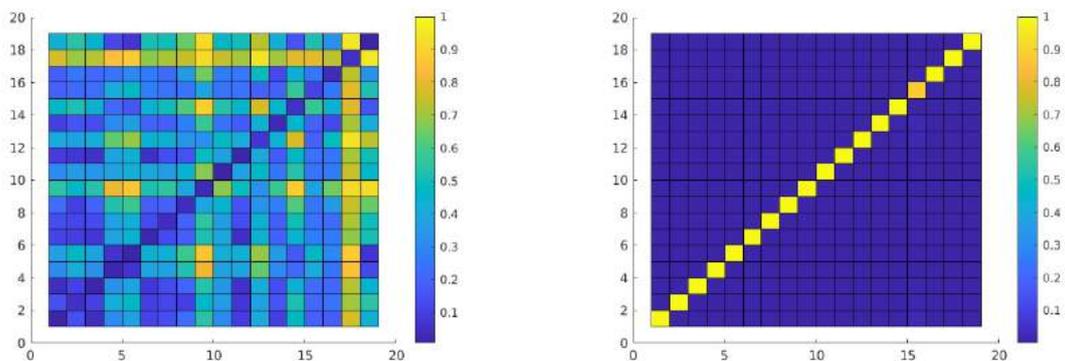
**Figura 3.17:** Histogramas e vista frontal das peças n.º 2 e n.º 8.



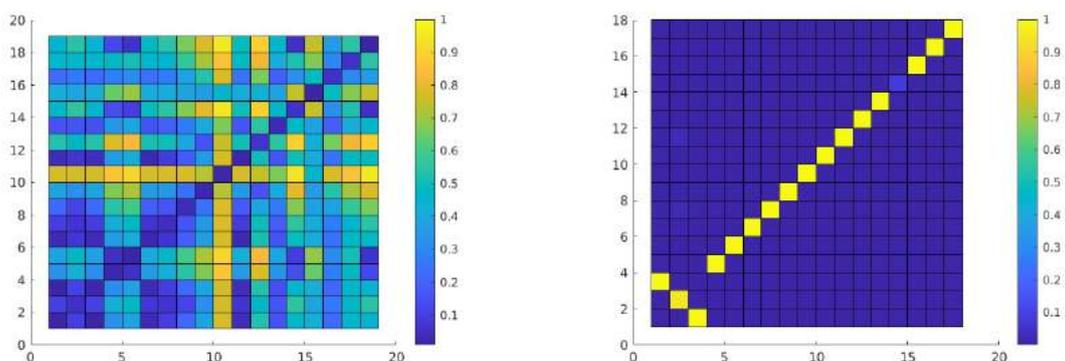
(a) *Features* altura e área



(b) *Features* altura, área, perímetro e W1 e W2



(c) Soma do descritor GOOD com as *features* altura e área



(d) Soma do descritor GOOD com as *features* altura, área, perímetro e W1 e W2

**Figura 3.18:** Matrizes de discriminação e softmax obtidas.

Na Figura 3.18a, a identificação das peças não é efetuada corretamente uma vez que, por exemplo, as peças n.º 1 e n.º 13 têm uma maior probabilidade de serem identificadas como sendo as peças n.º 13 e a peça n.º 16, respetivamente. Na Figura 3.18d apesar de existir uma certa melhoria na identificação das peças, ainda existem incorreções, mais especificamente nas peças n.º 1 e n.º 3 que são identificadas como sendo as peças n.º 3 e n.º 1, respetivamente.

Na Tabela 3.5 são apresentados os valores obtidos através da softmax nos testes mencionados anteriormente. Estes dados comprovam que a utilização do descritor GOOD com as *features* altura e área é a melhor solução para a identificação das peças 3D uma vez que o valor obtido na soma da diagonal principal softmax é muito próximo de 18 e a soma dos restantes valores é muito próxima de 0.

Descriptor	Diagonal	Others	Ratio
GOOD	16.0149	1.9848	8.0687
[H A]	14.1388	3.8612	3.6618
[H A P W1 W2]	8.0000	10.0000	0.8000
GOOD + [H A]	17.8751	0.1246	143.5167
GOOD + [H A P W1 W2]	15.0211	2.9789	5.0425

**Tabela 3.5:** Valores obtidos na soma do descritor GOOD com as *features* altura e área após a softmax.

Em suma, considerando os resultados obtidos pela softmax Figura 3.18c e na Tabela 3.5, pode-se considerar que apesar do grau de certeza da peça n.º 15 ser ligeiramente inferior a 1 (aproximadamente 0.8776), a identificação das peças é efetuada corretamente. Também é importante referir que este descritor cumpre os requisitos temporais, uma vez que os vetores descritores das 54 aquisições foram extraídos num total de 20,74 segundos.

### 3.7 Conclusões

Neste capítulo foi descrito o trabalho que permitiu efetuar uma correta identificação das peças.

Começaram-se por realizar vários testes ao posicionamento do sensor relativamente ao tapete transportador e, posteriormente, seguiram-se vários testes de exposição para cada um dos posicionamentos testados para o sensor e para cada um dos materiais das peças disponibilizadas. Todos estes testes realizados tiveram o objetivo de conseguir extrair os dados com a melhor qualidade possível.

Após uma correta aquisição dos dados, foi possível avançar para a identificação das peças. Para isto foram testados vários descritores 3D da PCL, nomeadamente, VFH, ESF, CVFH, OUR-CVFH, GRSD e GASD, onde não foi possível efetuar uma correta identificação das peças.

Por esta razão, foi testado o descritor GOOD que, relativamente aos descritores mencionados anteriormente, apresentou melhorias significativas nos resultados de identificação das peças. No entanto, nem todas as peças foram corretamente identificadas e, por essa razão, concluiu-se que era necessário adicionar mais informação ao vetor descritor. Desta forma, decidiu-se adicionar informação sobre altura, área, perímetro e dimensões das *bounding box* das peças, tendo obtido o melhor resultado através da soma do vetor descritor do GOOD e do vetor resultante da concatenação da informação da altura e área das peças, ambos normalizados.



## Capítulo 4

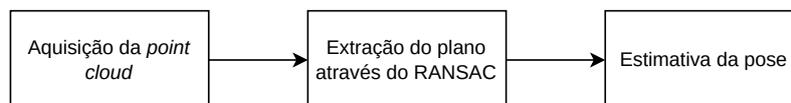
# Metodologia e resultados na estimativa da pose das peças

No presente caso de estudo, não só é importante efetuar uma correta identificação das peças, como também é de extrema importância estimar corretamente a sua posição e orientação no tapete transportador (pose). Neste capítulo será apresentado o trabalho desenvolvido nesse sentido, começando por apresentar as abordagens e resultados para a estimativa da orientação das peças, seguindo para a apresentação do trabalho desenvolvido e respectivos resultados na correta estimativa da posição destas.

### 4.1 Estimativa da pose das peças

Quando se faz o reconhecimento de objetos, não só interessa identificar qual o objeto que se encontra no tapete transportador, mas também a sua disposição sobre este, ou seja, a sua posição e orientação sobre o tapete.

No caso das peças com que foram realizados os ensaios, contrariamente ao que acontecia na identificação das peças onde se necessitava de toda a informação das *point clouds* adquiridas, na estimativa da orientação isso não é necessário uma vez que é no plano superior que se extrai a informação relevante para efetuar esta estimativa. Nestas condições, antes de se estimar a orientação das peças, é extraído o plano de todas as *point clouds*, como é ilustrado na Figura 4.1.



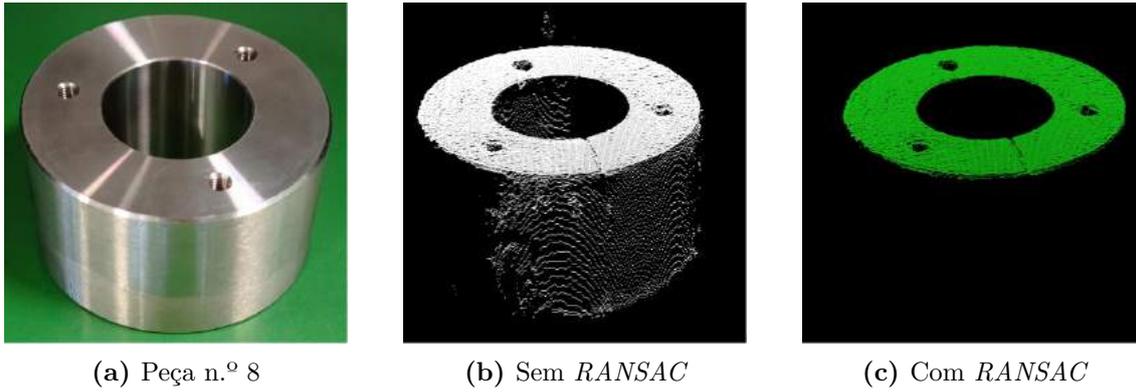
**Figura 4.1:** Diagrama de fluxo para obter a estimativa da pose.

Dadas as características das peças utilizadas para teste, decidiu-se aplicar o RANSAC com o objetivo de extrair o plano superior destas.

Na Figura 4.2 são apresentados os resultados das *point clouds* obtidas antes e após a aplicação do RANSAC. Como se pode verificar, é possível extrair apenas a face superior das peças que, para as peças utilizadas, é onde se tem toda a informação relevante para determinar a sua orientação.

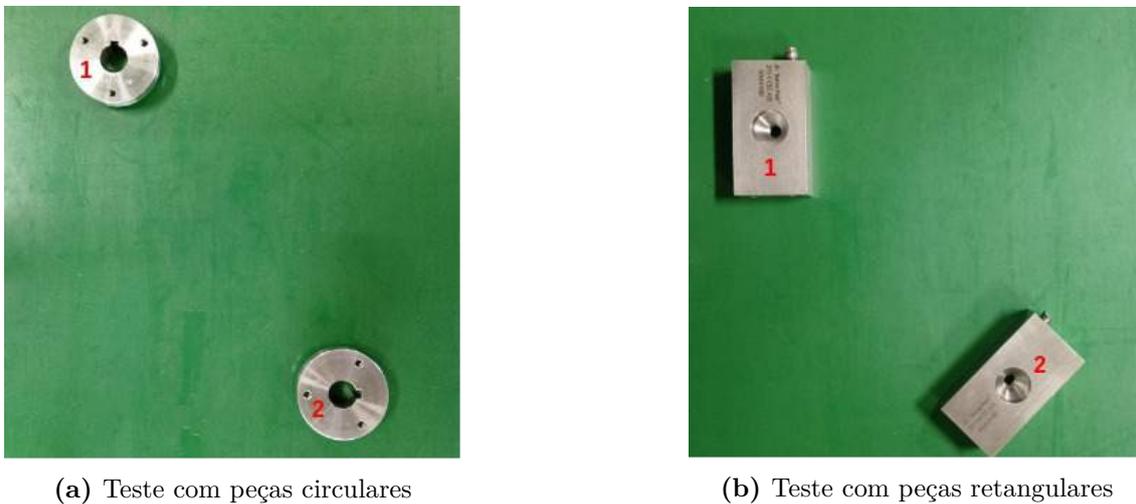
### 4.2 Estimativa da pose através do ICP

Começaram-se por fazer testes com peças de diferentes formas, mais concretamente, com uma peça circular e com uma peça retangular, como se pode verificar na Figura 4.3,



**Figura 4.2:** *Point cloud* antes de aplicar o RANSAC (b) e depois de aplicar o RANSAC (c) na peça n.º 8 (a)

onde as peças identificadas com “1” e “2” são as peças modelo e peças de teste, respectivamente.



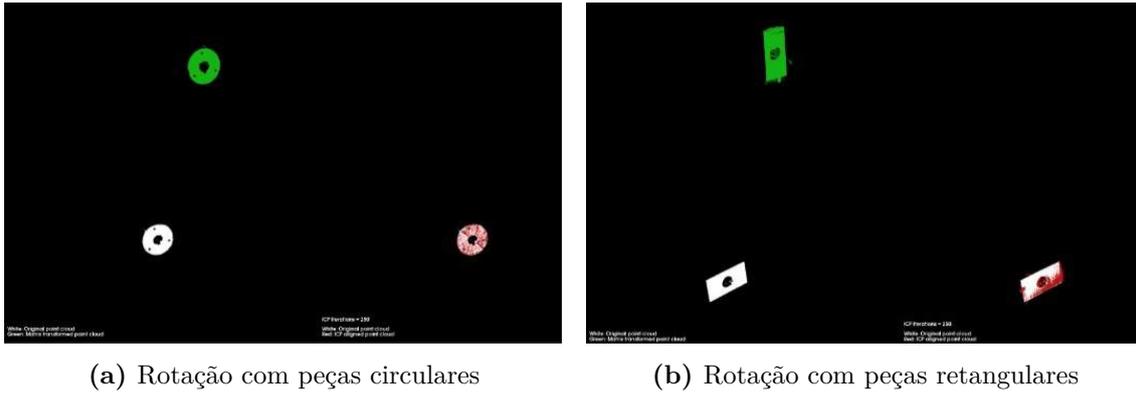
**Figura 4.3:** Teste realizado com peças circulares e retangulares.

Em seguida, passaram-se as *point clouds* pelo RANSAC e depois aplicou-se o ICP que, tal como mencionado no Capítulo 2, é um dos algoritmos mais populares para efetuar o alinhamento de *point clouds*, obtendo os resultados apresentados na Figura 4.4. Nas figuras 4.4a e 4.4b aparecem do lado esquerdo, a verde e a branco, as *point clouds* modelo e de teste, respetivamente. Nessas mesmas figuras, do lado direito, aparecem a vermelho e a branco, as *point clouds* modelo e de teste resultantes da aplicação do ICP, respetivamente.

Pela análise da Figura 4.4 é possível verificar que o ICP consegue centrar corretamente as peças. No entanto, após a análise da Tabela 4.1, é possível verificar que, apesar de centradas, as rotações das *point clouds* não estão corretas. Nesta tabela as colunas *Roll*, *Pitch* e *Yaw* referem-se às rotações em torno dos eixos do x, y e z, respetivamente.

	Resultados esperados			Resultados obtidos		
	Roll	Pitch	Yaw	Roll	Pitch	Yaw
Peça n.º 1	0°	0°	45°	179.75°	-0.13°	-55.08°
Peça n.º 13	0°	0°	90°	179.89°	0.08°	-88.01°

**Tabela 4.1:** Rotações obtidas através do ICP.



**Figura 4.4:** Resultados obtidos através do ICP.

Como mencionado no Capítulo 2, as únicas rotações possíveis seriam em torno do eixo  $z$  (*Yaw*). No entanto, através da Tabela 4.1, verifica-se que o ICP considera haver rotações de  $179.75^\circ$  e  $179.89^\circ$  no eixo do  $x$  da peça n.º 1 e n.º 13, respetivamente, e de  $-0.13^\circ$  e  $-0.08^\circ$  no eixo do  $y$  da peça n.º 1 e n.º 13, respetivamente. Apesar destes resultados não serem os esperados, não são considerados completamente errados uma vez que, em relação ao eixo do  $x$ , a rotação é de aproximadamente  $180^\circ$  e, por esta razão, consideram-se valores admissíveis. Para além disso, como as peças modelo e de teste são planares e existem simetrias, é normal que o ICP possa considerar que os planos estão em sentidos inversos.

À primeira vista, pode-se pensar que a rotação obtida está completamente errada, no entanto, olhando com maior detalhe, é possível verificar que não é necessariamente verdade uma vez que ainda existe uma rotação em torno do eixo do  $z$  que vai compensar a diferença de  $180^\circ$  em torno do eixo do  $x$ . Relativamente aos ângulos obtidos em torno do eixo do  $y$ , estes também são relativamente aceitáveis uma vez que diferem menos de  $1^\circ$  do valor esperado. No entanto, quando se comparam os ângulos obtidos em torno do eixo do  $z$  é possível verificar que a peça circular apresenta um erro de aproximadamente  $10^\circ$  e a peça retangular apresenta um erro de aproximadamente  $2^\circ$ . Apesar do erro apresentado em relação à peça retangular não ser demasiado elevado, o mesmo não acontece para a peça circular. Desta forma, é possível concluir que será necessário ter processos diferentes para o cálculo da rotação das peças.

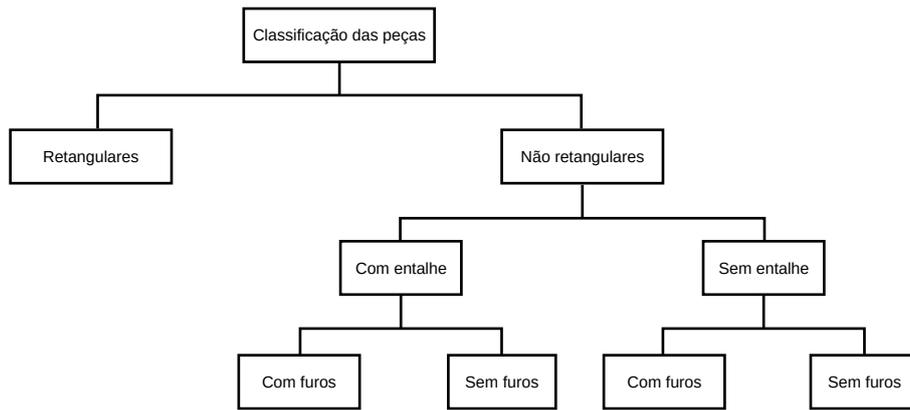
Na Tabela 4.2 são apresentados os valores estimados e os valores obtidos para a posição das peças, obtidos através do ICP. Através desta tabela também é possível verificar que existe algum erro em relação às transformações obtidas.

	Resultados esperados (cm)			Resultados obtidos (cm)		
	x	y	z	x	y	z
Peça n.º 1	-10.0	-15.0	0.0	-10.9	-13.7	-0.0
Peça n.º 13	10.0	10.0	0.0	9.5	9.0	3.6

**Tabela 4.2:** Translações obtidas através do ICP.

### 4.3 Estimativa da rotação das peças

Visto que os resultados obtidos anteriormente não são suficientemente precisos, partiu-se para outra abordagem e, após alguma discussão decidiu-se que, de acordo com o tipo de peça, serão usados algoritmos específicos para esta estimação e, por esta razão, é necessário classificar as peças de acordo com a sua forma, tal como apresentado na Figura 4.5.

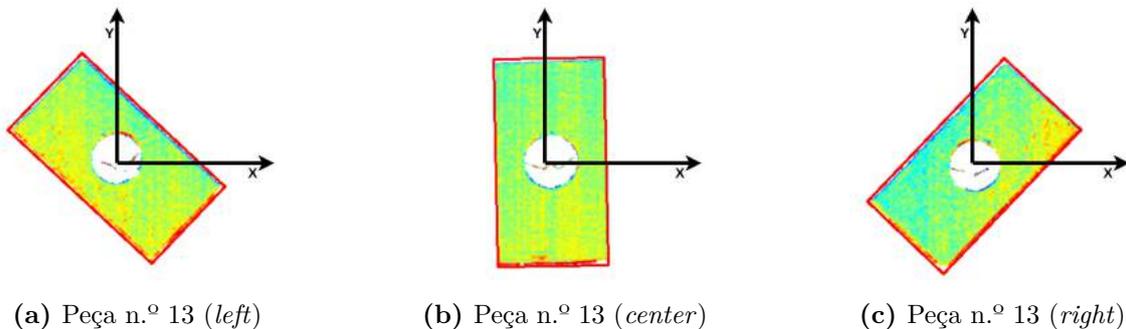


**Figura 4.5:** Classificação das peças com base na sua forma.

É importante referir que nesta dissertação apenas se irá descrever o processo do cálculo da rotação para as peças retangulares. As peças não retangulares foram abordadas por um investigador também pertencente ao projeto, no entanto, os resultados obtidos não serão aqui apresentados.

## Resultados

Nesta abordagem, começou-se por extrair o plano através do RANSAC e, posteriormente, converteram-se as *point clouds* obtidas em imagens. A seguir, foram detetados os contornos, como apresentado na Figura 4.6.



**Figura 4.6:** Resultados obtidos na deteção do retângulo de menor área para a peça n.º 13

Na Tabela 4.3 são apresentados os resultados obtidos através do cálculo da média das rotações obtidas após a realização de 10 experiências pelo método descrito acima.

Peças	Resultado esperado	Resultado obtido	Erro absoluto	Desvio padrão
n.º 13 4.6a	-45.0	-43.7	1.3	0.18
n.º 13 4.6b	0.0	0.9	0.9	0.26
n.º 13 4.6c	45.0	46.6	1.6	0.21

**Tabela 4.3:** Rotações obtidas através do método do retângulo de menor área.

Através da Tabela 4.3 é possível verificar que os resultados obtidos são muito satisfatórios uma vez que o erro absoluto é 1.3°, 0.9° e 1.6° para as peças apresentadas em 4.6a, 4.6b e 4.6c, respetivamente, e encontra-se dentro dos requisitos definidos pela empresa.

## 4.4 Estimativa da distância entre peças

Inicialmente, realizaram-se várias experiências para determinar o significado do parâmetro devolvido pelo sensor, denominado por *stamp encoder*, uma vez que foi detetado que cada vez que uma *point cloud* era adquirida, o sensor devolvia este parâmetro, bem como um *timestamp*.

Deste modo, o teste consistiu em colocar duas peças distanciadas de 15 cm para determinar se o *stamp encoder* correspondia ao valor do encoder no primeiro ou no último *frame* da peça, ou seja, se o valor devolvido correspondia ao valor do encoder quando a peça era detetada ou quando deixava de ser detetada.

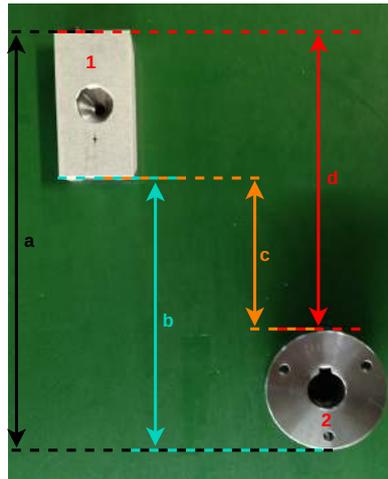


Figura 4.7: Estimativa da distância entre peças.

### Resultados

A Figura 4.7 representa o teste realizado, onde (a), (b), (c) e (d) são distâncias conhecidas e apresentadas a seguir:

- (a) distância entre a parte superior da peça 1 e a parte inferior da peça 2: 23 cm
- (b) distância entre a parte inferior da peça 1 e a parte inferior da peça 2: 15 cm
- (c) distância entre a parte inferior da peça 1 e a parte superior da peça 2: 9 cm
- (d) distância entre a parte superior das peças 1 e 2: 16.5 cm

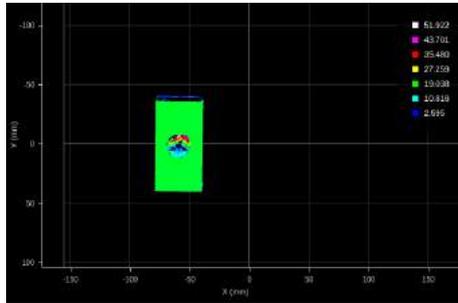
Com este teste foi possível concluir que o valor do *stamp encoder* corresponde à distância entre o final do *frame* de cada peça que passa pelo laser do sensor. É importante referir que este valor devolvido pelo sensor é o número atual de pulsos do encoder, ou seja, para converter este valor em *mm* é necessário utilizar (4.1):

$$value = pulses \times enc\_res \quad (4.1)$$

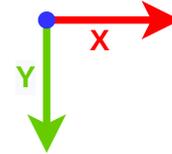
onde *pulses* é o valor da distância entre 2 peças em número de pulsos e *enc\_res* corresponde à resolução do encoder em mm/pulso. Na presente dissertação, o valor de *enc\_res* é de 0.047297, sendo este valor obtido através da calibração do sensor com o disco mencionado no Capítulo 3 e apresentado na Figura 3.6.

## 4.5 Estimativa da posição das peças

Adquirida a estimativa da distância entre peças, avançou-se para a estimativa da posição destas. Tendo em conta que a *point cloud* extraída pela Gocator é devolvida em relação ao centro do seu eixo do y, como apresentado na Figura 4.8a, definiu-se que a posição da peça, no eixo do x, seria dada em relação ao x do seu centróide e, por conseguinte, a posição de marcação na peça seria efetuada em relação a este.



(a) Resultado obtido através do scan



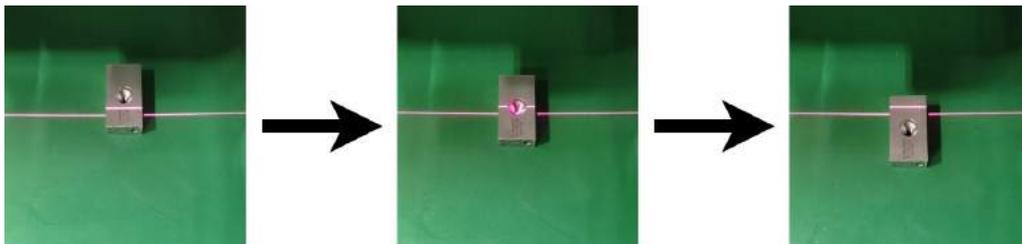
(b) Referenciais da Gocator

**Figura 4.8:** Scan da peça n.º 13

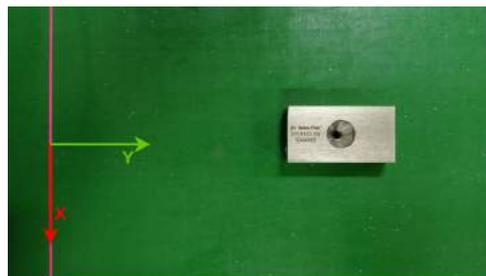
Tendo isto em atenção, utilizou-se a capacidade da Gocator conseguir obter a posição do centróide das peças e, deste modo, obteve-se a coordenada x do centróide da peça.

Para se obter a posição y da peça, começa-se por obter o seu *stamp encoder*, ou seja, o valor do encoder no momento que se obtém o último *frame* da peça. Para efeitos de teste, considerou-se que o tapete só é parado assim que a distância percorrida pela peça seja aproximadamente 15 cm, sendo esta distância escolhida apenas para dar espaço para a marcação das peças.

Na Figura 4.9 é apresentado um exemplo a demonstrar o processo descrito acima de modo a se obter a estimativa da posição das peças no tapete transportador e na Figura 4.10 é demonstrado o referencial do sistema.



**Figura 4.9:** Etapas de scan



**Figura 4.10:** Estimativa da posição da peça

## Resultados

Esta experiência foi repetida 10 vezes, sendo a média dos resultados obtidos apresentada na Tabela 4.4.

	Resultado esperado	Resultado obtido	Erro absoluto	Desvio padrão
x (cm)	2.0	2.26	0.26	0.04
y (cm)	15.0	17.74	2.74	0.05

**Tabela 4.4:** Média da estimativa da posição da peça n.º 13.

Através da Tabela 4.4 é possível verificar que os resultados obtidos para o eixo do x são bastante satisfatórios, uma vez que o erro absoluto é inferior a 0.5 cm.

O mesmo não acontece para o eixo y, que corresponde à distância percorrida pelas peças. Aqui obteve-se um erro de 2.74 cm. Isto acontece porque o tapete ainda se move desde o momento em que é dada a ordem de fim do movimento até que este termina efetivamente o seu movimento. No entanto, apesar de não se obter a distância pretendida, é possível obter o *tick* do encoder no momento em que o tapete transportador termina o movimento, ou seja, apesar dos 15 cm inicialmente determinados não serem respeitados, é possível saber a distância percorrida pela peça.

## 4.6 Marcação das peças

A marcação das peças baseou-se no diagrama apresentado na Figura 4.11.



**Figura 4.11:** Diagrama de fluxo para efetuar a marcação das peças.

Como se pode verificar, começa-se por extrair a *point cloud* das peças e, posteriormente é efetuado o cálculo da estimativa da posição das peças no tapete transportador, mais concretamente as coordenadas x e y do centróide das peças. Em seguida, esta estimativa é enviada para o manipulador robótico, bem como as coordenadas de marcação das peças. No final, o manipulador efetua a marcação das peças.

## Resultados

Na figura 4.12 é ilustrado o processo de marcação realizado com base em coordenadas de marcação definidas previamente.

Esta experiência também foi repetida 10 vezes para cada posição do tapete, ou seja, para o lado esquerdo, centro e lado direito do tapete. Deste modo, foram efetuadas 30 marcações e, no final, obteve-se aproximadamente um erro médio de 2.4 mm e um desvio padrão de 1.0 mm entre a posição de marcação estimada e obtida.

Na Figura 4.13 é apresentado o resultado obtido através da função distribuição acumulada que, neste caso, descreve a distribuição da probabilidade de a marcação obtida estar abaixo de um determinado valor. Por exemplo, através da Figura 4.13a, é possível verificar que existe uma probabilidade de 36.6% de a distância entre a posição de marcação obtida e a posição da marcação esperada ser inferior a 2 mm e que 73.3% das marcações obtidas estão a menos de 3 mm da marcação pretendida.

Na Figura 4.13b são apresentadas as funções distribuição nas diferentes posições do tapete, ou seja, no lado esquerdo, no centro e no lado direito do tapete. Daqui é possível

verificar que é na zona do lado direito do tapete onde as marcações são realizadas com maior precisão.

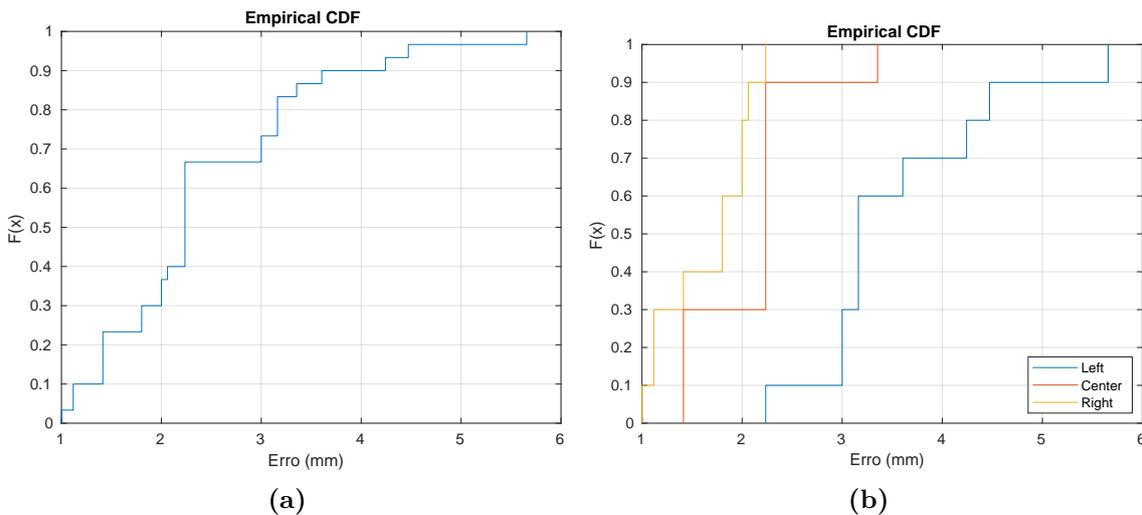


(a)



(b)

**Figura 4.12:** Simulação do marcador a laser



(a)

(b)

**Figura 4.13:** Função distribuição acumulada

## 4.7 Conclusão

Neste capítulo, começou-se por testar o ICP e, como os resultados obtidos para a estimativa da rotação não foram muito satisfatórios, decidiu-se passar para outra abordagem. Nesta abordagem, decidiu-se fazer a divisão das peças através da sua forma, ou seja, retangular e não retangular. Uma vez que este projeto conta com a colaboração de um investigador, o trabalho apresentado nesta dissertação apenas descreve a metodologia da estimativa da pose e resultados, das peças retangulares.

Tendo isto em atenção, e apenas considerando as peças retangulares, obtiveram-se estimativas bastante próximas da realidade.

Posteriormente, foram realizados testes de marcação onde o erro apresentado entre a posição estimada e a posição pretendida para a marcação, foi quase sempre inferior a

5 mm. Desta forma, é possível concluir que, em relação às peças abordadas na estimativa da pose, os resultados foram muito satisfatórios. Para além disso, foi possível verificar que a precisão da marcação varia de acordo com a posição da peça no tapete.



# Capítulo 5

## Conclusão

### 5.1 Considerações finais

A presente dissertação abordou o problema de identificação e marcação a laser de peças 3D. Este problema surge para facilitar o processo de marcação atual que depende de uma pessoa para o fazer. Para tal a JPM e a Universidade de Aveiro desenvolveram em co-promoção o projeto AdaptMark que tem como objetivo a investigação e o desenvolvimento de um sistema robótico de marcação a laser de peças de diferentes geometrias e materiais, com autonomia de operação, flexibilidade e adaptabilidade. O projeto foi dividido em três subsistemas: subsistema de identificação, subsistema de manipulação e marcação laser e subsistema de verificação de qualidade. Na presente dissertação abordou-se o subsistema de identificação. Aqui foram abordados vários temas como, por exemplo, os descritores 3D para efetuar a identificação das peças 3D e outros algoritmos como, por exemplo, o RANSAC e o ICP para ajudar na tarefa de obter a estimativa da pose dos objetos 3D.

Relativamente aos descritores 3D, fez-se um estudo aos descritores VFH, ESF, CVFH, OUR-CVFH, GRSD, GASD e, por último, o GOOD que revelou ser o melhor descritor para o problema em questão. No entanto, apesar deste último ser o descritor que apresentava os melhores resultados, não era capaz de identificar todas as peças e, por esta razão, era necessário adicionar mais informação ao vetor descritor obtido. Para tal, foi feito um estudo para descobrir quais as características a acrescentar para efetuar uma correta identificação das peças. No final, pôde-se verificar que as peças foram corretamente identificadas através da soma normalizada entre o vetor descritor GOOD e o vetor concatenado contendo o valor das alturas e áreas das peças.

Tendo esta etapa resolvida, passou-se para a estimativa da pose das peças. Aqui começou-se por aplicar o ICP e, uma vez que os resultados não foram satisfatórios, partiu-se para outra abordagem que consistiu na divisão das peças pela sua forma. Deste modo, na presente dissertação teve-se em conta o problema das peças retangulares uma vez que as peças não retangulares foram abordadas por um investigador envolvido no projeto. Assim sendo, e apenas testando para as peças retangulares, a solução final sobre a estimativa da pose destas peças apresentou resultados satisfatórios.

No final foram ainda feitos testes de marcação, sendo os resultados obtidos também satisfatórios uma vez que apresentaram um erro relativamente baixo, mais concretamente, inferiores a 5 mm. Deste teste também se pôde verificar que a precisão da marcação variava de acordo com a posição da peça no tapete.

### 5.2 Trabalho futuro

Um dos objetivos desta dissertação era fazer o reconhecimento de peças 3D e, desta forma, o trabalho baseou-se nos descritores 3D e no cálculo da distância euclidiana entre

estes. Assim sendo, no futuro, o objetivo passa por implementar algoritmos de *machine learning* para a identificação das peças.

Outro aspecto que não se teve em consideração, foi a possibilidade de existirem peças multi nível, ou seja, peças onde a marcação seria efetuada em várias camadas e, desta forma, teria que se ter maior atenção à fase de marcação das peças.

Nesta dissertação, também não se desenvolveu uma base de dados e, tendo em conta que o objetivo passa por implementar todo o sistema numa rede industrial, no futuro deverá ser desenvolvida uma base de dados para integrar num servidor dedicado, onde deverá ter, por exemplo, as características das peças a marcar e o tipo de marcação a efetuar.

Para efeitos de teste, as coordenadas de marcação foram inseridas manualmente, desta forma, no futuro deverá ser desenvolvida uma interface gráfica que permita agilizar o processo.

Para além disto, ainda é sugerida uma gestão do robô e do tapete, bem como a sua sincronização.

# Bibliografia

- [1] Abubakar Sulaiman Gezawa et al. “A Review on Deep Learning Approaches for 3D Data Representations in Retrieval and Classifications”. Em: *IEEE Access* 8 (2020), pp. 57566–57593. DOI: 10.1109/ACCESS.2020.2982196.
- [2] Aitor Aldoma et al. “Tutorial: Point Cloud Library: Three-Dimensional Object Recognition and 6 DOF Pose Estimation”. Em: *IEEE Robotics & Automation Magazine* 19 (2012), pp. 80–91.
- [3] Dirk Holz et al. “Registration with the Point Cloud Library - A Modular Framework for Aligning in 3-D”. Em: *IEEE Robotics & Automation Magazine* 22 (dez. de 2015), pp. 110–124. DOI: 10.1109/MRA.2015.2432331.
- [4] Radu Bogdan Rusu e Steve Cousins. “3D is here: Point Cloud Library (PCL)”. Em: *2011 IEEE International Conference on Robotics and Automation*. 2011, pp. 1–4. DOI: 10.1109/ICRA.2011.5980567.
- [5] Inês de Sousa Caldas. “Recognition and 6 DoF Pose Estimation of 3D Models in Depth Sensor Data”. Tese de mestrado. Faculdade de Engenharia da Universidade do Porto, 2018.
- [6] Qian-Yi Zhou, Jaesik Park e Vladlen Koltun. “Open3D: A Modern Library for 3D Data Processing”. Em: *arXiv:1801.09847* (2018).
- [7] Miguel Lopes Castro. *Desenvolvimento de uma toolbox de soluções para inspeção visual automática*. <https://hdl.handle.net/1822/69959>. 2019.
- [8] Khaled Alhamzi, Mohammed Elmogy e Sherif Barakat. “3D Object Recognition Based on Local and Global Features Using Point Cloud Library”. Em: *International Journal of Advancements in Computing Technology* 7 (mai. de 2015), pp. 43–54.
- [9] Radu Bogdan Rusu et al. “Fast 3D recognition and pose using the Viewpoint Feature Histogram”. Em: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2010, pp. 2155–2162. DOI: 10.1109/IROS.2010.5651280.
- [10] Radu Bogdan Rusu, Nico Blodow e Michael Beetz. “Fast Point Feature Histograms (FPFH) for 3D registration”. Em: *2009 IEEE International Conference on Robotics and Automation*. 2009, pp. 3212–3217. DOI: 10.1109/ROBOT.2009.5152473.
- [11] Radu Bogdan Rusu et al. “Learning informative point classes for the acquisition of object model maps”. Em: *2008 10th International Conference on Control, Automation, Robotics and Vision*. Dez. de 2008, pp. 643–650. DOI: 10.1109/ICARCV.2008.4795593.
- [12] Frank P. Ferrie, Jean Lagarde e Peter Whaite. “Darboux Frames, Snakes, and Super-Quadrics: Geometry from the Bottom Up”. Em: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 15.8 (1993). Cited by: 108, pp. 771–784. DOI: 10.1109/34.236252. URL: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-0027646706&doi=10.1109%5C%2F34.236252&partnerID=40&md5=81dc13a64b0e83c81d046778b7e30155>.

- [13] Khaled Alhamzi, Mohammed Elmogy e Sherif Barakat. “3D Object Recognition Based on Local and Global Features Using Point Cloud Library”. Em: *International Journal of Advancements in Computing Technology* 7 (mai. de 2015), pp. 43–54.
- [14] R. Osada et al. “Matching 3D models with shape distributions”. Em: *Proceedings International Conference on Shape Modeling and Applications*. 2001, pp. 154–166. DOI: 10.1109/SMA.2001.923386.
- [15] Walter Wohlkinger e Markus Vincze. “Ensemble of shape functions for 3D object classification”. Em: *2011 IEEE International Conference on Robotics and Biomimetics*. 2011, pp. 2987–2992. DOI: 10.1109/ROBIO.2011.6181760.
- [16] Aitor Aldoma et al. “CAD-model recognition and 6DOF pose estimation using 3D cues”. Em: *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*. 2011, pp. 585–592. DOI: 10.1109/ICCVW.2011.6130296.
- [17] Aitor Aldoma et al. “OUR-CVFH - Oriented, Unique and Repeatable Clustered Viewpoint Feature Histogram for Object Recognition and 6DOF Pose Estimation”. Em: *DAGM/OAGM Symposium*. 2012.
- [18] Zoltan-Csaba Marton et al. “Hierarchical object geometric categorization and appearance classification for mobile manipulation”. Em: *2010 10th IEEE-RAS International Conference on Humanoid Robots*. 2010, pp. 365–370. DOI: 10.1109/ICHR.2010.5686323.
- [19] Zoltan-Csaba Marton et al. “General 3D modelling of novel objects from a single view”. Em: *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. 2010, pp. 3700–3705. DOI: 10.1109/IRoS.2010.5650434.
- [20] João Paulo Silva do Monte Lima e Veronica Teichrieb. “An Efficient Global Point Cloud Descriptor for Object Recognition and Pose Estimation”. Em: *2016 29th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*. 2016, pp. 56–63. DOI: 10.1109/SIBGRAPI.2016.017.
- [21] Hamidreza Kasaei et al. “GOOD: A Global Orthographic Object Descriptor for 3D Object Recognition and Manipulation”. Em: *Pattern Recognition Letters* 83 (jul. de 2016). DOI: 10.1016/j.patrec.2016.07.006.
- [22] Anh Nguyen e Bac Le. “3D point cloud segmentation: A survey”. Em: *2013 6th IEEE Conference on Robotics, Automation and Mechatronics (RAM)*. 2013, pp. 225–230. DOI: 10.1109/RAM.2013.6758588.
- [23] Martin A. Fischler e Robert C. Bolles. “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography”. Em: *Commun. ACM* 24 (1981), pp. 381–395.
- [24] Konstantinos G Derpanis. “Overview of the RANSAC Algorithm”. Em: *Image Rochester NY* 4.1 (2010), pp. 2–3.
- [25] Paul Besl e H.D. McKay. “A method for registration of 3-D shapes. IEEE Trans Pattern Anal Mach Intell”. Em: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 14 (mar. de 1992), pp. 239–256. DOI: 10.1109/34.121791.
- [26] Paul J. Besl e Neil D. McKay. “A Method for Registration of 3-D Shapes”. Em: *IEEE Trans. Pattern Anal. Mach. Intell.* 14 (1992), pp. 239–256.
- [27] *Wiki*. URL: <http://wiki.ros.org/Distributions>.
- [28] Tully Foote. “tf: The transform library”. Em: *Technologies for Practical Robot Applications (TePRA), 2013 IEEE International Conference on*. Open-Source Software workshop. Abr. de 2013, pp. 1–6. DOI: 10.1109/TePRA.2013.6556373.

- [29] S. Hamidreza Kasaei et al. “An interactive open-ended learning approach for 3D object recognition”. Em: *2014 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*. 2014, pp. 47–52. DOI: 10.1109/ICARSC.2014.6849761.
- [30] A.E. Johnson e M. Hebert. “Using spin images for efficient object recognition in cluttered 3D scenes”. Em: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21.5 (1999), pp. 433–449. DOI: 10.1109/34.765655.
- [31] Luis A Alexandre. “3D descriptors for object and category recognition: a comparative evaluation”. Em: *Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vilamoura, Portugal*. Vol. 1. 3. Citeseer. 2012, p. 7.
- [32] Federico Tombari, Samuele Salti e Luigi Di Stefano. “A combined texture-shape descriptor for enhanced 3D feature matching”. Em: *2011 18th IEEE International Conference on Image Processing*. 2011, pp. 809–812. DOI: 10.1109/ICIP.2011.6116679.
- [33] Jie Tang et al. “A textured object recognition pipeline for color and depth image data”. Em: *2012 IEEE International Conference on Robotics and Automation*. 2012, pp. 3467–3474. DOI: 10.1109/ICRA.2012.6224891.
- [34] Kevin Lai et al. “A Large-Scale Hierarchical Multi-View RGB-D Object Dataset”. Em: mai. de 2011, pp. 1817–1824. DOI: 10.1109/ICRA.2011.5980382.
- [35] Joona Hänninen. *Using Gocator laser profiling sensor with HALCON machine vision software*. Rel. téc. Seinäjoen ammattikorkeakoulu, 2015.
- [36] *Gocator Line Profile Sensors*. User Manual. Canadá: LMI Technologies, 2022.
- [37] *E6B2-C Data Sheet*. Data Sheet. OMRON, jul. de 2017. URL: <https://www.ia.omron.com/products/family/487/>.



# Apêndices

Apêndice A

Resultados completos dos *models*

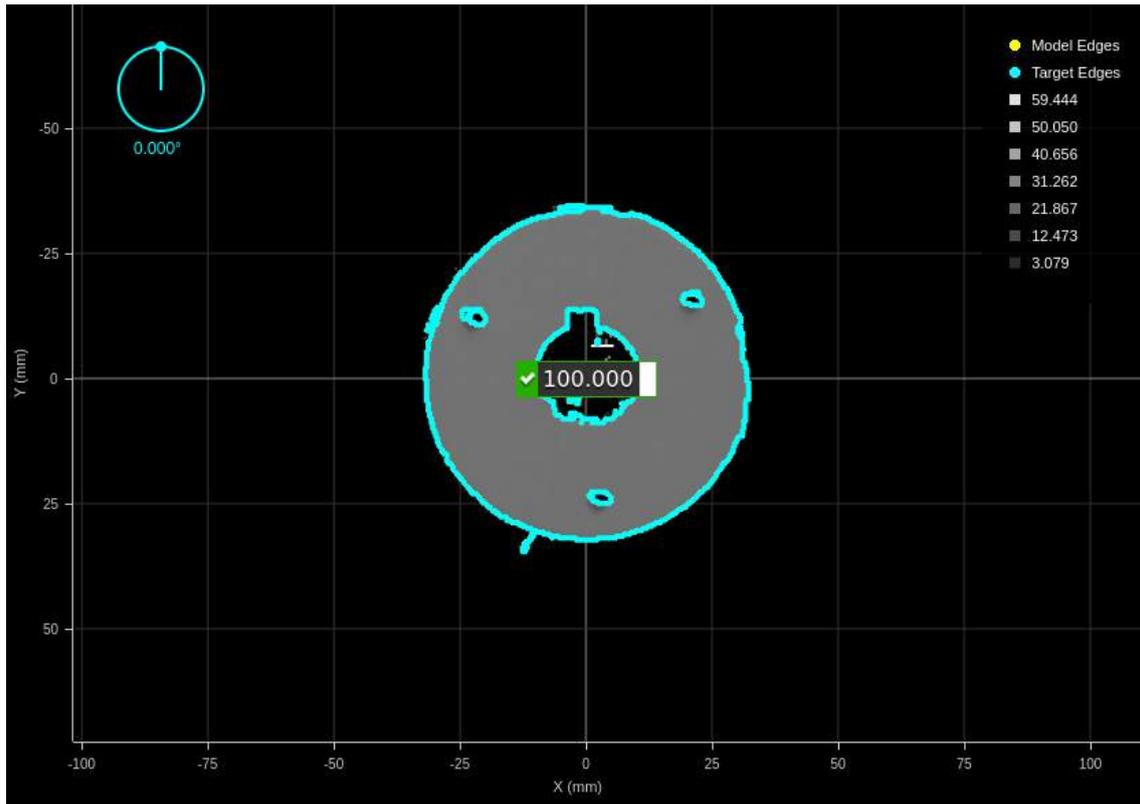


Figura A.1: Peça n.º 1 e peça n.º 1

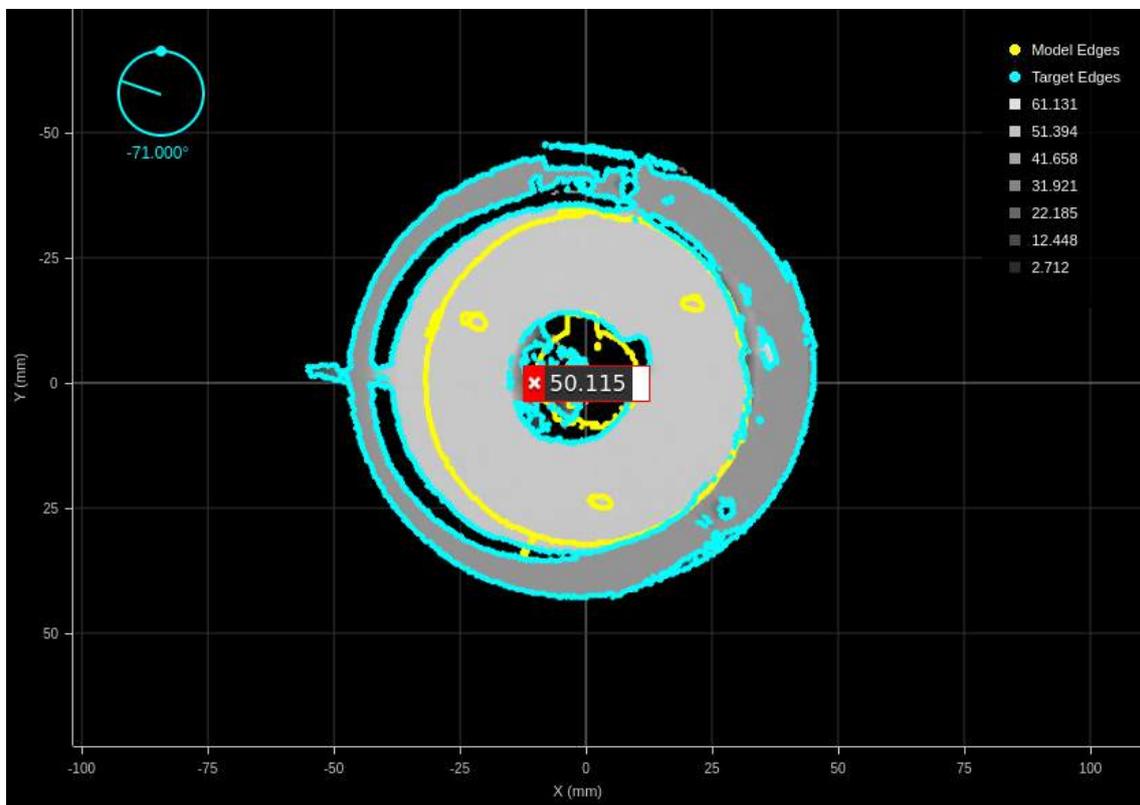


Figura A.2: Peça n.º 1 e peça n.º 2

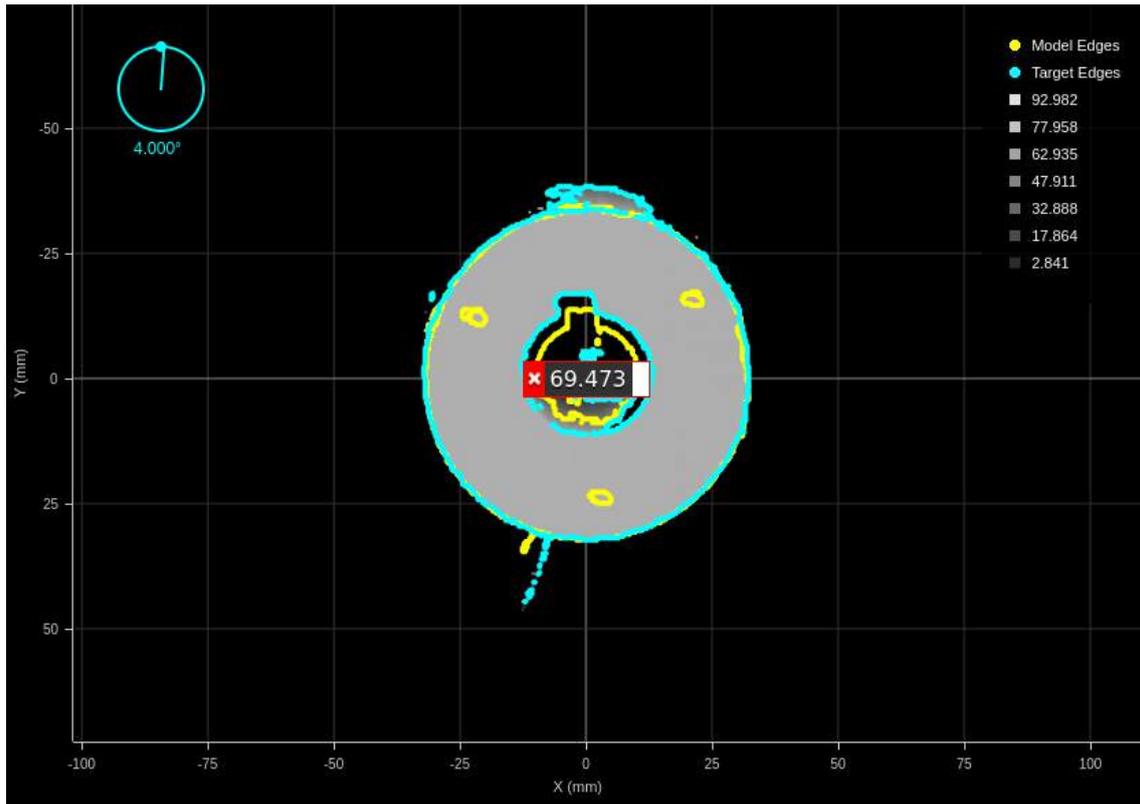


Figura A.3: Peça n.º 1 e peça n.º 6

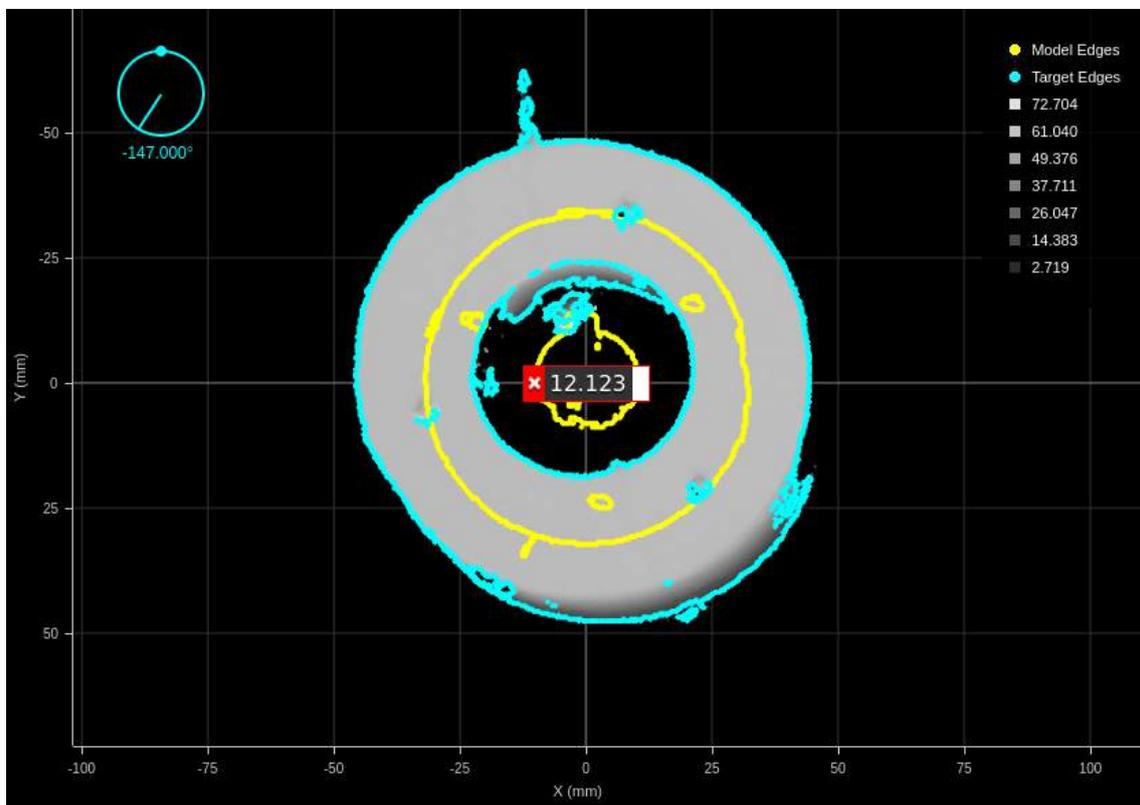


Figura A.4: Peça n.º 1 e peça n.º 8

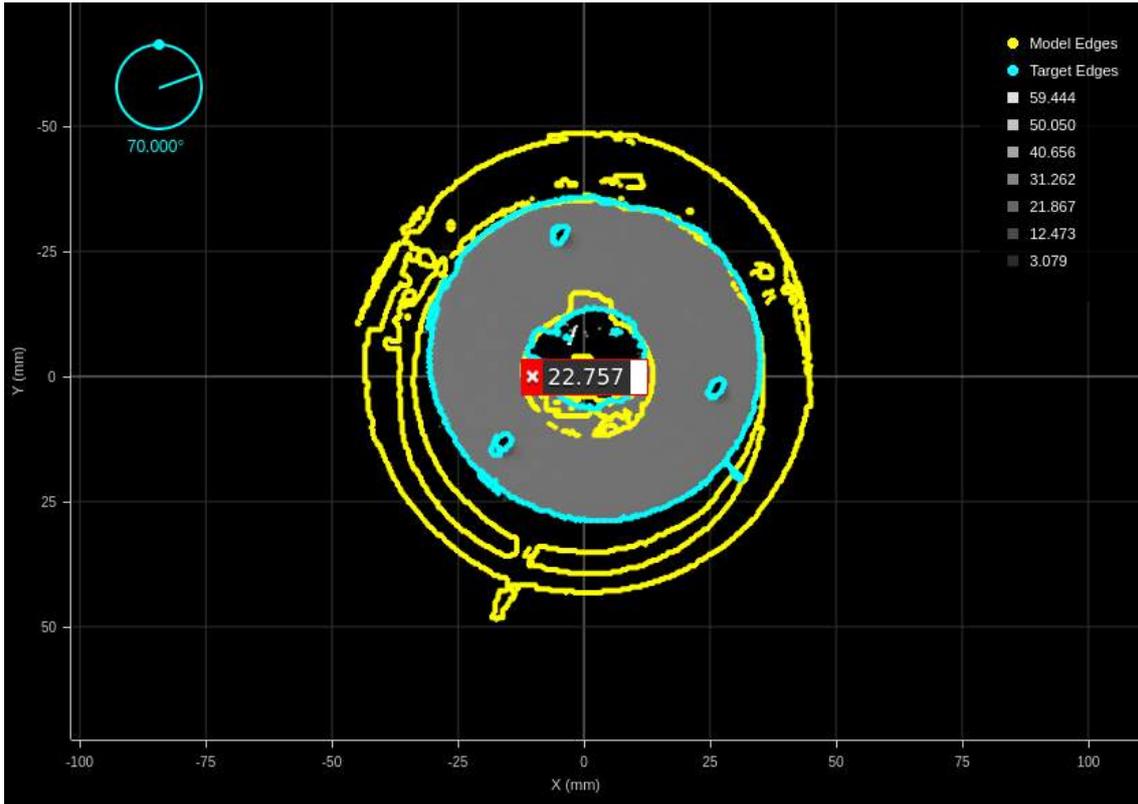


Figura A.5: Peça n.º 2 e peça n.º 1

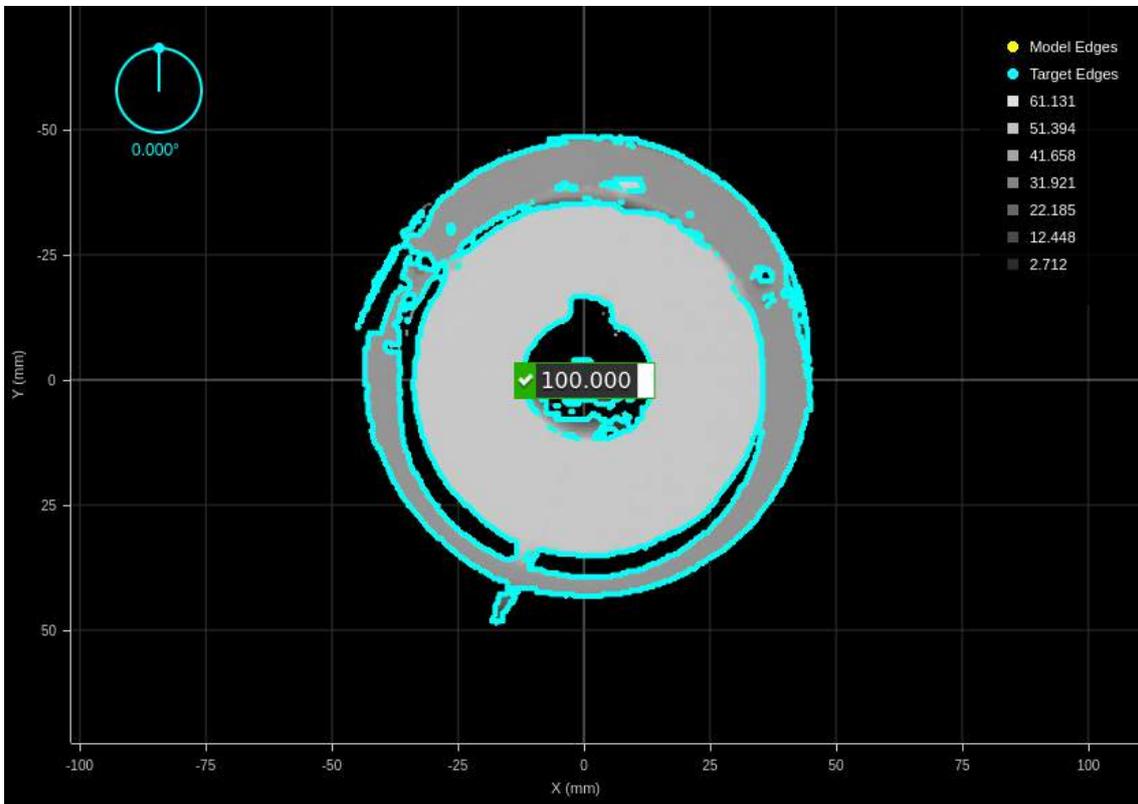


Figura A.6: Peça n.º 2 e peça n.º 2

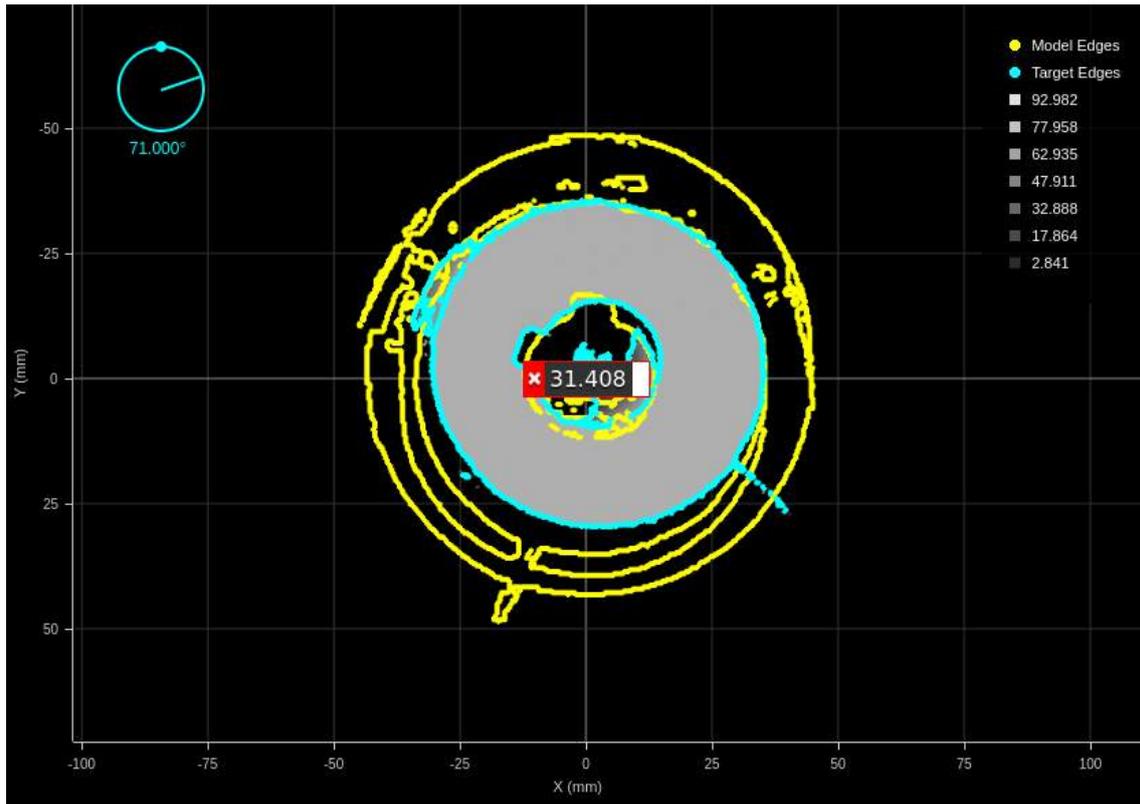


Figura A.7: Peça n.º 2 e peça n.º 6

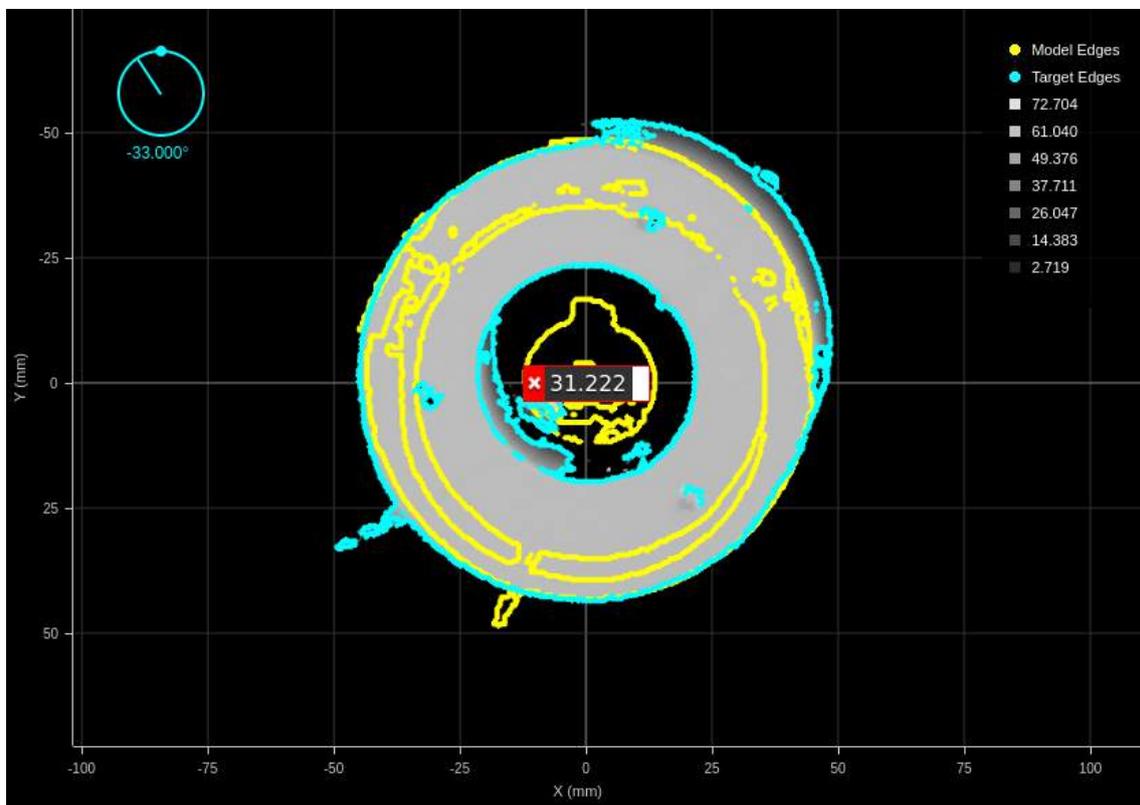


Figura A.8: Peça n.º 2 e peça n.º 8

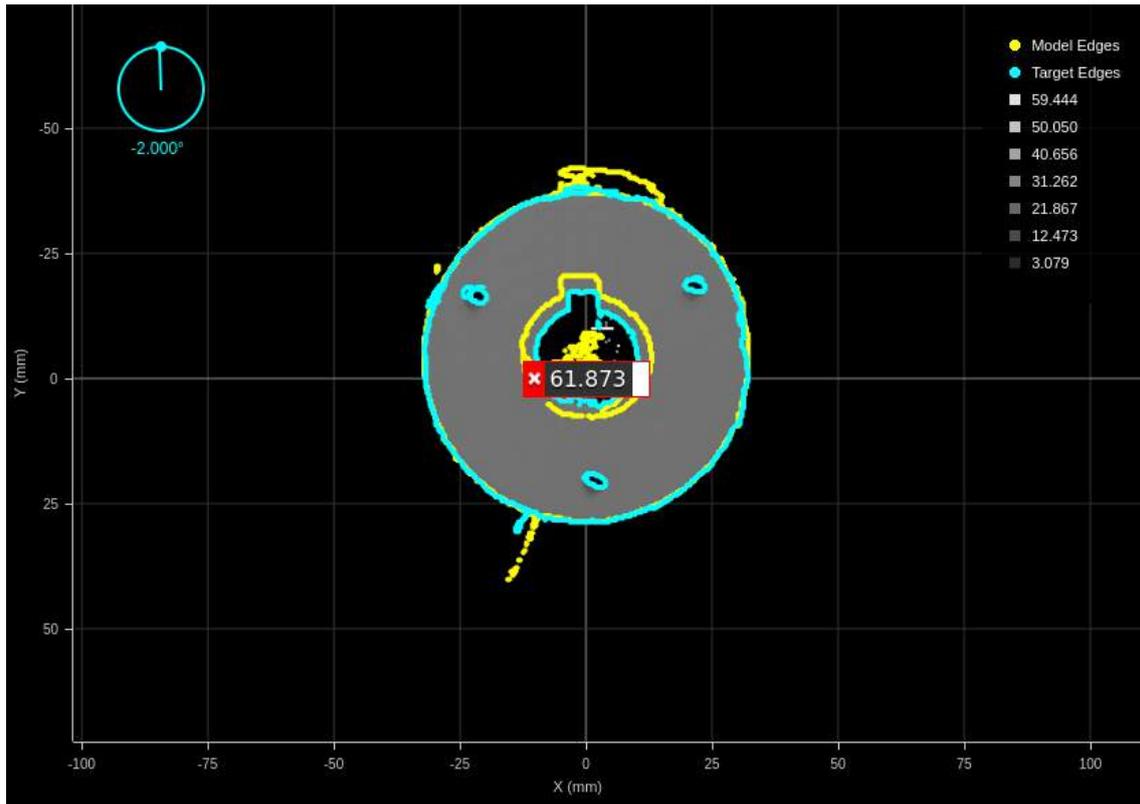


Figura A.9: Peça n.º 6 e peça n.º 1

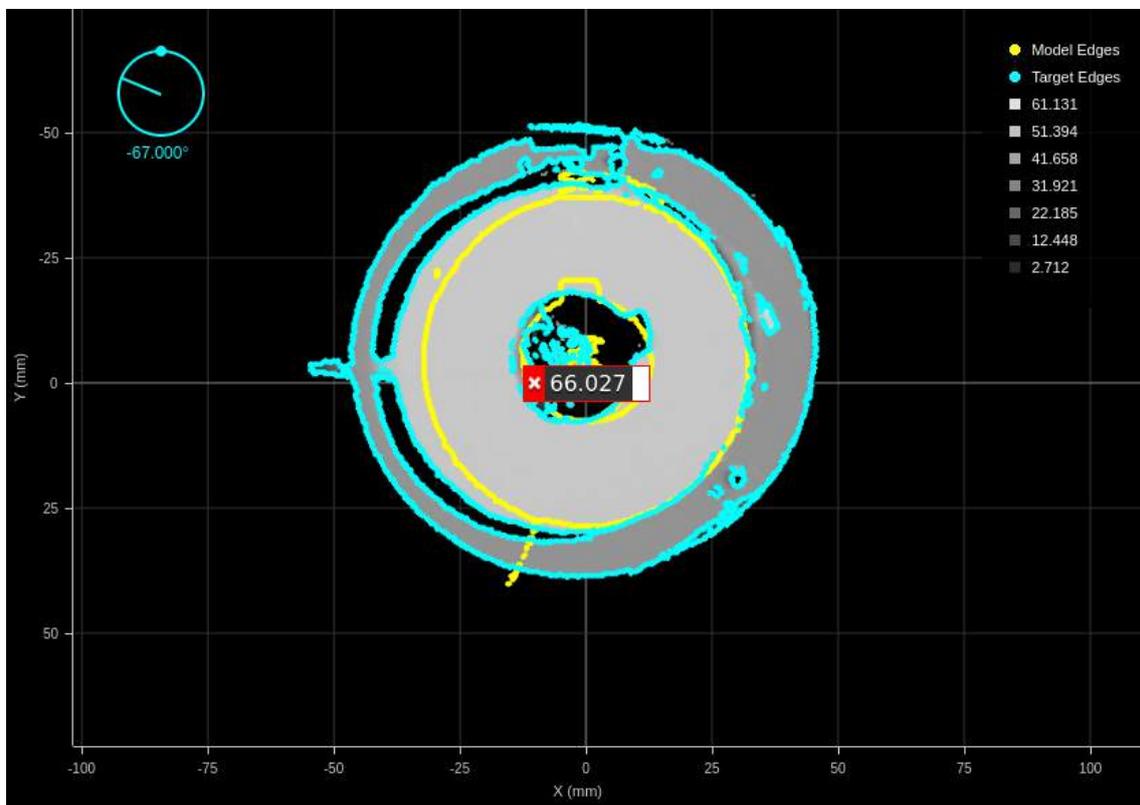


Figura A.10: Peça n.º 6 e peça n.º 2

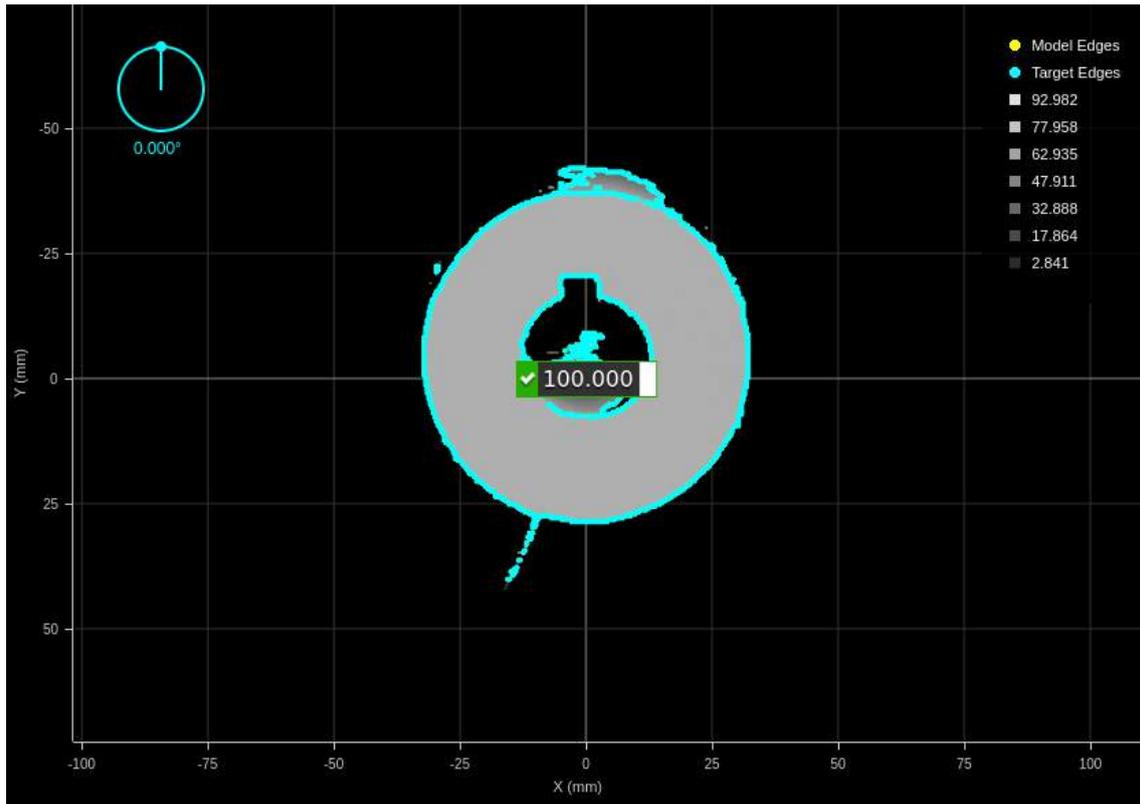


Figura A.11: Peça n.º 6 e peça n.º 6

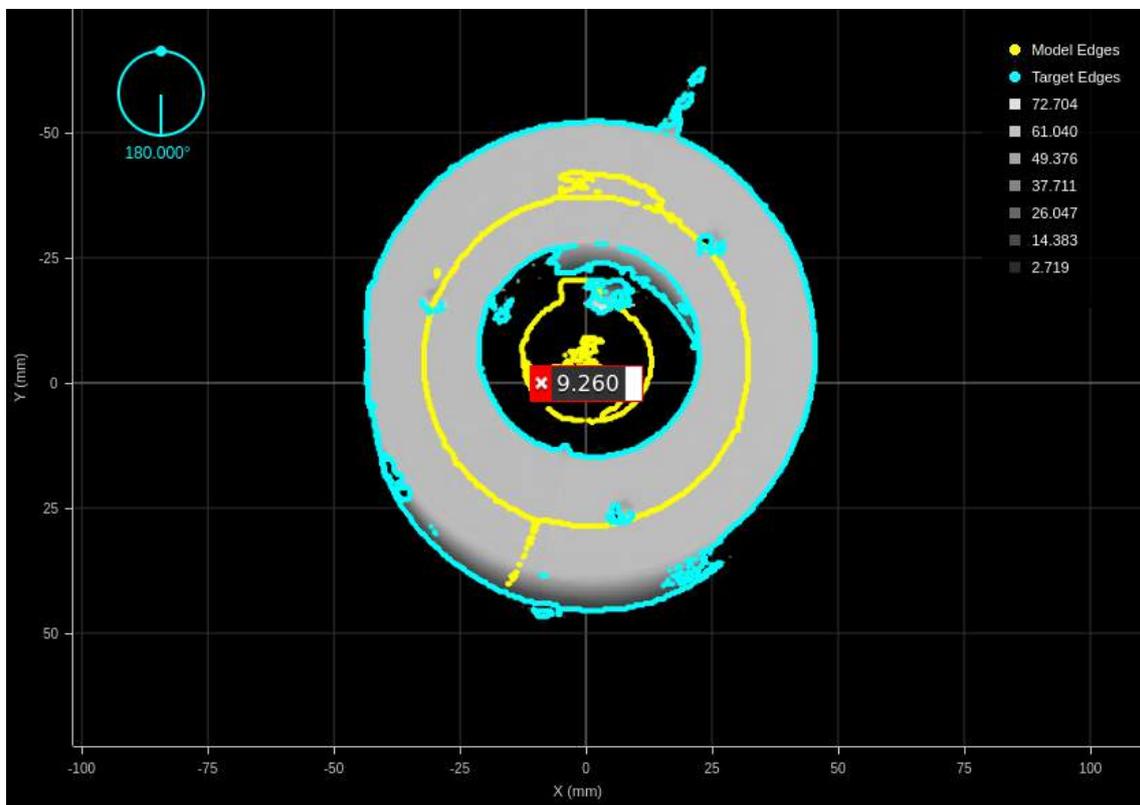


Figura A.12: Peça n.º 6 e peça n.º 8

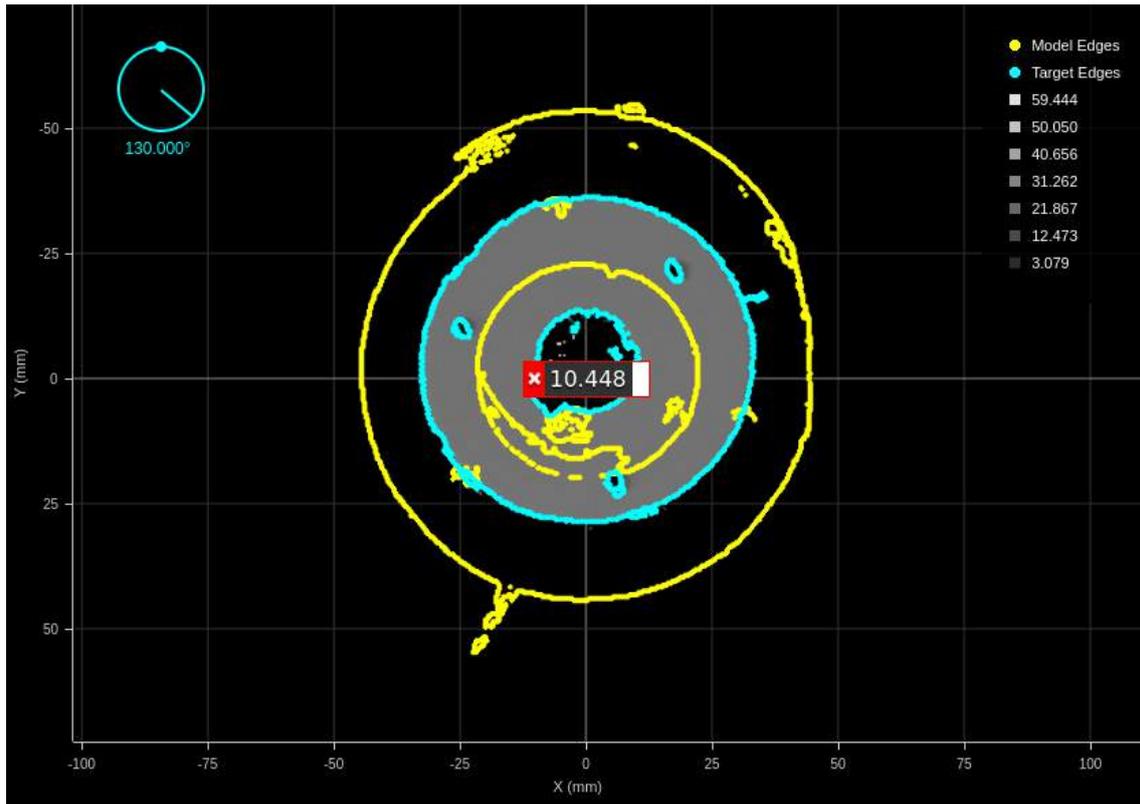


Figura A.13: Peça n.º 8 e peça n.º 1

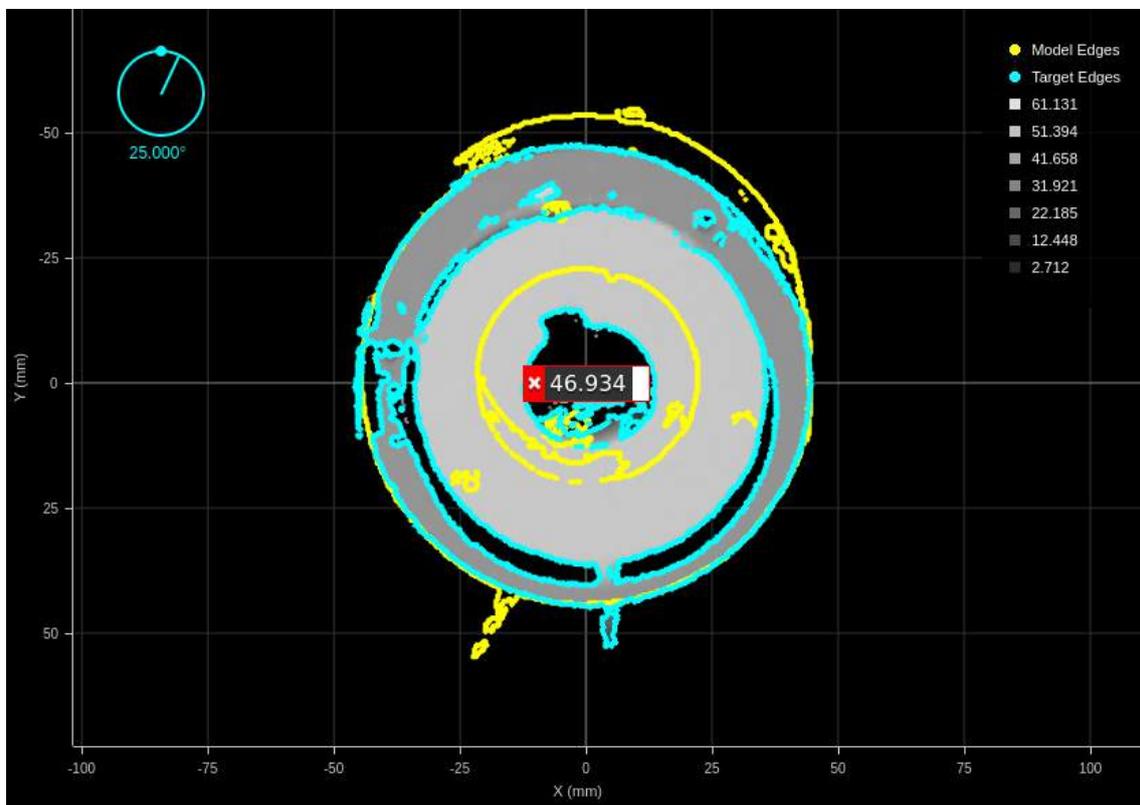


Figura A.14: Peça n.º 8 e peça n.º 2

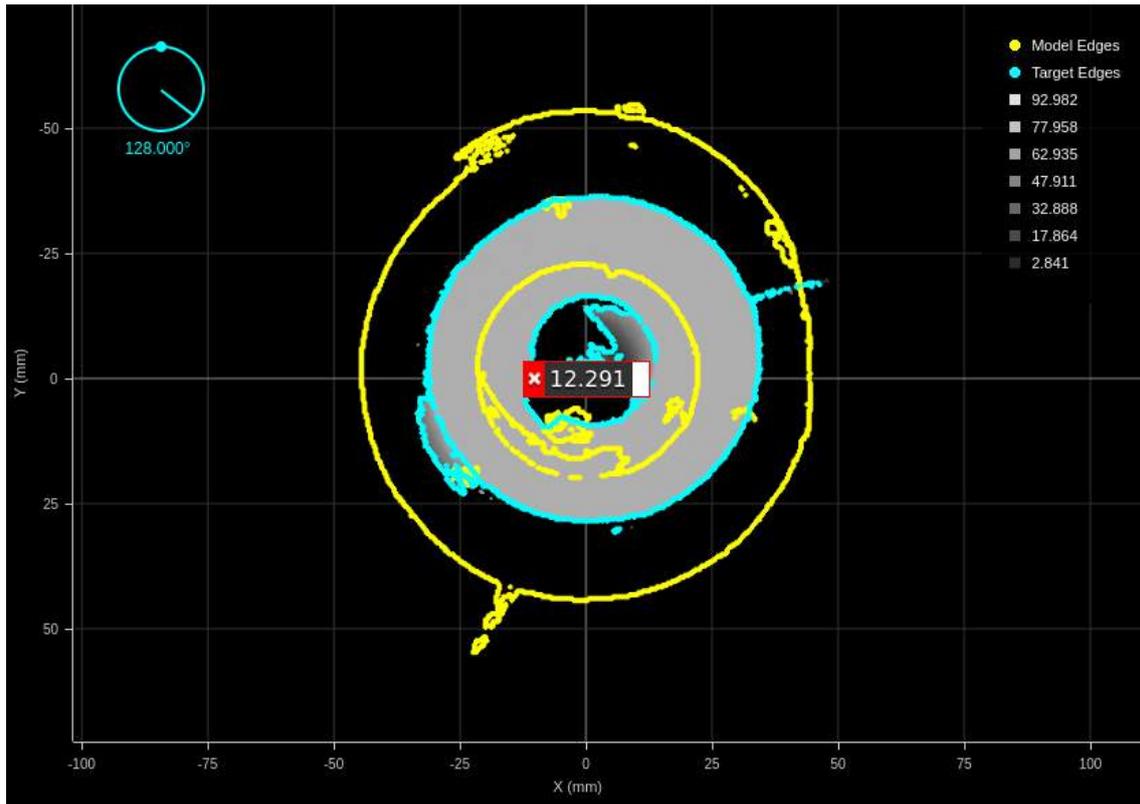


Figura A.15: Peça n.º 8 e peça n.º 6

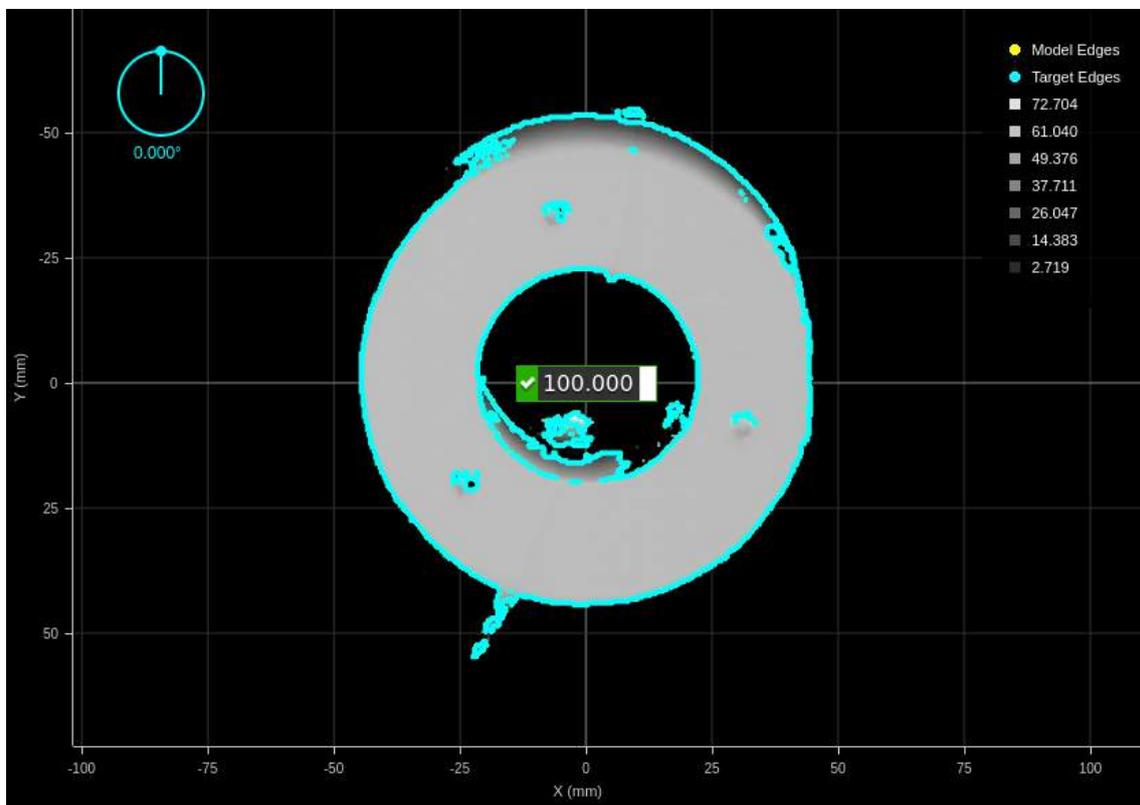
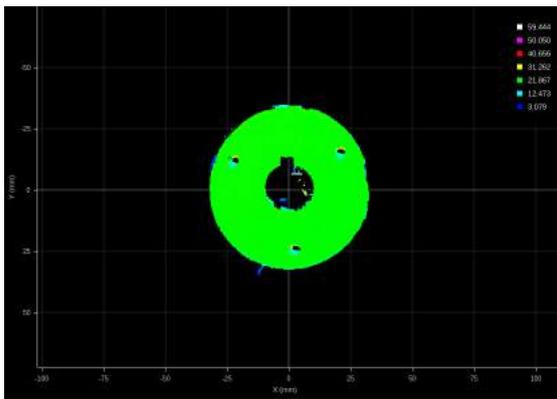
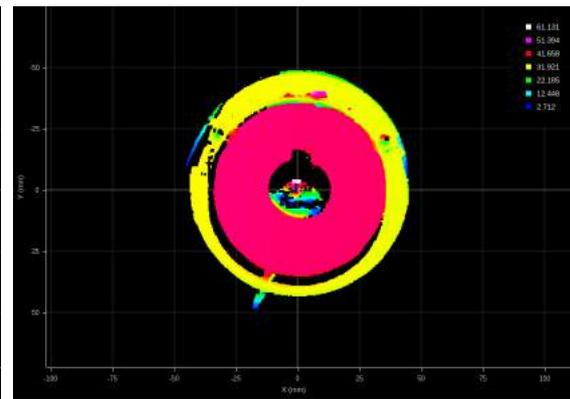


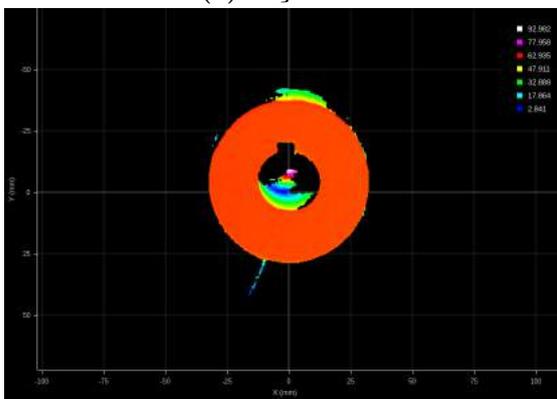
Figura A.16: Peça n.º 8 e peça n.º 8



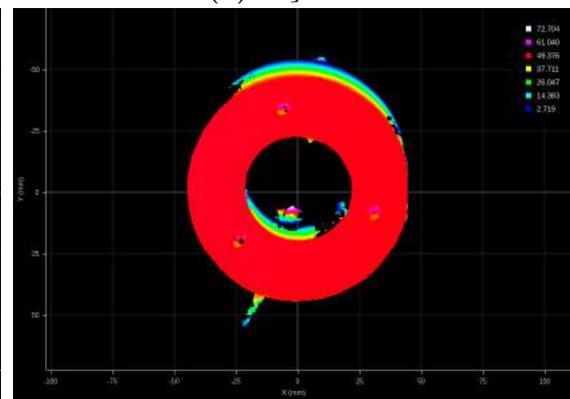
(a) Peça n.º 1



(b) Peça n.º 2



(c) Peça n.º 6

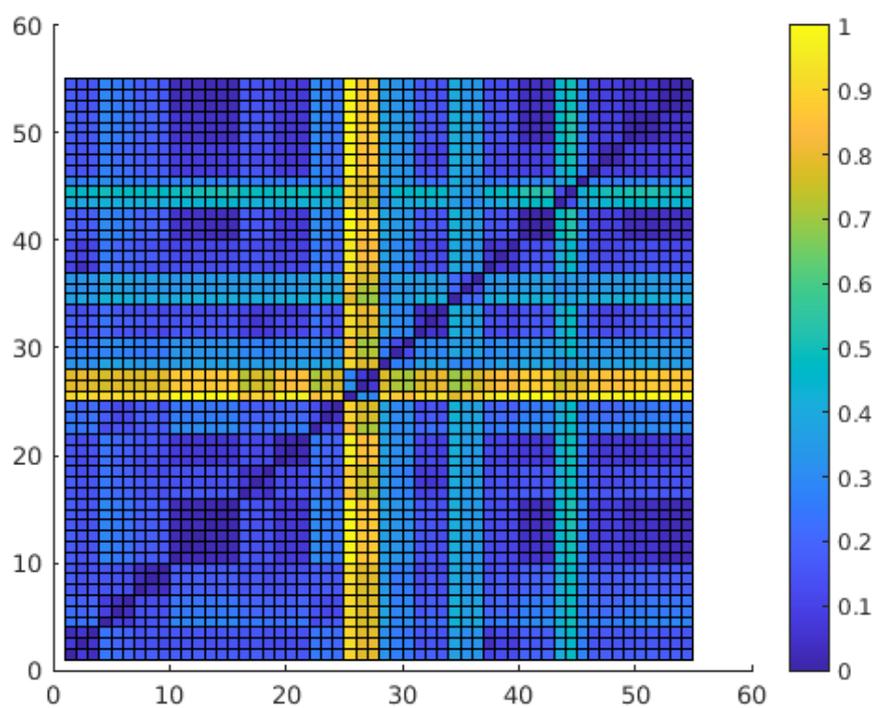


(d) Peça n.º 8

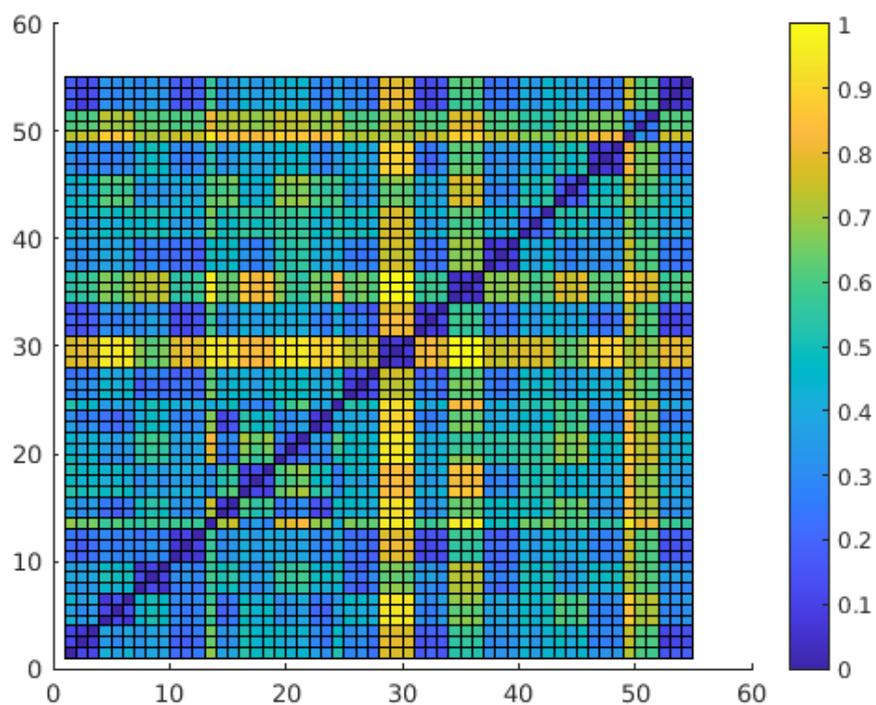
**Figura A.17:** Peças utilizadas para o *matching*

## Apêndice B

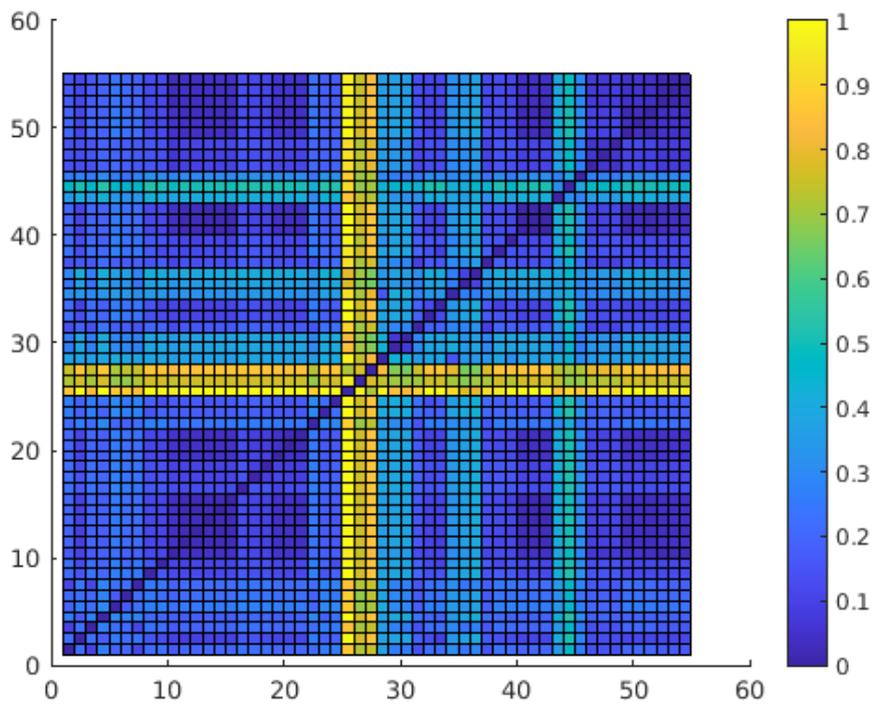
# Matrizes de discriminação completas



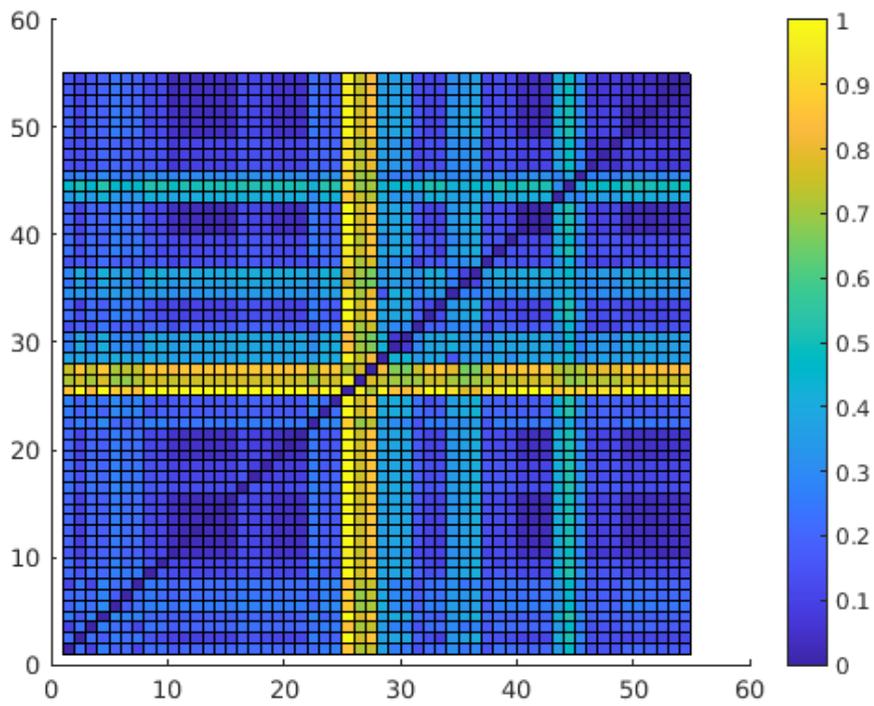
**Figura B.1:** Matriz de discriminação completa do VFH



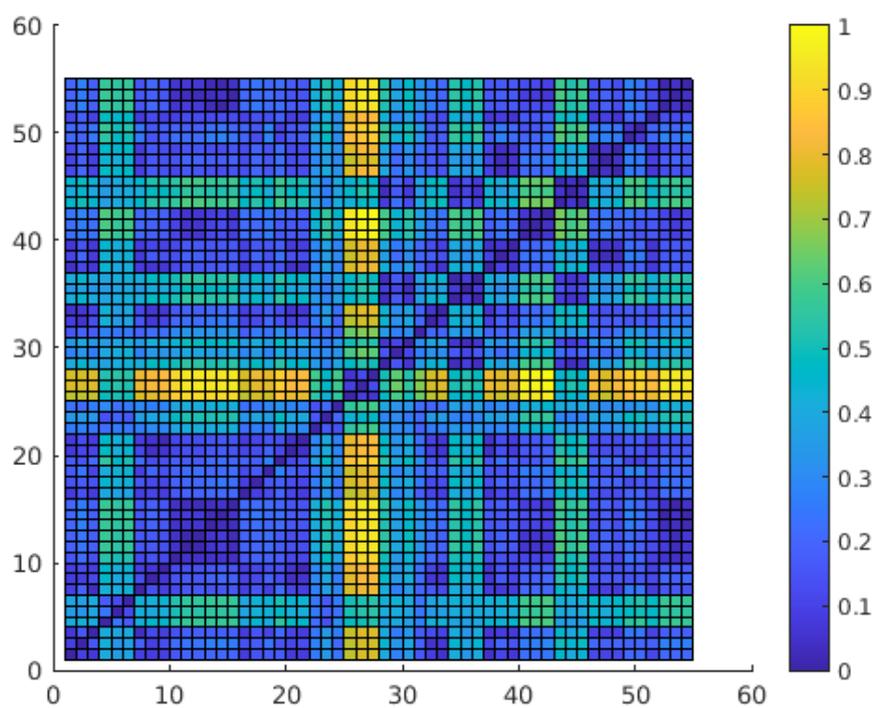
**Figura B.2:** Matriz de discriminação completa do ESF



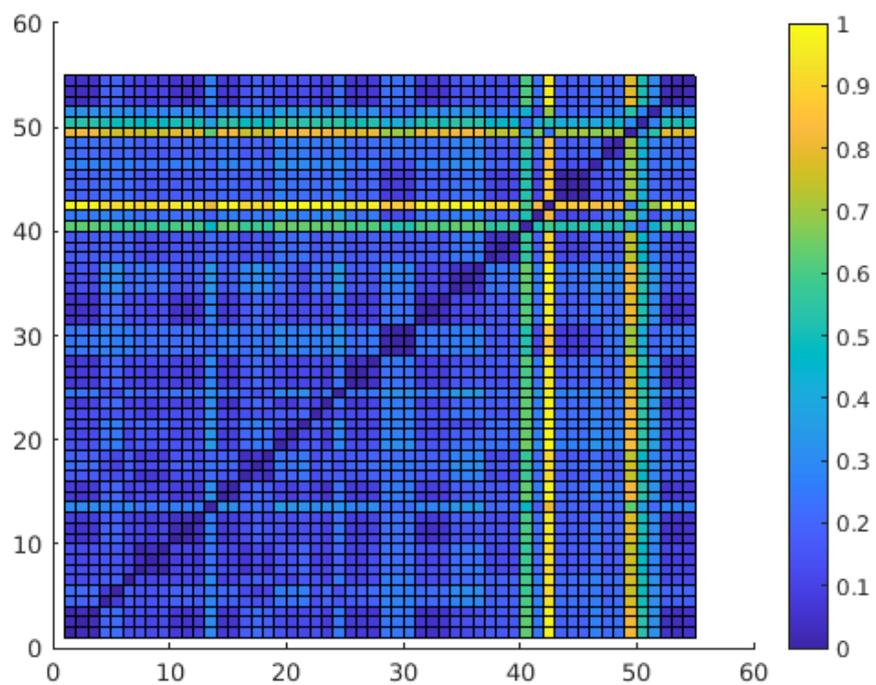
**Figura B.3:** Matriz de discriminação completa do CVFH



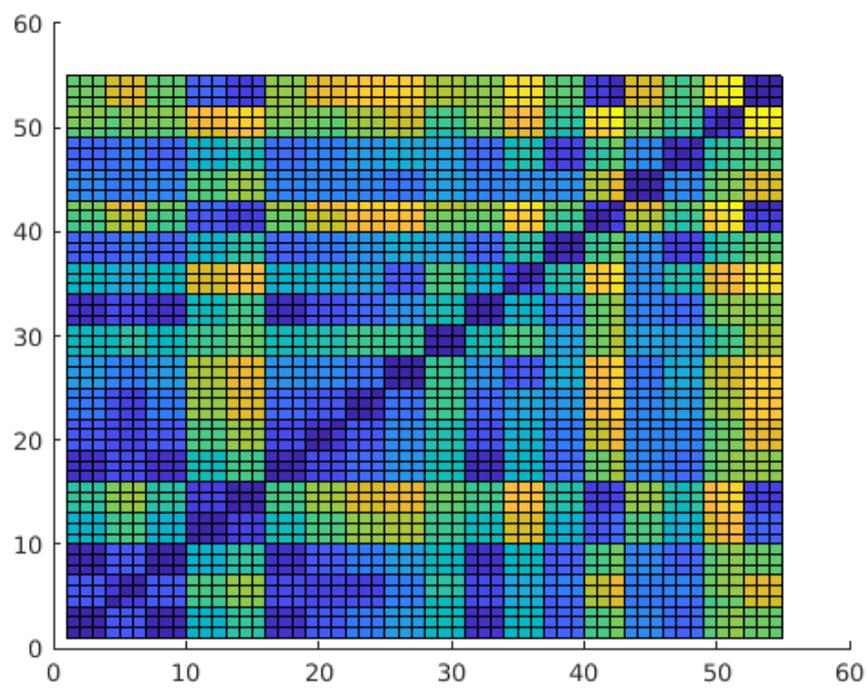
**Figura B.4:** Matriz de discriminação completa do OUR-CVFH



**Figura B.5:** Matriz de discriminação completa do GRSD



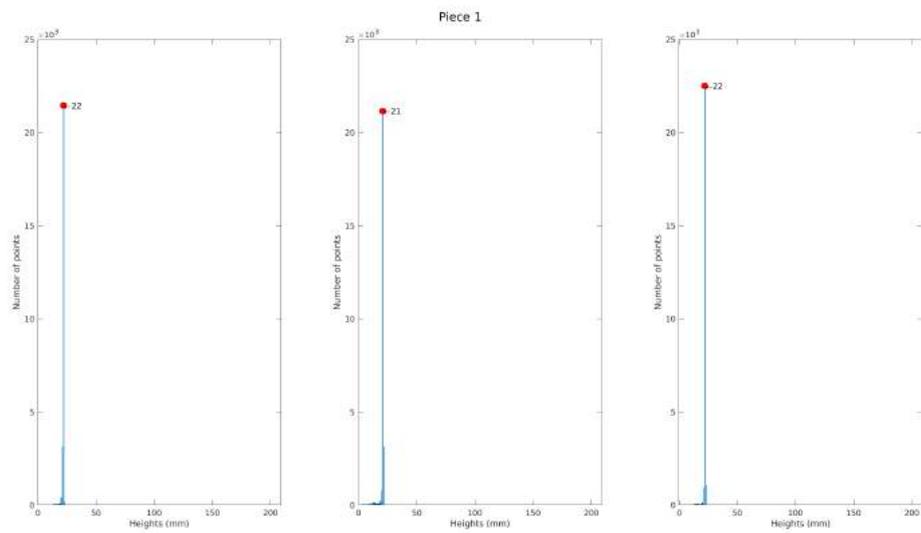
**Figura B.6:** Matriz de discriminação completa do GASD



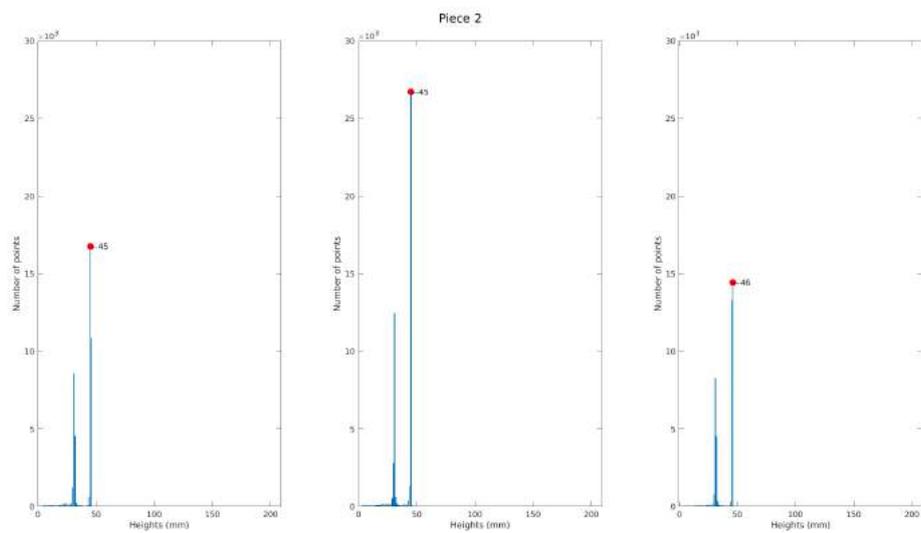
**Figura B.7:** Matriz de discriminação completa do GOOD

## Apêndice C

# Histogramas completos com cotas das peças



**Figura C.1:** Histogramas da peça N.º 1 nas 3 posições do tapete transportador



**Figura C.2:** Histogramas da peça N.º 2 nas 3 posições do tapete transportador

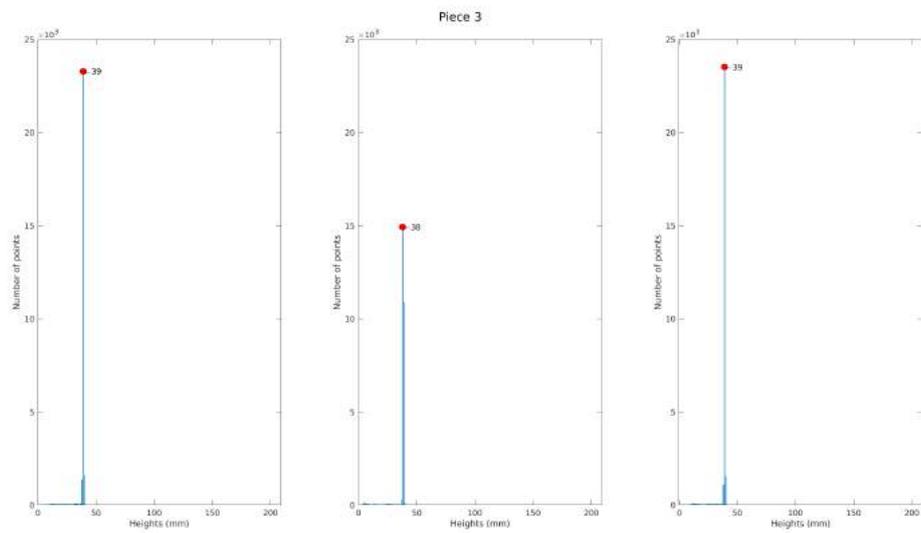


Figura C.3: Histogramas da peça N.º 3 nas 3 posições do tapete transportador

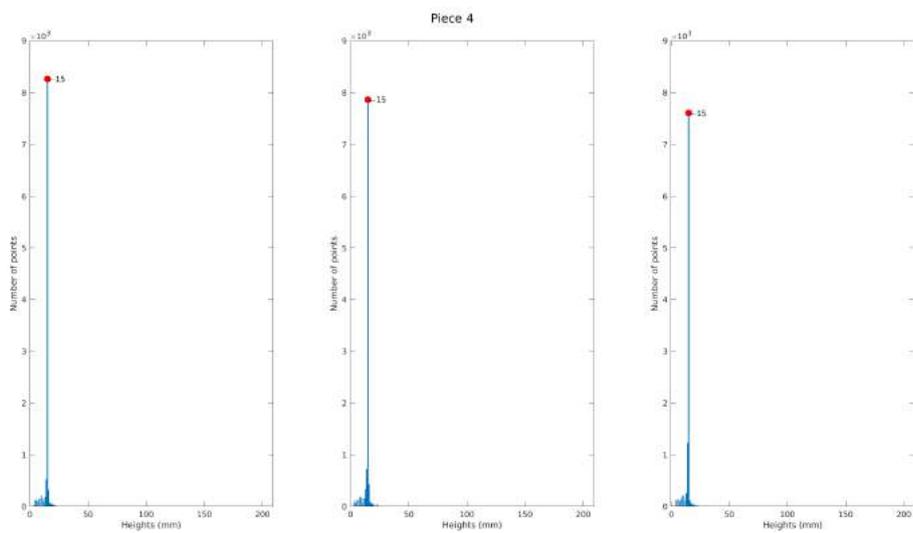


Figura C.4: Histogramas da peça N.º 4 nas 3 posições do tapete transportador

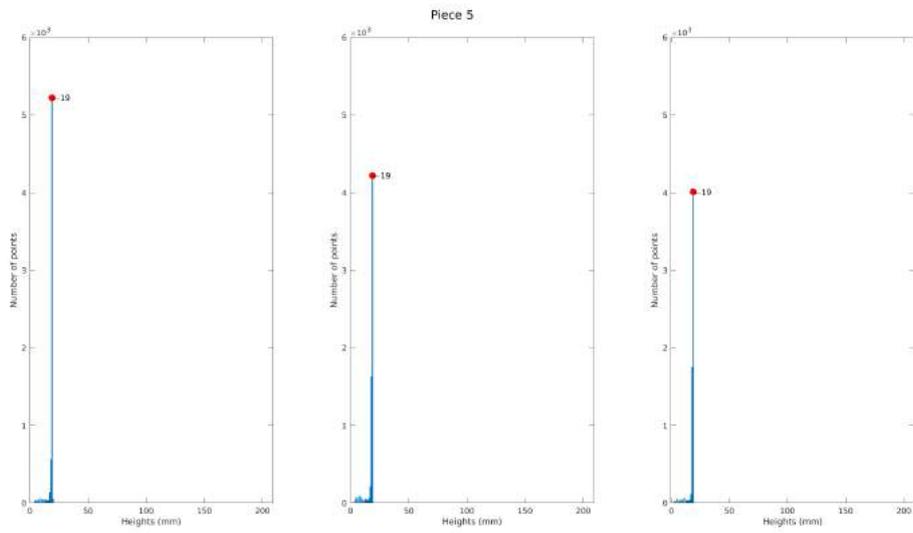


Figura C.5: Histogramas da peça N.º 5 nas 3 posições do tapete transportador

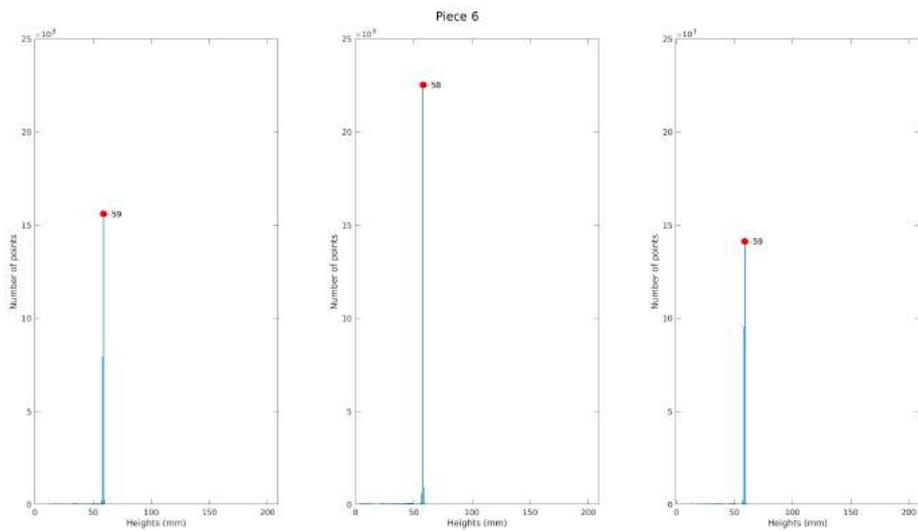


Figura C.6: Histogramas da peça N.º 6 nas 3 posições do tapete transportador

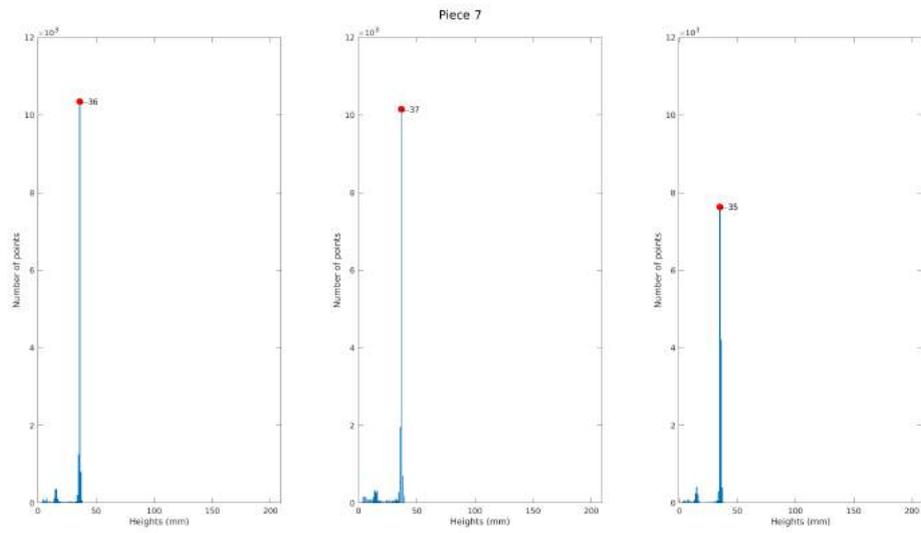


Figura C.7: Histogramas da peça N.º 7 nas 3 posições do tapete transportador

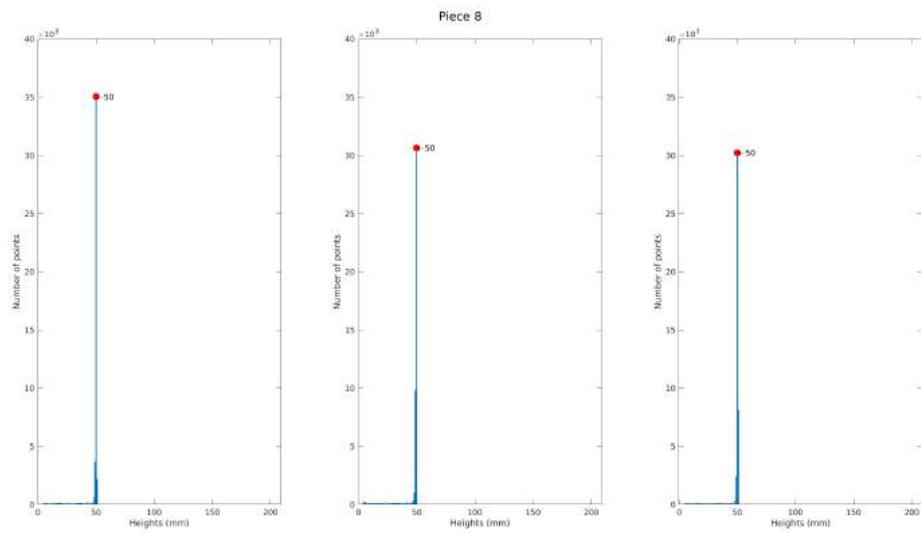
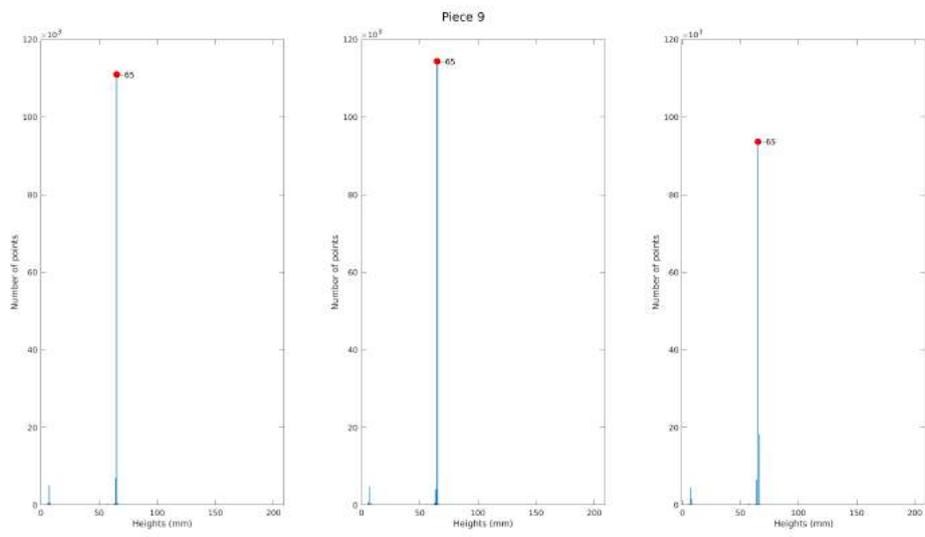
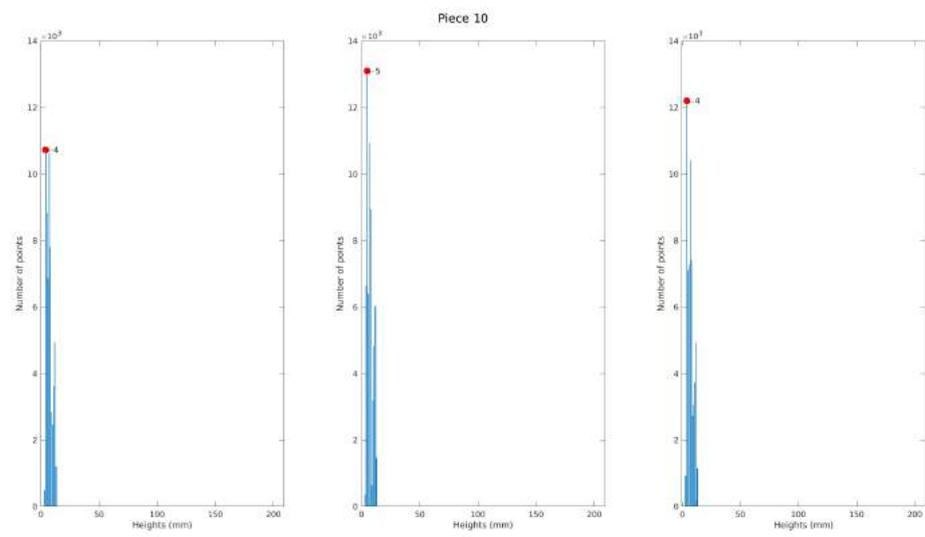


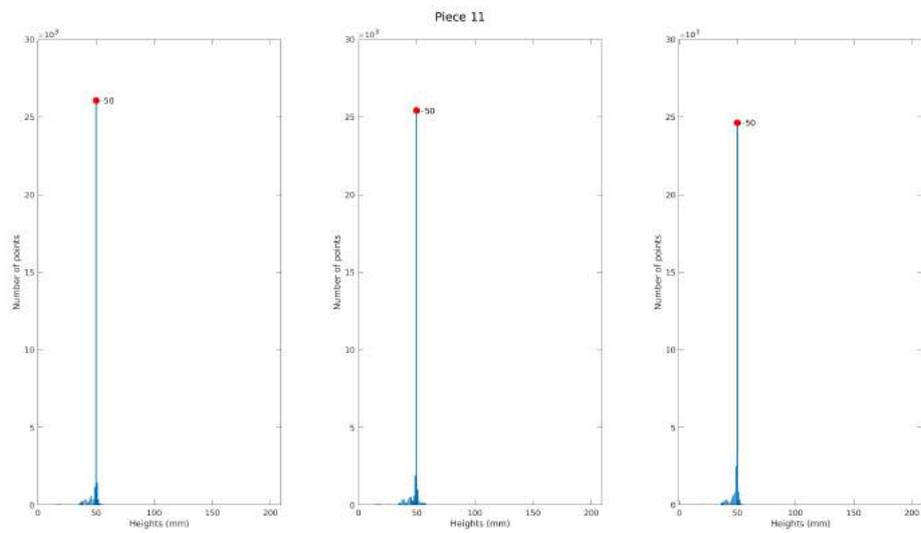
Figura C.8: Histogramas da peça N.º 8 nas 3 posições do tapete transportador



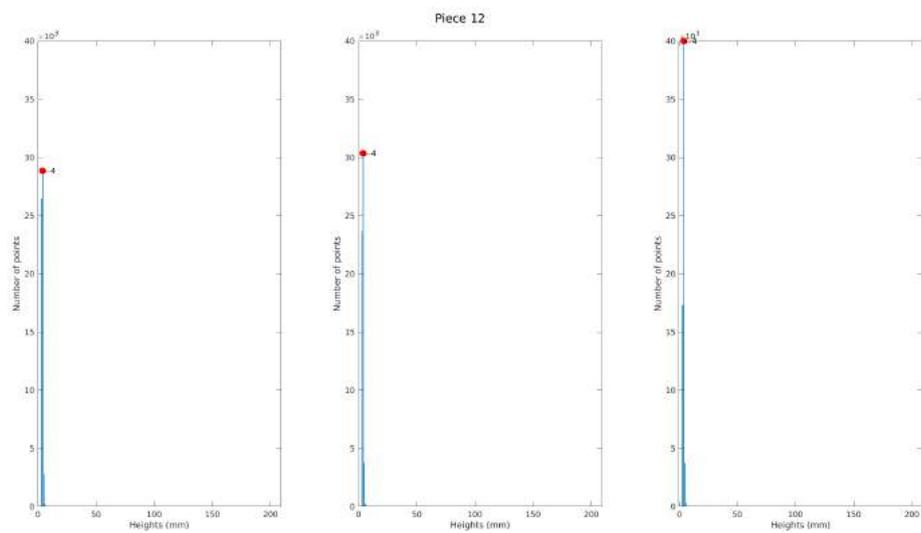
**Figura C.9:** Histogramas da peça N.º 9 nas 3 posições do tapete transportador



**Figura C.10:** Histogramas da peça N.º 10 nas 3 posições do tapete transportador



**Figura C.11:** Histogramas da peça N.º 11 nas 3 posições do tapete transportador



**Figura C.12:** Histogramas da peça N.º 12 nas 3 posições do tapete transportador

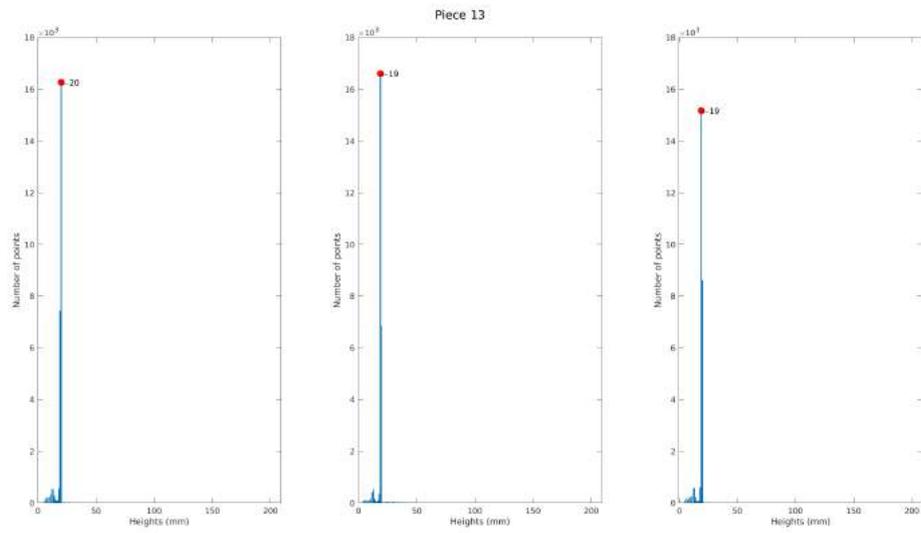


Figura C.13: Histogramas da peça N.º 13 nas 3 posições do tapete transportador

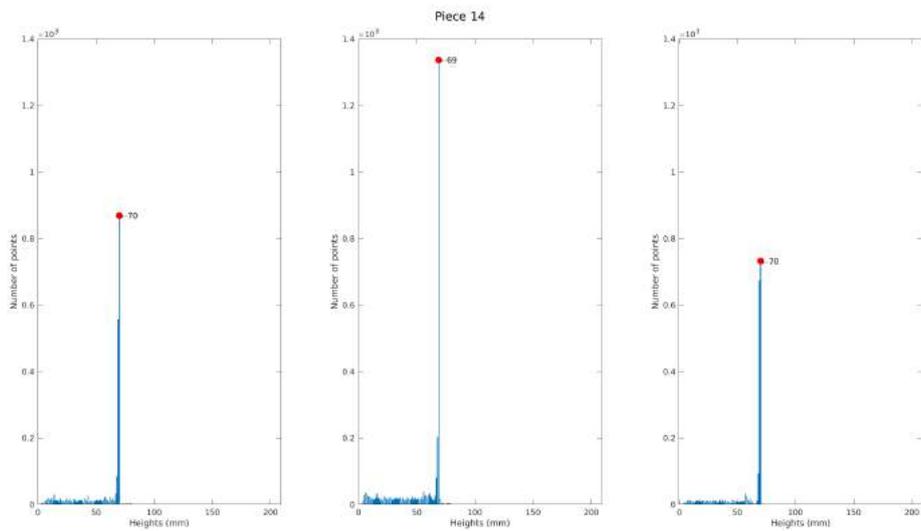
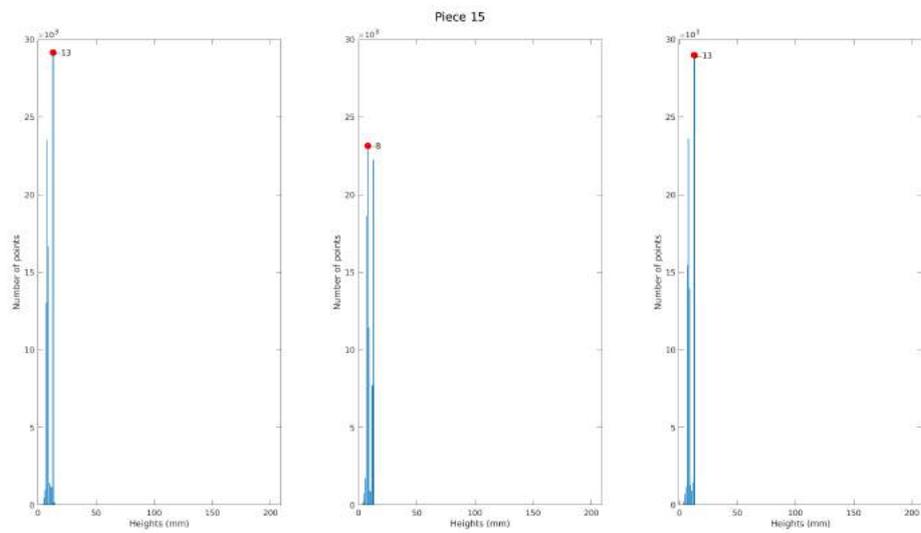
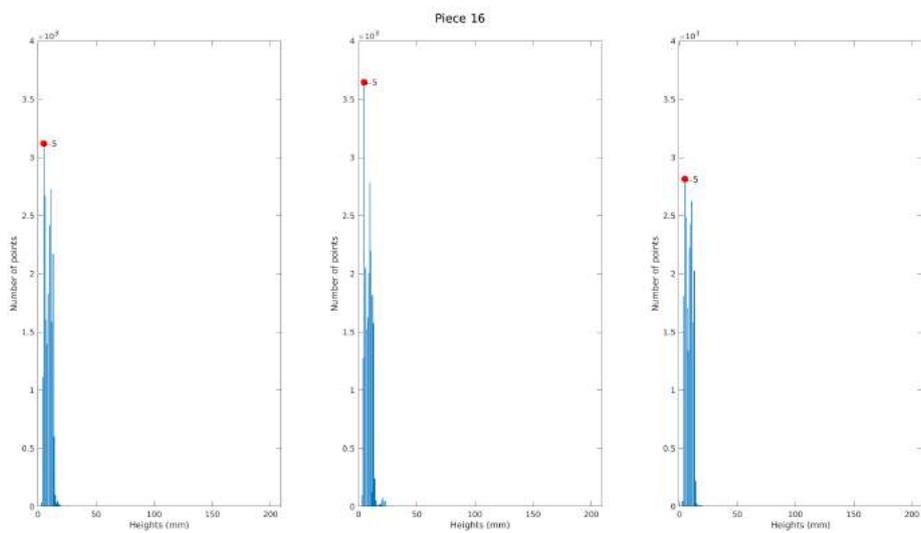


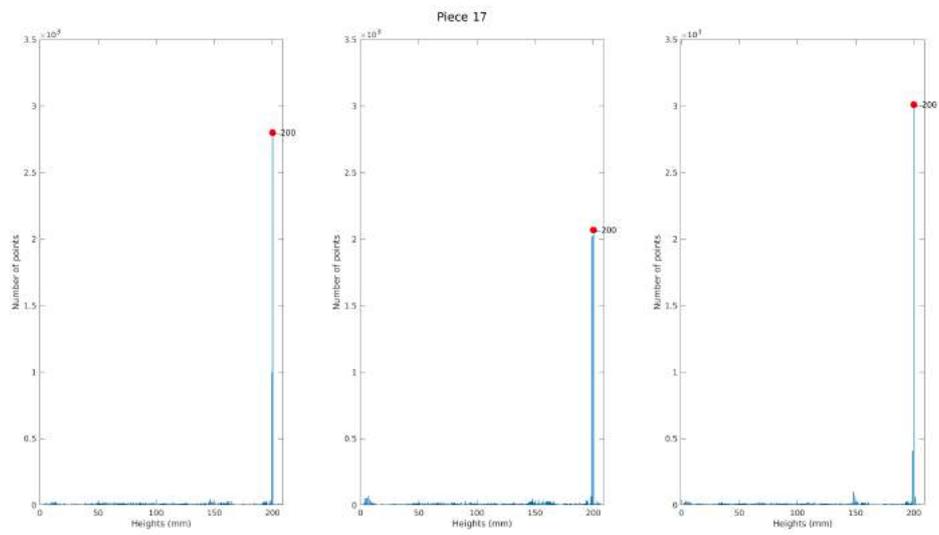
Figura C.14: Histogramas da peça N.º 14 nas 3 posições do tapete transportador



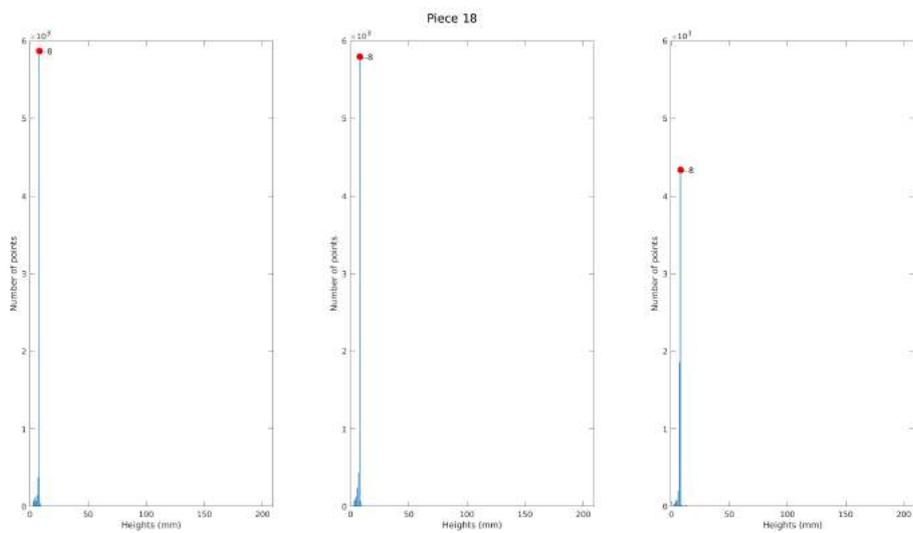
**Figura C.15:** Histogramas da peça N.º 15 nas 3 posições do tapete transportador



**Figura C.16:** Histogramas da peça N.º 16 nas 3 posições do tapete transportador



**Figura C.17:** Histogramas da peça N.º 17 nas 3 posições do tapete transportador



**Figura C.18:** Histogramas da peça N.º 18 nas 3 posições do tapete transportador