

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



# **Sistema de Auxílio à Colagem no Processo de Fundição por Cera Perdida**

**Ricardo Alexandre Duarte Pereira da Silva**

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Orientador: Professor Doutor Armando Jorge Sousa

Co-orientador: Doutor Valter Costa

06 de Julho de 2020



# Resumo

A produção de peças em aço pelo processo de fundição por cera perdida tem uma duração de cerca de duas a três semanas, e envolve diversas tarefas, algumas das quais são feitas de forma manual na empresa Zollern e Comandita Portugal. Os erros cometidos pelos operadores nas tarefas manuais, que só podem ser detetados no fim do processo, faz com que este apresente baixa produtividade. A tarefa de colagem de moldes de cera num tronco é realizada numa das etapas iniciais do processo, e requer o cumprimento de rígidos requisitos no que diz respeito à localização da colagem de cada peça, distância entre peças, etc. Na empresa Zollern e Comandita Portugal, esta tarefa é executada de forma totalmente manual pelos operadores desta empresa, sem qualquer sistema ou guia que os auxilie, pelo que os mesmos são submetidos a um período de aprendizagem de aproximadamente 1 ano antes de serem integrados no processo.

Esta dissertação tem como âmbito o estudo de um sistema que auxilie os operadores da empresa Zollern e Comandita Portugal durante a colagem de moldes de cera num tronco, e respetiva implementação. Este estudo é de elevada relevância, pois com um sistema de apoio nesta tarefa é possível reduzir-se consideravelmente o período de aprendizagem dos operadores, proporcionando também um importante aumento na produtividade global do processo. Devido às características da tarefa, o sistema deve fornecer assistência diretamente no plano de colagem e não deve envolver o uso de um ecrã ou *tablet*, de forma a que operador possa estar apenas focado na execução da colagem. Posto isto, a solução proposta consiste num par câmara-apontador, no qual a câmara tem como função a deteção do tronco e do plano deste onde o operador irá efetuar a colagem, para depois localizar os pontos onde cada peça deve ser colada. Por sua vez, o apontador projetará nos pontos localizados pelo sensor, dando a indicação ao operador para colar naquele local.

A implementação foi testada com dois sensores: Microsoft Kinect V1 e Intel Realsense D435. Os testes realizados permitiram concluir que, entre as duas câmaras, a Microsoft Kinect V1 apresenta melhores resultados, sendo mais indicada para este caso de estudo. Foi, ainda assim, feito um esforço no sentido de melhorar a performance da Intel Realsense D435. Destaca-se que, com a Kinect, registou-se um reduzido aumento do erro relativo à localização da última peça, à medida que as peças vão sendo coladas: entre a primeira medição, com 0 peças no plano, e a medição com todas as peças que são coladas num mesmo plano menos uma, registou-se um aumento no erro de 0,9111 milímetros.

**Palavras-chave:** Manufatura; Processamento de imagem; Sistema de apoio ao operador; nuvem de pontos.



# Abstract

The production of steel pieces by the investment casting method has a duration of two to three weeks, and involves several tasks, some of which are manually done. Mistakes made by operators in manual tasks, which can only be detected at the end of the process, results in low process productivity. The task of gluing wax moulds in a trunk is done in one of the earliest stages of the process, and requires the accomplishment of rigid requirements related to the gluing location of each piece, distance between pieces, etc. At Zollern e Comandita Portugal, this task is manually executed by its operators, without any system or guidance that helps them, reason why they are submitted to a learning period of approximately 1 year before their integration on the process.

This dissertation has the scope of studying a system that helps Zollern e Comandita Portugal operators through the gluing of wax moulds on the trunk, and the respective implementation. This study is highly relevant because with a supporting system it is possible to significantly reduce the operators learning period, while providing an important increase in the process global productivity. Due to the task characteristics, the system should provide assistance directly onto the gluing plane and the operator should not need to use any screen or tablet, so that he can only focus on the gluing action. Therefore, the proposed solution consists on a camera-pointer pair, in which the camera has the objective of detecting both trunk and plane where the gluing action will be performed, to later localize the points where each piece must be glued. On the other hand, the pointer will project over the points localized by the sensor, giving the operator the indication to glue in that specific location.

The implementation was tested with two sensors: Microsoft Kinect V1 and Intel Realsense D435. With these tests it was possible to conclude that, between the two cameras, the Microsoft Kinect V1 presents better results, being more indicated for this case study. Yet, an effort was made in order to improve Intel Realsense D435 performance. It stands out that, with Kinect, a small increase related to the error accumulation of the last piece gluing point location was registered, while more pieces are glued: between the first measurement, with 0 pieces over the plane, and the measurement with all pieces but one, it was registered an error increase of 0,9111 millimeters.

**Keywords:** Manufacturing; Image Processing; Operator Support System; Point Cloud



# Agradecimentos

Em primeiro lugar, gostaria de agradecer ao Professor Armando Sousa e ao Doutor Valter Costa, por toda a orientação e apoio que me deram durante a realização deste trabalho.

Gostaria de agradecer também a toda a minha família, por me terem acompanhado e motivado, não só durante este trabalho mas ao longo de todo o meu percurso académico e pessoal, e em especial aos meus pais, pelos valores e educação que me transmitiram. É certamente graças ao constante carinho, motivação e apoio que me deram que eu cheguei a esta fase.

Por último, agradeço também a todos os meus amigos, que sempre estiveram presentes quer nos momentos mais fáceis, quer nos mais difíceis.

Ricardo Alexandre Duarte Pereira da Silva

## Agradecimentos Institucionais

Este trabalho foi desenvolvido no âmbito do projeto com a referência POCI-01-0145-FEDER-032460 – GRIS: Refinamento de Grão Induzido por Agitação Eletromagnética: Desenvolvimento de uma Abordagem Preditiva para Implantes Médicos em Co-Cr, cofinanciado pelo Programa Operacional Competitividade e Internacionalização (COMPETE 2020), através do Fundo Europeu de Desenvolvimento Regional (FEDER) e por Fundos Nacionais através da Fundação para a Ciência e Tecnologia (FCT).



Cofinanciado por:





# Conteúdo

|  |             |
|--|-------------|
| <b>Resumo</b>  | <b>i</b>    |
| <b>Abstract</b>  | <b>iii</b>  |
| <b>Agradecimentos</b>  | <b>v</b>    |
| <b>Abreviaturas</b>  | <b>xiii</b> |
| <b>1 Introdução</b>  | <b>1</b>    |
| 1.1 Contexto . . . . .   | 1           |
| 1.2 Motivação . . . . .  | 2           |
| 1.3 Objetivos . . . . .  | 3           |
| 1.4 Estrutura do documento . . . . .                           | 3           |
| <b>2 Estado da Arte</b>  | <b>5</b>    |
| 2.1 Revisão da literatura . . . . .                            | 5           |
| 2.1.1 Sistemas de Apoio ao Operador . . . . .                  | 5           |
| 2.1.2 Projection Mapping . . . . .                             | 13          |
| 2.2 Estudo do mercado . . . . .                                | 22          |
| 2.2.1 Empresas produtoras de sistemas . . . . .                | 22          |
| 2.2.2 Empresas produtoras e consumidoras de sistemas . . . . . | 26          |
| 2.3 Sumário . . . . .  | 26          |
| <b>3 Escolha do Hardware e das ferramentas de Software</b>     | <b>29</b>   |
| 3.1 Hardware . . . . .   | 29          |
| 3.1.1 Sensor . . . . .   | 29          |
| 3.1.2 Atuador . . . . .  | 31          |
| 3.2 Ferramentas de Software . . . . .                          | 33          |
| 3.2.1 ROS (Robot Operating System) . . . . .                   | 33          |
| 3.2.2 PCL (Point Cloud Library) . . . . .                      | 34          |
| 3.2.3 Meshlab . . . . .  | 34          |
| 3.2.4 OpenCV . . . . .   | 35          |
| 3.2.5 Arduino . . . . .  | 35          |
| 3.3 Sumário . . . . .  | 35          |
| <b>4 Implementação</b>   | <b>37</b>   |
| 4.1 Definições . . . . .                                       | 37          |
| 4.2 Arquitetura do Sistema . . . . .                           | 38          |
| 4.3 Processamento da nuvem de pontos da câmara . . . . .       | 39          |

|          |  |           |
|----------|--|-----------|
| 4.3.1    | Deteção do plano de colagem . . . . .                | 41        |
| 4.3.2    | Localização dos pontos de colagem . . . . .          | 44        |
| 4.3.3    | Cálculo da próxima peça a ser colada . . . . .       | 46        |
| 4.4      | Projeção do laser nos pontos de colagem . . . . .    | 47        |
| 4.4.1    | Deteção do ponto de projeção do laser . . . . .      | 47        |
| 4.4.2    | Controlo dos servos . . . . .                        | 48        |
| 4.5      | Sumário . . . . .                                    | 51        |
| <b>5</b> | <b>Resultados</b>                                    | <b>53</b> |
| 5.1      | Processamento da nuvem de pontos da câmara . . . . . | 53        |
| 5.2      | Projeção do laser nos pontos pretendidos . . . . .   | 57        |
| 5.3      | Comparação com Intel Realsense D435 . . . . .        | 59        |
| 5.4      | Sumário . . . . .                                    | 65        |
| <b>6</b> | <b>Conclusões</b>                                    | <b>67</b> |
| 6.1      | Trabalho Futuro . . . . .                            | 69        |
|          | <b>Bibliografia</b>                                  | <b>71</b> |

# Lista de Figuras

|      |  |    |
|------|--|----|
| 1.1  | Algumas tarefas da produção de peças em aço pelo processo de fundição por cera perdida. . . . .  | 2  |
| 2.1  | Distinção entre dispositivos, no que diz respeito à incorporação da tecnologia no usuário. . . . .   | 6  |
| 2.2  | Diferença entre tecnologias que dão ao utilizador a sensação de estar onde ele de facto está e tecnologias que o inserem num ambiente virtual. . . . . | 7  |
| 2.3  | Exemplos de tecnologias com baixo grau de interatividade com o utilizador e de tecnologias com elevado grau de interatividade com o mesmo. . . . .     | 7  |
| 2.4  | Projeção clássica num ecrã plano vs Projection Mapping. . . . .  | 13 |
| 2.5  | Modelo matemático da câmara e do projetor. . . . .   | 15 |
| 2.6  | Iteração do algoritmo de calibração e resultado final pretendido . . . . .   | 18 |
| 2.7  | Relação entre os pontos do padrão de xadrez projetados e os pontos impressos. . . . .  | 19 |
| 2.8  | Algoritmo para a calibração do projetor. . . . .   | 21 |
| 4.1  | Arquitetura do sistema. . . . .  | 38 |
| 4.2  | Diagrama de sequência UML do sistema. . . . .  | 39 |
| 4.3  | Algoritmo de processamento da nuvem de pontos. . . . .   | 40 |
| 4.4  | nuvem de pontos contendo apenas a bacia. . . . .   | 42 |
| 4.5  | Alinhamento da bacia na nuvem de pontos filtrada. . . . .  | 42 |
| 4.6  | Alinhamento do cilindro na bacia. . . . .  | 43 |
| 4.7  | Planos do tronco detetados pela câmara. . . . .  | 44 |
| 4.8  | Eixos formados pelas arestas do plano de colagem. . . . .  | 45 |
| 4.9  | Plano de colagem transformado pelo novo sistema de coordenadas. . . . .  | 46 |
| 4.10 | Nuvem do plano de colagem, com 0 e 5 peças coladas . . . . .   | 47 |
| 4.11 | Algoritmo para deteção do ponto de projeção atual do ponteiro laser. . . . .   | 47 |
| 4.12 | Algoritmo para controlo dos servos. . . . .  | 48 |
| 5.1  | Número de pontos da nuvem do plano de colagem, com papel. . . . .  | 55 |
| 5.2  | Número de pontos da nuvem do plano de colagem, sem papel (superfície de cera). . . . .   | 55 |
| 5.3  | Erro angular do vetor normal do plano, a 0 graus. . . . .  | 56 |
| 5.4  | Erro angular do vetor normal do plano, com este inclinado a 25 graus nos dois sentidos . . . . .   | 56 |
| 5.5  | Tronco de colagem, visto de diferentes perspetivas . . . . .   | 60 |
| 5.6  | Deteção do plano de colagem, utilizando a câmara Realsense D435. . . . .   | 61 |
| 5.7  | Deteção do plano de colagem, utilizando a câmara Realsense D435. . . . .   | 61 |
| 5.8  | Enumeração dos cantos do plano de colagem. . . . .   | 62 |
| 5.9  | Orientações do plano de colagem para medição dos cantos . . . . .  | 63 |



# Lista de Tabelas

|     |   |    |
|-----|---|----|
| 2.1 | Taxonomia dos sistemas de apoio ao operador estudados . . . . .                                 | 8  |
| 2.2 | Avaliação dos sistemas de apoio, com base nos parâmetros de [1] . . . . .                       | 11 |
| 3.1 | Características da câmara Microsoft Kinect V1 . . . . .   | 31 |
| 3.2 | Características da câmara Intel Realsense D435 . . . . .  | 31 |
| 3.3 | Tabela de decisão do atuador. . . . .   | 32 |
| 5.1 | Especificações de diferentes câmaras de profundidade . . . . .                                  | 53 |
| 5.2 | Configuração dos algoritmos de registo e segmentação utilizados . . . . .                       | 54 |
| 5.3 | Erro na localização do ponto de colagem da última peça . . . . .                                | 57 |
| 5.4 | Erro após convergência do ponto de projeção em cada ponto de colagem . . . . .                  | 58 |
| 5.5 | Comparação dos dois sensores na deteção dos cantos do plano de colagem, a 0 graus               | 64 |
| 5.6 | Comparação dos dois sensores na deteção dos cantos do plano de colagem, a 15<br>graus . . . . . | 64 |
| 5.7 | Comparação dos dois sensores na deteção dos cantos do plano de colagem, a 30<br>graus . . . . . | 64 |
| 5.8 | Comparação dos dois sensores na deteção dos cantos do plano de colagem, a 40<br>graus . . . . . | 65 |



# Abreviaturas e Símbolos

|        |                                  |
|--------|----------------------------------|
| VTE    | Virtual Training Environment     |
| TDS    | Training Distribution System     |
| TES    | Training in Emergency Situations |
| ITS    | Immersive Training Simulator     |
| ID     | Identifier                       |
| AR     | Augmented Reality                |
| VR     | Virtual Reality                  |
| LED    | Light-emitting diode             |
| 2D     | Two dimensional                  |
| 3D     | Three dimensional                |
| BCH    | Bose–Chaudhuri–Hocquenghem       |
| RGB    | Red, Green, Blue                 |
| RGB-D  | Red, Green, Blue, Depth          |
| SDK    | Software Development Kit         |
| IR     | Infra-red                        |
| AHP    | Análise Hierárquica do processo  |
| PCL    | Point Cloud Library              |
| ROS    | Robotic Operating System         |
| UML    | Unified Modeling Language        |
| ICP    | Iterative Closest Point          |
| RANSAC | RANdom SAmple Consensus          |
| min    | mínimo                           |
| max    | máximo                           |
| m      | metros                           |
| mm     | milímetros                       |
| min    | mínimo                           |
| max    | máximo                           |
| MLS    | Moving Least Squares             |



# Capítulo 1

## Introdução

### 1.1 Contexto

O processo da produção de objetos em aço pelo método de fundição por cera perdida começa pela produção em série de moldes por fundição injetada. Depois, estes moldes são preenchidos por cera com propriedades especiais, resultando em moldes em cera (figura 1.1a) que são consecutivamente transportados até um operador, cuja tarefa é de colar manualmente o número previamente estipulado de peças num tronco, também ele feito de cera. Esta colagem tem de respeitar um conjunto de requisitos, como a correta orientação e posição de cada peça, a distância entre peças, entre outros (figura 1.1b), sendo que estes dependem do produto final a ser produzido. Quando este operador termina a sua tarefa, um conjunto de árvores (que corresponde ao tronco, já com todas as peças coladas) é agrupado e é submetido a um líquido cerâmico, seguido de diferentes tipos de areia. Posteriormente, a cera é derretida num forno, e os moldes cerâmicos são cozidos, enquanto o aço é também ele derretido. Por fim, o aço é derramado dentro da cerâmica (figura 1.1c), e após o seu arrefecimento, esta é partida, resultando assim nas peças pretendidas, feitas em aço (figura 1.1d). Todo este processo tem a duração estimada de duas a três semanas.

As imagens abaixo foram extraídas do episódio 3 da temporada 14 da série "How It's Made".



Figura 1.1: Algumas tarefas da produção de peças em aço pelo processo de fundição por cera perdida.

Esta dissertação foi conduzida em ambiente industrial, e tem como objetivo o estudo de um sistema, e respetiva implementação, que ajude o operador na tarefa de colar os moldes de cera no tronco.

## 1.2 Motivação

A produção de objetos em aço é um processo altamente exigente no que diz respeito aos requisitos referentes à estrutura física do produto final. Como mencionado acima, na colagem dos moldes de cera no tronco, os primeiros têm de ser colados em posições e com orientação predefinidas, e com distância entre peças específica. Estes parâmetros têm de ser rigorosamente cumpridos, porque caso contrário a peça final é considerada como defeituosa e é rejeitada. Além disso, só é possível detetar estes defeitos no fim de todo o processo, e visto que a tarefa de colagem é das primeiras a ser realizada, estas restrições podem causar elevado tempo não produtivo.

Zollern e Comandita Portugal [2] é uma empresa produtora de peças em aço baseadas nos requisitos técnicos e estruturais fornecidos pelo cliente, abrangendo um grande leque de ligas de aço, e o peso destes produtos pode variar de 1-2 gramas, até 15 quilogramas. Nesta empresa, a tarefa de colagem é feita manualmente por operadores, sem lhes ser fornecido qualquer suporte ou guia, pelo que as suas decisões e respetiva performance baseiam-se unicamente na sua experiência e nas suas medições a olho nu. Isto requer um período de aprendizagem, por parte de cada operador, de aproximadamente um ano, e a uma baixa produtividade do processo, causada principalmente por erros cometidos pelo operador nesta tarefa. Atualmente, os operadores desta empresa medem a distância entre peças através da grossura de um dos dedos que utilizam para pegar na peça que vão colar. Como a distância entre peças é uma constante que deve ser respeitada durante a colagem, a técnica que estes usam apresenta sempre algum erro em relação à distância estipulada, erro esse que se acumula ao longo da colagem das várias peças. Acrescenta-se ainda que operadores

diferentes têm dedos com grossura diferente, pelo que mesmo este erro varia de operador para operador.

### 1.3 Objetivos

Esta dissertação tem como objetivo o estudo de um sistema que guie os operadores no decorrer da tarefa de colar moldes de cera no tronco. Tendo em conta as características da tarefa em questão, e os principais erros cometidos atualmente pelos operadores:

- A colagem deve ser feita com a maior precisão possível;
- Os operadores responsáveis pela colagem trabalham nesta tarefa durante várias horas seguidas;
- A técnica que os operadores utilizam para medir a distância entre peças resulta em distâncias diferentes para operadores diferentes, pois dependem da grossura do dedo do operador;
- O plano onde a colagem é efetuada tem área reduzida.

O sistema deve cumprir os seguintes requisitos:

- A assistência deve ser prestada diretamente no plano do tronco onde a colagem está a ser realizada. Assim, o operador não é obrigado a alternar o olhar entre o plano e um ecrã;
- Ainda assim, não deve ser introduzida demasiada informação no plano, devido à sua área reduzida, para não confundir o operador;
- Como o operador trabalha nesta tarefa durante várias horas seguidas, este não deve ter de usar qualquer dispositivo no seu corpo, para não causar cansaço adicional;
- O Hardware escolhido não deve limitar os movimentos do operador, nem deve deixar este desconfortável;
- A implementação deve minimizar a acumulação de erro entre o ponto onde um molde deve ser colado, e o ponto onde o operador efetivamente cola cada molde, ao longo da execução da colagem dos vários moldes. Além disso, a distância entre peças não deve variar mediante as características físicas do operador que estiver a efetuar a colagem.

### 1.4 Estrutura do documento

O restante documento está organizado da seguinte forma: no capítulo 2 é analisada a literatura estudada no âmbito de sistemas de apoio ao operador, juntamente com técnicas de calibração de câmara e projetor. Ainda neste capítulo, são também mencionados exemplos de empresas que já desenvolveram sistemas de apoio ao operador, quer para a fase de aprendizagem quer na assistência durante a execução de uma tarefa na indústria da manufatura, e as respetivas tecnologias

são descritas. No capítulo 3, abordam-se as diferentes propostas de Hardware consideradas para serem utilizadas no teste da implementação, e é feito um resumo dos Softwares utilizados durante a fase de desenvolvimento. No capítulo 4, é apresentada a implementação da solução ao problema. No capítulo 5, são comentados os resultados obtidos com a implementação feita. Por fim, no capítulo 6 são tiradas conclusões relativamente ao estudo realizado, e são propostas melhorias como trabalho futuro.

# Capítulo 2

## Estado da Arte

### 2.1 Revisão da literatura

#### 2.1.1 Sistemas de Apoio ao Operador

Vários investigadores têm-se focado no desenvolvimento de novos sistemas que melhorem significativamente a produtividade das empresas. Tais sistemas podem ter o propósito de ser integrados quer nas camadas inferiores, quer nas superiores da organização de uma empresa. Isto é, desde o planeamento da sequência de ações que um operador deve realizar para cumprir uma tarefa, até o planeamento e escalonamento de processos.

A investigação levada a cabo por estes na criação de ambientes virtuais tem também experienciado um crescimento rápido, juntamente com a sua integração e cooperação com humanos, de forma a que estes interajam com o mundo real de forma diferente. Na indústria da manufatura, esta interação pode ser útil ao assistir os operadores em ambos períodos de aprendizagem e de execução das tarefas.

Apesar das tecnologias de aprendizagem ou assistência já desenvolvidas poderem ser agrupadas com base em vários parâmetros, uma categorização interessante foi feita em [1], na qual os autores consideram três parâmetros principais: incorporação da tecnologia no utilizador, grau de presença no mundo real, e interatividade entre o utilizador e a tecnologia.

- Incorporação da tecnologia

A tecnologia tem vindo a evoluir no sentido de integrar o dia a dia das pessoas. Dispositivos mais recentes são cada vez mais pequenos, mais conectados às ações dos humanos, e o seu impacto nas atividades diárias dos utilizadores tem-se tornado cada vez mais notável. Além disso, algumas são mesmo ligadas ao corpo do utilizador, intensificando as funções sensoriais deste.

Na indústria da manufatura, esta incorporação pode ajudar o operador a executar uma tarefa ao estender a sua perceção do que o rodeia, e melhorando a sua capacidade motora.

Os autores dividiram as tecnologias em duas categorias, com base no seu nível de incorporação em relação ao utilizador (fig. 2.1):

- Dispositivos externos: Podem ser dispositivos externos estacionários, como um computador ou um monitor, ou dispositivos externos portáteis, como um telemóvel ou um *tablet*.
- Dispositivos internos: Podem ser dispositivos usados pelo operador, como *head-mounted displays* ou aparelhos implantados.



Figura 2.1: Distinção entre dispositivos, no que diz respeito à incorporação da tecnologia no usuário.

Extraído de [1].

- Presença

Neste contexto, "presença" é definida como sendo a percepção que o humano tem de estar no espaço físico no qual ele realmente está, ou de estar no espaço em que a tecnologia o insere. Esta sensação de estar num outro espaço que não o real é induzida pela tecnologia, sendo este um ambiente virtual. Dentro dele, ao efetuar uma ação específica, o programa tem a capacidade de responder, alterando o cenário ou retornando uma indicação sensorial.

Na indústria da manufatura, o grau desejável de presença de um sistema de apoio ao operador é dependente das características das tarefas que este tem de executar. Enquanto que a sensação de estar num ambiente virtual é mais útil em grande parte dos sistemas de aprendizagem, nas quais o utilizador pode simular as ações necessárias e receber instruções visuais dentro deste ambiente, em tarefas que requeiram movimentos com maior precisão (como a montagem de peças cujo produto final tenha requisitos físicos ou estruturais muito rígidos), os operadores necessitam de praticar num cenário o mais realista possível, pelo que um ambiente virtual não é o mais indicado.

Como mencionado na categoria anterior, a capacidade de uma tecnologia em fornecer ao seu utilizador a sensação de presença está dependente do respetivo grau de incorporação: um dispositivo externo não consegue criar a sensação de estar num ambiente virtual (fig. 2.2).

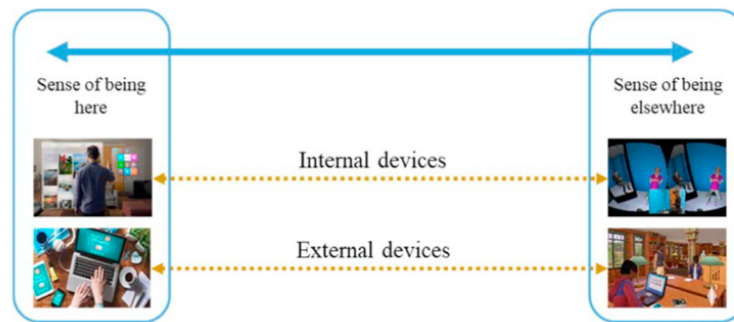


Figura 2.2: Diferença entre tecnologias que dão ao utilizador a sensação de estar onde ele de facto está e tecnologias que o inserem num ambiente virtual.

Extraído de [1].

- Interatividade

Os autores definem interatividade como "um processo dinâmico baseado na interação entre dois agentes: utilizadores e tecnologia"[1]. Por outras palavras, "interatividade é definida como a capacidade do seu utilizador de modificar e receber *feedback* das suas ações na realidade onde a experiência está a ocorrer"[1]. Posto isto, sistemas cuja tecnologia necessite de pouca interatividade com o utilizador são usualmente utilizadas para controlo de navegação, e sistemas cuja tecnologia requira elevada interatividade com o utilizador são usadas para fornecer a este a capacidade de efetuar ações dentro do ambiente.

Na indústria da manufatura, o nível ótimo de interatividade entre o sistema de apoio ao operador e o próprio operador é tipicamente elevado, visto que a forma mais eficaz de assistência é a de acompanhar este durante a execução da sequência de movimentos. Para isto, o sistema tem de identificar quando um movimento é finalizado e atualizar a instrução dada para ajudar o operador no passo seguinte (fig. 2.3).

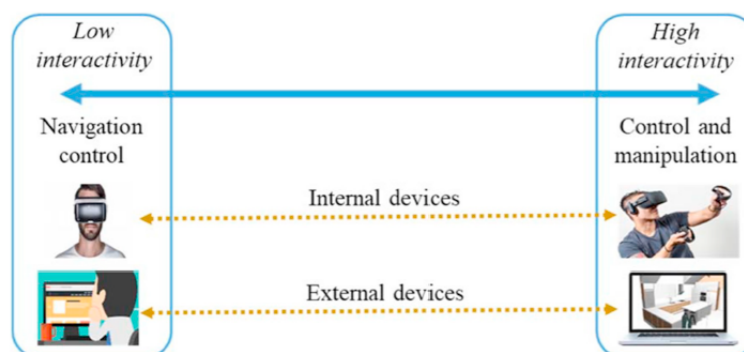


Figura 2.3: Exemplos de tecnologias com baixo grau de interatividade com o utilizador e de tecnologias com elevado grau de interatividade com o mesmo.

Extraído de [1].

Com base nestes parâmetros, três principais categorias foram consideradas para agrupar os diferentes tipos de sistemas estudados. A tabela 2.1 apresenta esta taxonomia, juntamente com uma breve descrição de cada grupo.

Tabela 2.1: Taxonomia dos sistemas de apoio ao operador estudados

| <b>Tecnologia</b>   | <b>Descrição</b>   | <b>Interação Humano-Tecnologia</b> |
|---------------------|--|------------------------------------|
| Realidade Virtual   | Tecnologia que substitui o ambiente real por um virtual.                         | Ocorre no ambiente virtual.        |
| Realidade Aumentada | Tecnologia que reproduz elementos virtuais no ambiente real, através de um ecrã. | Ocorre através de um ecrã.         |
| Realidade Mista     | Tecnologia que reproduz elementos virtuais diretamente no ambiente real.         | Ocorre no ambiente real.           |

### 2.1.1.1 Sistemas baseados em Realidade Virtual

- Ambiente virtual exibido num computador

Em [3], os autores exploram os problemas no desenho de um sistema de treino que apoia os operadores no seu período de aprendizagem, juntamente com uma apropriada gestão dos dados relevantes ao treino virtual.

Para o sistema de treino, esta solução envolve um computador, no qual o ambiente virtual é processado, um controlador Nintendo e uma câmara Microsoft Kinect, de forma a permitir aos formandos a simulação da execução de tarefas e a visualização da simulação dos seus movimentos através de um monitor. O sistema consiste numa abordagem similar a um jogo, permitindo ao operador seleccionar entre vários níveis de dificuldade de treino, começando no nível "fácil", no qual todas as ferramentas e instruções são disponibilizadas. Mediante o progresso do operador, este vai avançando para níveis mais difíceis, nos quais o apoio é gradualmente limitado. O processo de treino é constantemente monitorizado e o utilizador só pode subir de nível após manter uma performance estável e positiva no nível actual. Apenas quando concluir todos os níveis o operador pode começar a treinar na fábrica real.

- Inclusão do operador num ambiente virtual

Juan Jesús Roldán estudou uma abordagem para um sistema de aprendizagem baseado em Realidade Virtual, a ser aplicado na indústria da manufatura, que combina Realidade Virtual com *Process Mining*, desenvolvendo também um protótipo deste sistema [4]. Usando um *head-mounted display*, o operador é inserido num ambiente virtual, através do qual um outro operário, mais experiente, interage com o formando dando-lhe indicações e ajudando-o na tarefa. Neste ambiente, o utilizador pode seleccionar entre dois modos:

- Modo experiente - Neste modo um operador mais experiente realiza a montagem, de forma a que o sistema guarde a informação relevante de cada passo envolvido,

movimentos e decisões que caracterizam cada processo de montagem. É recomendado que cada montagem seja efetuada por mais do que um operador experiente, para que o sistema guarde as diferentes combinações de movimentos possíveis que atinjam o mesmo resultado final, aumentando assim a sua robustez.

- Modo de treino - Neste modo é usado o mesmo cenário do que no modo experiente, e o operador pode praticar qualquer montagem já inserida no sistema. É dado ao utilizador um estado inicial, uma descrição da tarefa a realizar e um estado final.

Ao registar e interpretar a informação recolhida pelo sistema, através de *event logs*, este reconhece modelos e padrões para cada montagem. Tais modelos são depois decompostos em pequenas tarefas, em *task logs*, sendo estas usadas posteriormente no modo de treino. *Process Mining* é usado quando uma grande quantidade de *data logs* é registada, de forma a filtrar eventuais erros cometidos no modo experiente, para que os modelos criados sejam fiáveis.

### 2.1.1.2 Sistemas baseados em Realidade Aumentada

- Assistência por intermédio de um ecrã

Em [5], os autores propõem um sistema para assistência ao operador em tarefas de montagem, nas quais as instruções são fornecidas através de um monitor *touch-screen*. Estas instruções, dadas passo a passo, são transmitidas por *stream* vídeo, que capta toda a estação de trabalho, sobrepondo-as à captação nos locais pretendidos. Quando um operador acaba um passo, a instrução é atualizada quer tocando no ecrã, quer por Software que deteta os movimentos do operador. O sistema identifica também erros cometidos pelo utilizador, e temporiza o tempo que este demora a executar a montagem, calculando assim a performance e eficiência quer da tecnologia, quer do processo.

- Assistência fornecida por *HoloLens*

Uma abordagem diferente é explorada por Tariq Masood em [6], na qual a assistência para tarefas de montagem é dada através de um *head-mounted display*, Microsoft HoloLens. Além disso, o Software Unity foi também usado no desenvolvimento do ambiente 3D. Com estas duas tecnologias, o operário pode visualizar a representação 3D dos passos da montagem a ser conduzida através de um holograma. Uma interface virtual é também disponibilizada, de forma a permitir ao operador reiniciar o processo ou avançar para o passo seguinte. Os componentes necessários à ação que o utilizador estiver a efetuar naquele instante são também destacados, juntamente com a indicação de onde os colocar. Com este sistema, o operador não precisa de desviar o olhar da estação de trabalho, visto que toda a assistência é fornecida na mesma.

### 2.1.1.3 Sistemas baseados em Realidade Mista

- Assistência fornecida diretamente no local de trabalho

Em [7], Leonardo Rodriguez propõe um método para assistência virtual ao operador em tarefas de montagem manual, usando técnicas de *Projection Mapping*, isto é, projetando instruções diretamente na mesa de trabalho, fornecendo-lhe assim assistência visual. *Projection Mapping* será explicada mais pormenorizadamente na secção seguinte.

Para que o sistema funcione como planeado, as pessoas responsáveis pelo desenvolvimento deste têm de ter conhecimento promenorizado de todos os passos envolvidos no processo de montagem, e respetiva sequência, para que as instruções possam ser corretamente apresentadas, com a ordem correta, e atualizadas sempre que uma fique concluída. Além disso, "a modelação e interpretação de todas as listas de objetos e ferramentas envolvidas [...] bem como a estação de trabalho (modelo de montagem) como um mundo virtual 3D foi feito com base num conjunto de processamentos de imagem dos modelos matemáticos que descrevem o mundo real"[7].

A estação de trabalho foi então equipada com uma linha de sensores de movimento (neste caso, uma câmara Microsoft Kinect), que funciona como um sistema de reconhecimento de movimento, e um ecrã que reproduz as instruções para o operador realizar. Depois, um projetor foi adicionado, numa posição fixa, para que o operador seja assistido através de projeção, diretamente no local de montagem, de forma a que o operador não tenha de desviar o olhar deste para um ecrã. A posição do projetor foi escolhida com base na perspectiva do operador e com a garantia de que toda a estação de trabalho é cobrida pela projeção.

O sistema foi concebido para assistir na montagem manual de suporte de cartões de negócio num *poke-yoke* e funciona da seguinte forma: quando o processo de montagem começa, o projetor projeta na mesa de montagem o sítio exato onde a primeira peça deve ser colocada. A câmara está constantemente a captar os movimentos do operador, e quando este pousa a peça no lugar indicado, o sistema identifica esta ação e a projeção é atualizada, projetando o sítio onde a peça seguinte deve ser colocada, e assim em diante.

- Assistência fornecida diretamente no local de trabalho - outra abordagem

Em 2019, um sistema similar ao anteriormente descrito foi implementado por Hugo Álvarez para uma empresa de empacotamento, para assistir o operador numa execução mais precisa da tarefa de produzir *templates* de correção, para serem posteriormente usados na máquina de corte [8]. Um *template* de correção é um papel onde o operador, após medir o desalinhamento da máquina de corte, marca com fitas as posições exatas onde a máquina tem de intensificar o contacto com o cartão da caixa que pretende construir.

Antes do sistema ser implementado, sempre que uma nova caixa iria ser produzida, o operador tinha de se deslocar da área de trabalho para o armazém para arquivar o *template* previamente utilizado, pegar no *template* associado à nova caixa, e deslocar-se de volta para a área de trabalho para executar o corte. Além disso, cada máquina tem o seu desalinhamento, pelo que mesmo se forem cortadas duas caixas iguais, caso as máquinas não sejam

as mesmas, o *template* de correção também não será o mesmo. Estes foram também desenhados pelo operador, sem qualquer suporte, apenas pela análise visual baseada na sua experiência, sendo por isso muito suscetível a erros.

O sistema consiste numa câmara que digitaliza o *template*, extraindo a informação das fitas (posição e orientação), e guardando a mesma num servidor. Depois, o operador coloca a placa de cartão na máquina de corte, e o projetor projeta o *template* na placa. Quer o projetor quer a câmara devem ser posicionadas garantindo que toda a área é coberta. Visto que um produto necessita de um diferente *template* para cada máquina, uma digitalização tem de ser feita para cada produto e para cada máquina. Após todas as digitalizações terem sido feitas, o operador tem apenas de especificar qual a máquina a ser utilizada e a caixa a ser produzida. Para calibrar o centro da câmara, foi usada uma folha de papel branca, como *template* de correção padrão, com quadrados pretos nos cantos.

Com este sistema, o arquivamento e organização dos *templates* de correção foram automatizados, e o operador deixou de ter de se deslocar da sua área de trabalho cada vez que o produto final é alterado.

#### 2.1.1.4 Comparação das tecnologias

Com base nos parâmetros mencionados em 2.1, extraídos de [1], os sistemas mencionados foram incluídos nos seguintes grupos (tabela 2.2):

Tabela 2.2: Avaliação dos sistemas de apoio, com base nos parâmetros de [1]

| <b>Tecnologia</b>  | <b>Incorporação</b> | <b>Presença</b>  | <b>Interatividade</b> |
|--|---------------------|------------------|-----------------------|
| Ambiente virtual exibido num computador(2.1.1.1)                 | Externo             | Ambiente real    | Elevada               |
| Inclusão do operador num ambiente virtual (2.1.1.1)              | Interno             | Ambiente virtual | Elevada               |
| Assistência por intermédio de um ecrã (2.1.1.2)                  | Externo             | Ambiente real    | Elevada               |
| Assistência fornecida por <i>HoloLens</i> (2.1.1.2)              | Interno             | Ambiente real    | Elevada               |
| Assistência fornecida diretamente no local de trabalho (2.1.1.3) | Externo             | Ambiente real    | Elevada               |
| Assistência fornecida diretamente no local de trabalho (2.1.1.3) | Externo             | Ambiente real    | Elevada               |

Adicionalmente, com base nas definições de Realidade Virtual, Aumentada e Mista, e tendo em conta a literatura estudada, pode-se concluir que:

- **Sistemas baseados em Realidade Virtual** são os mais indicados para processos e tarefas mais complexas, e para atividades onde um erro cometido pelo operador pode causar consequências dispendiosas ou até perigosas. Além disso, é também ótima a possibilidade de simular processos que requeiram elevada mobilidade dos operadores dentro da fábrica, fornecendo instruções ao utilizador sobre como se movimentar (lugares a percorrer), e sobre como executar as tarefas. No entanto, este tipo de sistemas não é indicado para montagens que não requerem que o operador saia da sua estação de trabalho, e que por outro lado exigem elevada precisão por parte deste, para garantir a qualidade do produto final. Assistem, portanto, no que fazer e como fazer, mas não fornecem a precisão motora que alguns processos exigem. Posto isto, são usados tipicamente como sistemas de aprendizagem, em vez de como sistemas de assistência em tempo real.
- **Sistemas baseados em Realidade Aumentada** são os mais indicados para apoio ao operador em tempo real, durante a execução de uma tarefa e quando a mesma requer que o operador se movimente por diferentes sítios, ou quando não é desejável que informação adicional seja inserida na mesa de trabalho. Visto que estes sistemas não alteram o ambiente, a assistência é fornecida com o máximo grau de realismo possível. No entanto, não são os indicados para operações complexas: o ecrã tem limitações em termos de dimensões, e caso seja apresentada demasiada informação nele, o operador necessitará de dedicar algum tempo a perceber o que fazer. Pode ser usada quer para fins de aprendizagem, quer para assistência.
- **Sistemas baseados em Realidade Mista** são os mais indicados para assistência em tempo real, enquanto o operador efetua uma tarefa e o produto final possui requisitos muito exigentes. Ao receber as instruções diretamente na mesa de trabalho, o operário pode focar-se apenas na execução da tarefa, não tendo de alternar o olhar entre a mesa e um ecrã. Tal propriedade maximiza a precisão dos movimentos. Adicionalmente, não tendo de usar nenhum *head-mounted display* ou de carregar um monitor, o operário sente-se mais confortável, especialmente caso a tarefa seja bastante repetitiva e este tenha de a executar durante horas. No entanto, apesar da área onde as instruções são descritas ser maior, a limitação indicada nos sistemas baseados em Realidade Aumentada também está presente neste tipo de sistemas.

Como mencionado no capítulo 1, a tarefa de colagem de moldes de cera num tronco é muito repetitiva e exigente, visto que um pequeno erro na colagem de uma peça pode levar à rejeição da peça final. Além disso, o operador tem de medir manualmente a distância entre peças e o respetivo ângulo, sendo esta medição muito suscetível à ocorrência de erros. Com base nestas características, o tipo de sistemas mais benéfico para esta aplicação é a baseada em Realidade Mista.

### 2.1.2 Projection Mapping

*Projection Mapping* é uma técnica de projeção de uma imagem estacionária no tempo, ou de um vídeo, numa superfície 3D, independentemente da forma desta (plana ou irregular). Esta técnica requer soluções de Hardware e Software apropriados, dependendo da aplicação onde tal sistema será aplicado [9][10].



Figura 2.4: Projeção clássica num ecrã plano vs Projection Mapping.  
Extraído de [9].

Como alternativa ao uso de um ecrã para o fornecimento de instruções, o uso de *Projection Mapping* permite ao operador ser assistido diretamente na mesa de trabalho, reduzindo o seu esforço mental e aumentando a precisão da execução da tarefa.

A correta calibração quer da câmara quer do projetor é de grande importância para que o sistema cumpra o seu correto funcionamento. Em sistemas de apoio ao operador, uma má calibração da câmara ou do projetor pode induzir erro na projeção do laser, em relação á área pretendida, que pode ser o suficiente para a rejeição do produto final.

#### 2.1.2.1 Modelos da câmara e do projetor

- Modelo linear

Antes do estudo das técnicas de calibração de uma câmara e de um projetor, os respetivos modelos matemáticos devem ser desenhados.

A câmara é tipicamente descrita através do seu modelo *pinhole*. De acordo com o artigo escrito por Anwar [11], com este modelo é possível mapear as coordenadas de um ponto no espaço, em três dimensões e no sistema de coordenadas do mundo,  $P^w = [X^w Y^w Z^w 1]^T$ , no plano 2D da imagem da câmara,  $P^c = [u^c v^c 1]^T$ :

$$P^c = sM^c[R^c t^c]P^w \quad M^c = \begin{bmatrix} f_x^c & 0 & c_x^c \\ 0 & f_y^c & c_y^c \\ 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

Onde  $s$  corresponde ao fator de escala da projeção de  $P^w$  em  $P^c$ . Os parâmetros intrínsecos da câmara são representados por  $M^c$ : as coordenadas da focagem principal são representadas

por  $c_x^c$  e  $c_y^c$ , e os comprimentos focais nos eixos  $x$  e  $y$  são representados por  $f_x^c$  e  $f_y^c$ . Por outro lado, os parâmetros extrínsecos da câmara, isto é, a matriz de transformação (matriz de rotação e translação) entre o sistema de coordenadas da câmara e do mundo é representada por  $[R^c t^c]$ .

Em [11], H. Anwar considera o projetor como sendo equivalente a uma câmara invertida, o que significa que este projeta uma imagem em vez de a capturar. Esta propriedade permite que o primeiro seja representado pelas mesmas equações[11].

$$P^p = sM^p[R^p t^p]P^w \quad M^p = \begin{bmatrix} f_x^p & 0 & c_x^p \\ 0 & f_y^p & c_y^p \\ 0 & 0 & 1 \end{bmatrix} \quad (2.2)$$

Aqui,  $P^p = [u^p v^p 1]^T$  é um ponto 2D no plano da imagem do projetor e a equação apresentada à esquerda define a sua relação com o respetivo ponto com informação 3D do espaço do mundo,  $P^w = sM^p[R^p t^p]P^w$ . Além disso,  $M^p$  e  $[R^p t^p]$  denotam os parâmetros intrínsecos e extrínsecos do projetor, respetivamente.

Das equações acima, constata-se que  $sM^\alpha[R^\alpha t^\alpha]^T$  define uma matriz de transformação homogénea e linear do sistema de coordenadas do mundo para o plano de  $\alpha$ . Por outras palavras, é uma matriz de homografia.

$$H_w^c = sM^c[R^p t^p] = sM^c \begin{bmatrix} r_{11}^c & r_{12}^c & t_x^c \\ r_{21}^c & r_{22}^c & t_y^c \\ r_{31}^c & r_{32}^c & t_z^c \end{bmatrix} \quad (2.3)$$

Pelo que as equações 2.1 e 2.2 podem ser escritas como  $P^c = H_w^c P^w$  e  $P^p = H_w^p P^w$ . Combinando estas duas equações:

$$P^p = H_c^p P^c \quad (2.4)$$

A figura 2.5 mostra a relação entre os sistemas de coordenadas da câmara, do projetor, e do ecrã.

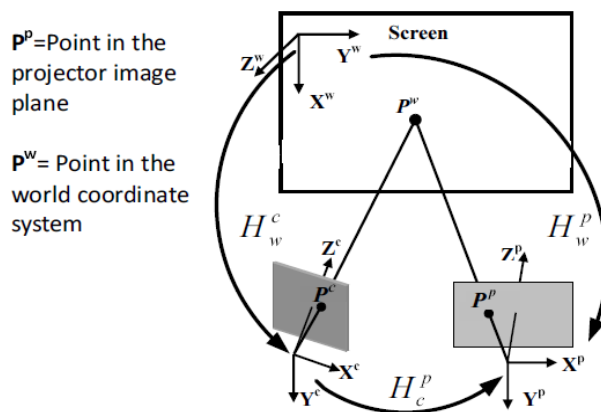


Figura 2.5: Modelo matemático da câmara e do projetor.  
Extraído de [11].

As equações 2.1, 2.2, 2.3 e 2.4 foram extraídas de [11].

- Considerações não lineares

Daniel Moreno, em [12], considera a não linearidade induzida pelas distorções tangencial e radial de ambos câmara e projetor. Para aplicações que requiram elevada precisão da projeção, estes parâmetros podem causar um erro significativo e o sistema deixará então de ser útil. Posto isto, modelos mais completos foram desenvolvidos, incluindo estas distorções.

Neste modelo, a relação entre um ponto no sistema de coordenadas do mundo  $X$ , com origem no centro da câmara, e as coordenadas de um píxel da sua imagem, no plano da câmara  $u$  é:

$$X = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad \tilde{u} = \begin{bmatrix} \tilde{u}_x \\ \tilde{u}_y \end{bmatrix} = \begin{bmatrix} x/z \\ y/z \end{bmatrix} \quad (2.5)$$

$$u = K_c L(\tilde{u}) \quad (2.6)$$

Onde  $K_c$  representa os parâmetros de calibração intrínseca:

$$K^c = \begin{bmatrix} f_x^p & \gamma & o_x^p \\ 0 & f_y^p & o_y^p \\ 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

Esta matriz é similar a  $M^c$ , mas contendo o parâmetro que descreve a assimetria dos dois planos,  $\gamma$ .  $L(\tilde{u})$  é definida como:

$$L(\tilde{u}) = \begin{bmatrix} \tilde{u}(1 + k_1 r^2 + k_2 r^4) + \Delta_t(\tilde{u}) \\ 1 \end{bmatrix} \quad (2.8)$$

$$\Delta_t(\tilde{u}) = \begin{bmatrix} 2k_3 \tilde{u}_x \tilde{u}_y + k_4 (r^2 + 2\tilde{u}_x^2) \\ k_3 (r^2 + 2\tilde{u}_y^2) + 2k_4 \tilde{u}_x \tilde{u}_y \end{bmatrix} \quad (2.9)$$

$$r^2 = \tilde{u}_x^2 + \tilde{u}_y^2 \quad (2.10)$$

Aqui,  $k_1$  and  $k_2$  são os coeficientes de distorção radial, e  $k_3$  e  $k_4$  representam os coeficientes de distorção tangencial.

O projetor é novamente considerado como uma câmara invertida, pelo que as equações acima podem também ser aplicadas na representação deste.

As equações 2.5, 2.6, 2.7, 2.8, 2.9 e 2.10 foram extraídas de [12].

### 2.1.2.2 Calibração da câmara

Para a calibração da câmara, Z. Zhang propõe um método baseado na identificação das coordenadas de todos os cantos de um padrão de xadrez, no sistema de coordenadas do plano da câmara [13]. Este método requer que o padrão seja impresso e colado numa superfície plana, e posteriormente sejam tiradas fotografias a este, colocando a câmara em diferentes orientações.

Com base na abordagem de Zhang, Daniel Moreno, em [12], usa uma imagem totalmente iluminada de um padrão de xadrez, em cada orientação da câmara, para a calibração do sensor. Utilizando a função da biblioteca *OpenCV* `findChessboardCorners()`, é esperado encontrar-se a localização de todos os cantos, para posterior ajuste de forma a atingir precisão ao nível do sub-píxel. No fim, os parâmetros da câmara, já calibrada, são calculados com a função `calibrateCamera()`.

### 2.1.2.3 Calibração do projetor

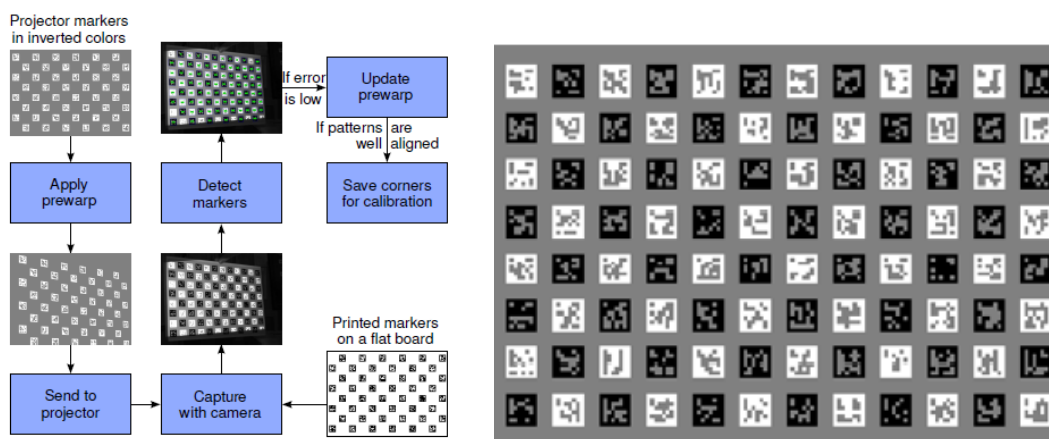
- Calibração por ajuste iterativo da projeção
  - Em [14], os autores exploram um método de calibração de um projetor, usando para isso uma câmara não calibrada e a ferramenta *Bouguet's Matlab Toolbox*. Neste, é utilizado o mesmo padrão de xadrez para a calibração da câmara e do projetor. O padrão é projetado de forma a sobrepor um padrão igual impresso, com cor contrastante. Desta forma, não é necessário fazer-se a correspondência entre a grelha de pontos no plano do projetor e a grelha do padrão real. Depois, a câmara não calibrada captura esta sobreposição, no sistema de coordenadas da câmara. Com esta captura, ambos os padrões são comparados e, caso a distância entre os pontos seja

superior a um valor predefinido, a projeção é movida de forma a reduzir esta distância. Este procedimento é executado iterativamente e, depois da distância de todos os pontos se encontrar abaixo do limite mencionado, a *toolbox* de Matlab é diretamente aplicada para calcular os parâmetros do projetor. Este processo é repetido para diferentes posições do projetor, e no fim do processo de otimização incorporado na *toolbox* é executado.

Para este método, primeiro é necessário estimar-se a matriz de homografia que representa a transformação de um ponto do sistema de coordenadas da câmara para o sistema do projetor. Para isto, quatro pontos com cor contrastante são projetados, formando uma figura retangular no plano do projetor e a homografia é calculada aplicando a técnica de Transformação Linear Direta. Os autores afirmam que encontrar esta relação ajuda na minimização quer do erro inicial da estimação dos cantos no plano do projetor, quer do tempo dispendido nas várias iterações. Depois, a câmara captura o padrão de xadrez impresso, deteta os seus cantos, e usa a matriz de homografia para traduzir esta informação em pontos do projetor. Para cada ponto de um canto, o projetor projeta vários pontos, compara as coordenadas de cada um destes com as coordenadas do canto, e seleciona o ponto com menor erro.

Através de uma série de experiências, os autores estudaram o efeito do aumento da distância da câmara na qualidade da calibração, e observaram que a degradação desta é significativa. Constataram também que quanto maior o ângulo entre a câmara e o projetor, maior é o erro de reprojeção. Uma vantagem deste método é que é utilizada uma câmara não calibrada, pelo que os erros da calibração desta não afetam a calibração do projetor.

- Outra técnica foi desenvolvida por S. Audet, baseada na deteção de *BCH fiducial markers* [15]. Estes marcadores são fornecidos pela *toolkit ARToolKitPlus*. Nesta abordagem, metade dos marcadores é impressa, enquanto a outra metade é projetada com cores invertidas, sendo que cada um dos marcadores possui um identificador. A *toolkit* tem um algoritmo que deteta os contornos de cada marcador projetado, formando quadrados à volta destes. No fim, o padrão dentro de cada retângulo é analisado e o respetivo identificador é recuperado. O algoritmo contém também implementado *adaptive thresholding* e *subpixel corner extraction*: enquanto o primeiro fornece uma performance mais robusta, no que diz respeito às variações do nível de brilho ao longo da imagem, o segundo melhora a deteção dos quadrados. Neste método não é exigido que o campo de visão da câmara cubra todo o padrão impresso, visto que qualquer captação contém uma mistura dos dois tipos de marcadores (impressos e projetados). O algoritmo de uma única iteração e o resultado final desejado são apresentados nas figuras 2.6a e 2.6b, respetivamente.



(a) Exemplo de uma iteração do algoritmo de calibração. Extraído de [15].

(b) Resultado final pretendido. Extraído de [15].

Figura 2.6: Iteração do algoritmo de calibração e resultado final pretendido

A calibração é efetuada após um número suficiente de cantos tiver sido acumulado. Usando os cantos do padrão impresso, e dentro destes os detetados, com a matriz de homografia  $H_{printed}^{detected}$ , as distorções lineares são removidas. Com isto, os pontos dos cantos impressos, já sem distorção, e os correspondentes pontos no plano da imagem do projetor podem ser usados na calibração do projetor através da biblioteca *OpenCV*, usando o método de Zhang.

- Calibração com o uso de padrões de luz estruturada

Outro método de calibração de um projetor foi proposto por Daniel Moreno e Gabriel Taubin [12]. Este foi desenvolvido para calibrar um par câmara-projetor para aplicações de luz estruturada, e o projetor é descrito através do modelo *pinhole* estendido da câmara, tendo em conta os coeficientes de distorção radial e tangencial: foram utilizadas as equações presentes em 2.1.2.1. Nesta proposta, o projetor é calibrado independentemente da câmara, pelo que os erros ocorrentes na calibração da última não afetam a calibração do primeiro. Foi também usado um padrão de xadrez como imagem de calibração.

A primeira etapa desta abordagem é de adquirir imagens que contenham informação sobre um padrão de xadrez plano. Sobre este padrão, uma sequência completa de padrões *gray code* é projetado e uma imagem é captada para cada um. Uma nota importante é que uma sequência é composta por duas subsequências: uma com codificação horizontal e outra com codificação vertical. Depois de capturar a sequência completa, a pose do projetor é alterada e todos os passos são repetidos.

Após o atrás exposto, uma sequência *gray code* para uma posição específica do projetor é decodificada. As duas subsequências mencionadas complementam-se mutuamente nesta etapa: a subsequência horizontal permite a associação de cada píxel da imagem da câmara a uma coluna no plano do projetor, enquanto que a subsequência vertical permite a associação de cada píxel a uma linha do plano do projetor. Após esta associação estar concluída, a

homografia local é estimada para cada canto do padrão, na imagem da câmara, e cada canto é convertido das coordenadas da câmara para as do projetor através da respetiva matriz de homografia local.

Os autores afirmam que a independência entre as calibrações da câmara e do projetor resultaram num pequeno erro de reprojeção (o valor registado foi 0.1447 píxeis), quando comparado com métodos que usam uma câmara calibrada para a calibração do projetor (tendo sido o valor utilizado para a comparação 0.8671 píxeis). Isto é justificado pelo facto de que os erros induzidos pela calibração da câmara não afetam a calibração do projetor. Adicionalmente, para avaliar o efeito da distorção das lentes do projetor na projeção, foram medidos os coeficientes  $k_1$ ,  $k_2$ ,  $k_3$  e  $k_4$ . Os resultados obtidos mostraram que o coeficiente radial  $k_2$  apresenta um valor considerável, de 0.3365 píxeis. A distorção máxima foi identificada no canto superior esquerdo da imagem projetada, e esta foi nula no centro. Concluindo, as distorções não lineares não devem ser negligenciadas.

- Calibração do projetor através da matriz de homografia
  - No início do processo de calibração, proposto por Hafeez Anwar em [11], o padrão de xadrez impresso é colado a um ecrã e a câmara, fixada com um ângulo de 20 graus em relação a este, captura a sua imagem. Depois, o padrão impresso é removido, e o projetor é posicionado de forma a apontar perpendicularmente ao plano do ecrã e é capturada uma outra imagem pela câmara. A figura 2.7 mostra a relação entre os pontos do padrão de xadrez projetados e os pontos impressos.

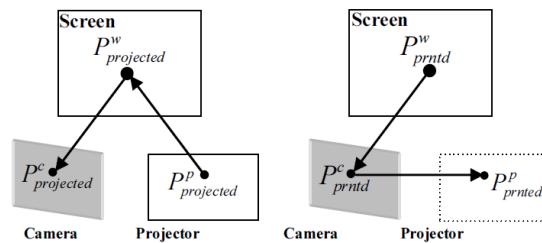


Figura 2.7: Relação entre os pontos do padrão de xadrez projetados e os pontos impressos. Extraído de [11].

Ao aplicar-se a equação 2.4, a figura 2.7 é matematicamente descrita por:

$$P^p_{proj} = H^p_c P^c_{proj} \qquad P^p_{prntd} = H^p_c P^c_{prntd} \qquad (2.11)$$

Depois, usando as funções da biblioteca *OpenCV*, os cantos do padrão de xadrez das imagens da câmara podem ser detetados. Isto é necessário para calcular a matriz de homografia entre a posição frontal e paralela entre o projetor e a câmara,  $H^p_c$ .  $P^p_{prntd}$  representa o padrão de calibração virtual.

Após calculada esta relação, a pose do projetor é alterada. Sendo  $H_w^c = H_c^p H_w^p$ ,

$$H_w^c = H_{p1}^c H_w^{p1} \qquad H_w^c = H_{p2}^c H_w^{p2} \qquad (2.12)$$

Resultando na relação

$$H_{p1}^{p2} = H_c^{p2} H_{p1}^c \qquad (2.13)$$

Na qual  $p1$  e  $p2$  representam a pose inicial do projetor e a nova, respetivamente. É também importante mencionar que a pose da câmara não pode ser alterada durante este procedimento. Adicionalmente, ao analisar a equação 2.13, conclui-se que a matriz de homografia entre a primeira pose do projetor e qualquer outra pose deste pode ser facilmente calculada usando a matriz de homografia do projetor para a câmara, de ambas as poses. O padrão de calibração a ser projetado é também alterado pela relação  $P_{printed}^{pi} = H_{p1}^{pi} P_{printed}^{p1}$ .

O algoritmo de calibração é sumarizado na figura 2.8.

Este método tem as vantagens de usar uma câmara não calibrada para a calibração do projetor, e o facto de que a câmara e o ecrã são ambos estacionários, pelo que a matriz de transformação entre estes não é alterada (os parâmetros extrínsecos só têm de ser calculados uma vez). A maior desvantagem desta abordagem é que, visto que uma matriz de homografia representa a transformação linear entre dois sistemas de coordenadas, as distorções não lineares são negligenciadas, e portanto estes erros não são considerados.

Uma outra abordagem é usada por J. Liao em [16], na qual um padrão de xadrez é impresso na metade inferior de uma placa plana, e uma câmara tira uma fotografia deste, sem conhecimento prévio sobre o padrão usado. Depois, o projetor projeta a imagem fotografada na metade superior da placa e a câmara tira outra fotografia, captando desta vez toda a placa. Estes passos são repetidos para várias posições e orientações da placa, para a calibração da câmara. Por fim, as coordenadas 3D do padrão projetado são calculadas, e com os resultados obtidos os parâmetros intrínsecos e extrínsecos do projetor são extraídos, usando para isso a matriz de homografia  $H_c^p$ .

Este método tem as mesmas desvantagens das presentes no trabalho realizado por Anwar, juntamente com o facto de que é utilizada uma câmara calibrada para a calibração do projetor (os erros poderão acumular).

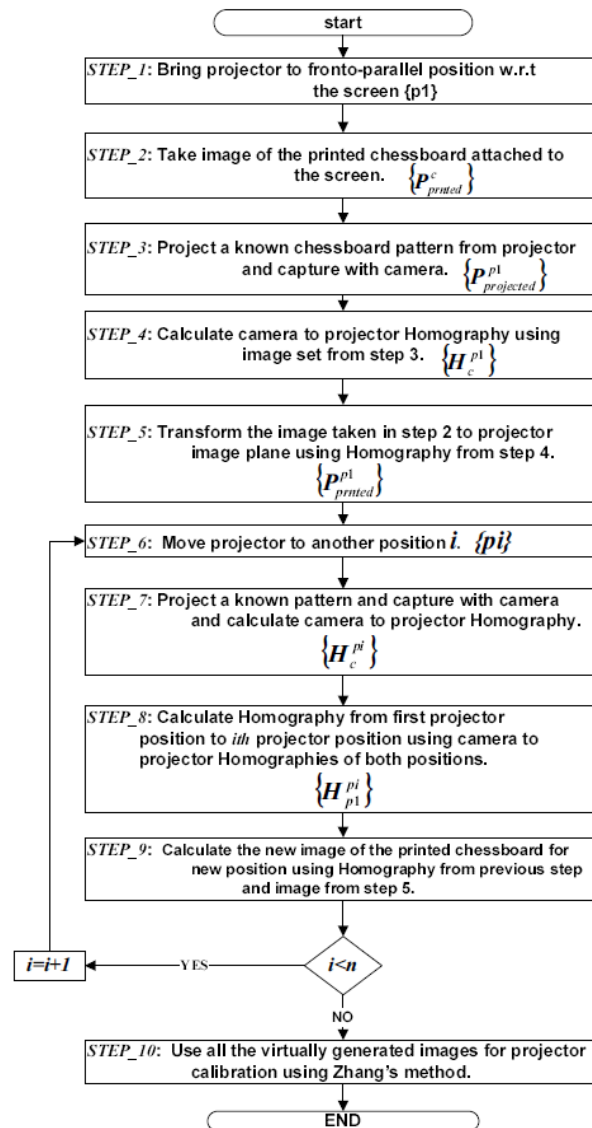


Figura 2.8: Algoritmo para a calibração do projetor.  
Extraído de [11].

## 2.2 Estudo do mercado

A rápida evolução industrial e o aumento da competitividade de mercado presente em grande parte das indústrias tem culminado na corrida das empresas em implementar sistemas que minimizem o tempo gasto em atividades não produtivas. Um dos maiores limitadores da produtividade de uma empresa de manufatura é a performance dos operadores nas tarefas executadas manualmente por estes, pelo que tem sido prestada especial atenção na resolução deste problema. Para isso, várias empresas têm já implementados sistemas tecnológicos focados na ajuda ao operador, quer na fase de aprendizagem, quer na execução da tarefa propriamente dita.

### 2.2.1 Empresas produtoras de sistemas

#### 2.2.1.1 EON Reality

EON Reality [17] é uma empresa que desenvolve sistemas de Software baseados em Realidade Virtual ou Aumentada, sediada na Califórnia, nos Estados Unidos da América. EON Reality cria sistemas para diferentes aplicações:

- Educação
- Energia
- Medicina e Saúde
- Indústria e Manufatura
- Aeroespacia e Transporte
- Segurança e Defesa

Na indústria da manufatura, esta empresa fornece uma solução de um sistema baseado em Realidade Aumentada para assistência nas fases de montagem, denominada *LIAA Project*, que foi desenvolvida numa cooperação com a Opel e a universidade de Patras, para ajudar o operador e a sua interatividade com um braço robótico na montagem de um turbocompressor. *LIAA Project* [18] consiste num *head-mounted display*, responsável por fornecer instruções em tempo real ao trabalhador, juntamente com um braço robótico controlado por um *smartwatch*.

Com este sistema, o robot pode efetuar tarefas que requeiram maior força física, como levantar componentes pesados, enquanto o operador trabalha neles. Outra possível função do robot poderá ser executar tarefas mais repetitivas. A sua utilização resulta na considerável redução no tempo de execução da montagem do turbocompressor e em menores danos físicos nos operadores. A solução é compatível com diferentes modelos de turbocompressores da Opel e a respetiva geometria.

### 2.2.1.2 AVEVA

AVEVA [19] é uma empresa de tecnologias de informação sediada em Cambridge, no Reino Unido. Tem como foco principal o desenvolvimento de sistemas de otimização para serem incorporados em várias áreas como a marinha, química, de produção de ferro, etc. Para tal, AVEVA fornece soluções focadas no planeamento e escalonamento de processos, engenharia integrada, ou na performance das operações.

Para o treino dos operadores, esta empresa produziu dois sistemas de simulação [20]:

- Simuladores para treino dos operadores

Simuladores para treino virtual baseados em computador, que têm como objetivo ajudar o operador a aprender como gerir uma refinaria através de um modelo que controla as respostas da planta às ações do utilizador. Para além do computador, o sistema é também composto por um *head-mounted display*. Adicionalmente, o seu Software inclui sistemas emulados de controlo, para uma representação virtual mais realista da planta da fábrica, juntamente com a interação desta com o formando. Com este ambiente, o operador é capaz de aprender como reagir face a uma situação incomum, ou até perigosa, que ocorra na fábrica real.

- Sistema de treino imersivo, baseado em Realidade Virtual e Aumentada

Ambiente virtual para aprendizagem baseado num *head-mounted display*, usado para acelerar o período de aprendizagem dos operadores. Dentro deste ambiente, o operador pode testar equipamentos e montagens para posterior implementação. Este sistema foi desenvolvido sobre uma arquitetura modular, de forma a ser usado em cooperação com os "Simuladores para treino dos operadores".

Com este sistema, AVEVA atingiu a redução de 20% no tempo de arranque, 30% de poupança em termos económicos e de tempo, e 3% de redução dos gastos em manutenção. Esta informação e a descrição dos sistemas foi extraída de [20].

### 2.2.1.3 Appsbow

Appsbow [21] é uma empresa que produz sistemas baseados em Realidade Virtual para treino, para depois vendê-los a empresas de manufatura. Para treino dos trabalhadores, Appsbow desenvolveu o sistema *VTE (Virtual Training Environment)* [22].

Aquando da implementação de um *VTE*, como mencionado no vídeo disponibilizado no *website* da empresa, os Engenheiros da Appsbow começam por coletar a informação necessária sobre a infraestrutura da empresa cliente, juntamente com os seus processos e todos os materiais envolvidos nestes, e posteriormente constroem um ambiente de treino totalmente interativo com o utilizador, baseado nos processos e nos respetivos procedimentos e cenários físicos envolvidos, com o máximo grau de realismo e detalhe possível. O sistema permite quer o treino de um único operador, quer o treino de vários operadores em simultâneo, sendo que um supervisor pode também participar para controlo e assistência. Além disso, estão presentes vários níveis de dificuldade: "básico", no qual são fornecidas e descritas todas as instruções relativas aos movimentos a

efetuar pelo operador; "assistido", no qual as instruções são fornecidas, mas não como as efetuar; e "exame", no qual o sistema não fornece qualquer ajuda ou instruções.

Para complementar o sistema *VTE*, Appsbow desenvolveu três outras soluções para a camada de gestão, planeamento e controlo de processos:

- *VTE ADM - Training Management*

Acompanha o progresso do formando. Guarda informação, como por exemplo os *log-ins* efetuados, a hora de começo do treino, uma lista com as tarefas executadas incorretamente e a respetiva descrição dos erros cometidos, etc.

- *VTE TDS - Training Distribution System*

Caso um cliente possua mais do que uma empresa, *VTE TDS* permite a redistribuição dos treinos entre as diferentes empresas, simplificando quer o seu controlo, quer a respetiva distribuição.

- *VTE TES - Training in Emergency Situations*

Simulador para situações de emergência, as quais são tipicamente impossíveis de recriar na fábrica real. Com este sistema, o operador pode praticar como reagir face a um acidente perigoso no local de trabalho.

#### 2.2.1.4 LAP e Virtek

LAP [23] e Virtek [24] são fornecedores de sistemas de assistência, cujas soluções são baseadas em projeção a laser, diretamente na estação de trabalho, combinada com uma câmara para a interação e coordenação entre o humano e o sistema. Tais sistemas contêm Software para coordenação em tempo real entre a câmara e o projetor.

- LAP

LAP produz soluções para várias indústrias, como a de terapia por radiação, aeroespacial, etc. Para o apoio ao operador em tarefas de montagem, este deve inicialmente especificar o produto que pretende montar, para depois ser projetado, no local de trabalho, o nome do primeiro componente e o local exato onde o operador o deve colocar. Depois, este deve carregar num botão, para que a instrução seguinte seja projetada (peça seguinte e local de montagem). Neste sistema, a câmara é apenas usada para ajudar na calibração do projetor. Com esta assistência, o processo é acelerado e a ocorrência de erros é reduzida significativamente [25].

- Virtek

Virtek produz sistemas baseados em projeção laser, que vende posteriormente a empresas de manufatura para suporte aos operadores. Estes sistemas consistem em:

- Projeção laser 2D: ajuda os operadores em tarefas de montagem de materiais planos (em "2D", ou seja, com espessura muito reduzida).
- Posicionamento visual 3D: ajuda os operadores na montagem de componentes com formatos complexos em 3D.
- Posicionamento espacial 3D: combina a projeção laser com gravação através de uma câmara, para obter a localização de um componente específico.
- Inspeção de qualidade: avalia, em tempo real, a qualidade da montagem, identificando erros cometidos.

Estas soluções minimizam os erros devido a medições manuais e a desalinhamentos, melhorando simultaneamente o tempo de *setup* e o fluxo de trabalho, resultando na significativa redução da percentagem de produtos finais defeituosos e da repetição da montagem dos mesmos, tendo um impacto positivo na produtividade dos processos das empresas que adquirem estes sistemas. Informação extraída de [24].

#### 2.2.1.5 Homag

A empresa Homag [26] criou uma solução para a automatização do processo de corte de painéis em madeira, combinando a projeção LED para assistência ao operador, denominada *intelliGuide* [27]. Ao usar *intelliGuide*, o operador começa por seleccionar num tablet o painel, ainda sem cortes, e o produto final que pretende obter. O programa consequentemente apresenta as instruções que o operador deve executar, passo a passo. Além disso, o feixe LED indica ao operador onde deve colocar o painel que irá ser cortado. Depois, o sensor identifica o momento de colocação do painel, e se este foi pousado na orientação correta, e posteriormente o sistema efetua automaticamente os cortes respetivos. *IntelliGuide* contém também um sistema de projeção, em forma de setas, sobre o painel, com a finalidade de guiar o operador ao longo do processo. Estas setas indicam os movimentos de translação e rotação que o operador necessita de fazer no painel, e as mesmas são ajustadas em tempo real devido a uma câmara que capta os movimentos, processando-os.

#### 2.2.1.6 Assembly Solutions

Assembly Solutions [28] é uma equipa de Engenheiros Industriais e de Tecnologias da Informação, localizada em Detmold, que desenvolveu também uma tecnologia que não envolve um ambiente virtual. A solução proposta consiste na projeção, diretamente no local de trabalho, das instruções passo a passo que devem ser conduzidas pelo operador em tarefas de montagem. Adicionalmente, e tendo em conta que os diferentes componentes encontram-se organizados em caixas, cada uma destas contendo uma tira de LEDs, para cada instrução os LEDs correspondentes às caixas que contêm os componentes necessários à sua execução são destacados. O operador pode optar por receber as instruções através de voz, gestos, ou por toque.

## 2.2.2 Empresas produtoras e consumidoras de sistemas

### 2.2.2.1 Siemens

Siemens [29] é um conglomerado industrial que cria também sistemas para treino virtual, denominados *COMOS Walkinside Immersive Training Simulator (ITS)* [30]. Estes sistemas são similares ao produzido pela empresa Appsbow, *VTE*: reproduz cenários da fábrica com elevado grau de realismo, tem a opção de treino individual ou com vários trabalhadores, as tarefas são apresentadas num ecrã, com a respetiva descrição, por texto. Possui também a opção de reproduzir situações de emergência. Como o sistema *VTE*, *COMOS* é útil para que os utilizadores fiquem familiarizados com a infraestrutura da empresa, bem como para o treino da execução de tarefas complexas e para a preparação caso ocorram eventos inesperados, que ponham em risco a saúde física do operador.

### 2.2.2.2 Bosch

*ActiveAssist* é o nome dado pela Bosch [31] à sua solução tecnológica para assistência ao operador. *ActiveAssist* [32] é um sistema baseado numa câmara e num projetor que, tal como o sistema desenvolvido pela equipa Assembly Solution, usa o projetor para exibir ao operador as instruções passo a passo diretamente no local de montagem. Outra similaridade entre os dois sistemas é que as caixas onde os componentes necessários estão guardados são destacadas. A câmara está constantemente a gravar os movimentos do operador e, quando este finaliza um passo, este pode avançar para o passo seguinte passando a mão por cima da área de projeção. Adicionalmente, a câmara identifica também quando uma instrução não é corretamente executada, retornando uma mensagem de erro, também no local de montagem.

### 2.2.2.3 Fraunhofer

Fraunhofer [33] é a organização líder da Europa na investigação orientada a aplicações, cobrindo áreas desde a saúde e ambiente, segurança e transporte, ou até comunicação e conhecimento.

Para a assistência ao operador em processos de produção, esta empresa implementou uma solução baseada num robot que contém um projetor laser [34]. Este robot fornece assistência visual, projetando a posição de montagem na estação de trabalho, sendo a interação entre o operador e a tecnologia feita através de reconhecimento do toque. Depois da montagem manual estar finalizada, o robot inspeciona a peça final para detetar erros, e caso estes existam é retornada uma descrição dos mesmos por intermédio de um ecrã.

## 2.3 Sumário

Neste capítulo, foi feito um estudo sobre a investigação já realizada no âmbito do desenvolvimento de sistemas de apoio ao operador em processos de manufatura. Este estudo permitiu

agrupar as tecnologias mediante as características em comum, e tirar conclusões sobre que tipo de sistema é mais adequado para cada tipo de tarefa ou processo. Explorou-se também o conceito de *Projection Mapping*, bem como os modelos matemáticos pelos quais são representados uma câmara e um projetor, juntamente com técnicas de calibração destes dois componentes, que são frequentemente usados em sistemas de apoio ao operador. Por último, fez-se um estudo do mercado, no sentido de analisar a relação entre os vários tipos de sistemas que algumas empresas criaram e as características das tarefas para as quais tais sistemas se destinam.



## Capítulo 3

# Escolha do Hardware e das ferramentas de Software

### 3.1 Hardware

Tendo em conta as características da tarefa para a qual este estudo se destina, e como mencionado no capítulo 1, pretende-se que a presença do *Hardware* não restrinja os movimentos do operador. Além disso, para que o operador não tenha de alternar o olhar entre um ecrã e o plano no qual as peças serão coladas, é preferível que o sistema auxilie diretamente no último. Como os operadores trabalham várias horas seguidas na colagem, também não é desejável que estes tenham de usar um *head-mounted display* ou qualquer outro dispositivo, de forma a preservar o seu conforto. Optou-se, portanto, pelo estudo de um sistema baseado num sensor e num atuador, sendo eles uma câmara e um projetor, respetivamente. Enquanto a câmara é responsável pela identificação do plano de colagem, para posterior identificação dos pontos onde o operador deve colar cada peça, o projetor tem como função projetar diretamente em cada ponto de colagem, dando ao operador a indicação para colar naquele local.

Ambos os componentes foram estrategicamente fixados numa estrutura de forma a que o campo de visão do projetor abrangesse todo o local de trabalho. Foi também garantido que a presença desta estrutura não incomode o trabalhador, preservando o seu conforto e fluxo de trabalho.

#### 3.1.1 Sensor

Como sensor foi utilizada uma câmara e, na escolha desta, dois principais tipos foram inicialmente considerados:

- câmaras RGB

câmaras convencionais, cujas imagens contêm duas dimensões, nas quais cada píxel é caracterizado por três canais: vermelho (R de red), verde (G de green) e azul (B de blue). São as mais utilizadas para aplicações de processamento de imagem em duas dimensões, sendo

também possível obter uma reconstrução 3D de um cenário através de, por exemplo, uma câmara e manipulação da iluminação do cenário, como o sistema implementado em [35], ou usando duas câmaras RGB calibradas e a triangulação dos seus sistemas de coordenadas, como em [36].

- Câmaras RGB-D

Estas câmaras, para além dos valores das cores primárias, medem também a profundidade (D de depth) de cada píxel, em relação ao sensor, acrescentando assim uma terceira dimensão às suas capturas. Desta forma, uma única câmara pode ser utilizada em aplicações de processamento 3D de imagem sem necessitar de pré-processamento.

Numa abordagem inicial ao problema foi utilizada uma câmara RGB, mas rapidamente se constatou que a falta de informação relativa à terceira dimensão da imagem iria dificultar bastante o processamento necessário ao cumprimento dos objetivos da dissertação. Considerou-se, então, essencial a obtenção da informação tridimensional do local de trabalho, para a mais precisa deteção do tronco de colagem, tal como de cada plano onde esta será efetuada pelo operador. Além disso, o uso de múltiplas câmaras RGB ou de Hardware adicional para a reconstrução 3D não seria adequado, quer devido à acrescida dimensão que o suporte do Hardware necessitaria, o que dificultaria a colocação do mesmo de forma a não comprometer o conforto do operador, quer devido ao peso computacional que a reconstrução propriamente dita exigiria, em comparação ao uso de uma única câmara RGB-D. Posto isto, optou-se por usar uma câmara deste tipo. A Microsoft Kinect V1 e a Intel Realsense D435 foram escolhidas para o teste da implementação, por serem as disponíveis no momento.

### 3.1.1.1 Microsoft Kinect V1

A Microsoft Kinect V1 foi a primeira versão da linha de câmaras de deteção de movimento Kinect, desenvolvida pela empresa Microsoft, lançada para o mercado em 2010. Esta câmara foi inicialmente desenvolvida para ser usada em jogos da consola XBox360, mas as suas propriedades, juntamente com o seu baixo custo em comparação com as restantes câmaras que possuem as mesmas capacidades, fez com que esta fosse também requirida para projetos que envolvessem processamento de imagem em três dimensões. A Microsoft, em 2012, lançou um SDK (Software Development Kit) para Windows, de forma a facilitar o desenvolvimento de Software para aplicações comerciais [37].

A tabela 3.1 apresenta as principais especificações operacionais da Microsoft Kinect V1 (informação retirada de [38]):

Tabela 3.1: Características da câmara Microsoft Kinect V1

| <b>Especificação</b>                     | <b>Valores do sensor</b> |
|--|--------------------------|
| Tipo de sensor de profundidade           | <i>Structured Light.</i> |
| Resolução da câmara RGB                  | 640x480.                 |
| Resolução da câmara de IR                | 320x240.                 |
| Amplitude angular horizontal operacional | 57°.                     |
| Amplitude angular vertical operacional   | 43°.                     |
| Profundidade mínima detetada             | 0.4m.                    |
| Profundidade máxima detetada             | 4.5m.                    |

### 3.1.1.2 Intel Realsense D435

Esta câmara pertence à linha de câmaras da Intel D400, e foi desenvolvida para ser integrada em projetos de desenvolvimento de processamento de imagem 3D. É indicada como ideal em ambientes com pouca iluminação, e possui um largo campo de visão, sendo útil em sistemas cuja captação da câmara deva abranger uma área considerável. A empresa produtora desta linha desenvolveu um Software de visualização, *Realsense Viewer*, que permite ao utilizador ver as imagens, quer RGB quer de profundidade, captadas pela câmara. Este, para visualização de nuvens de pontos, possui uma ferramenta de auto calibração do sensor, bem como um menu com vários parâmetros, para calibração manual do mesmo.

A tabela 3.2 apresenta as principais especificações operacionais da Intel Realsense D435 (informação retirada de [39]):

Tabela 3.2: Características da câmara Intel Realsense D435

| <b>Especificação</b>                     | <b>Valores do sensor</b> |
|--|--------------------------|
| Tipo de sensor de profundidade           | <i>Active IR Stereo</i>  |
| Resolução da câmara RGB                  | 1920x1080.               |
| Resolução da câmara de IR                | 1280x720.                |
| Amplitude angular horizontal operacional | 86°.                     |
| Amplitude angular vertical operacional   | 57°.                     |
| Profundidade mínima detetada             | 0.11m.                   |
| Profundidade máxima detetada             | 10m.                     |

### 3.1.2 Atuador

Como atuador, duas opções foram também consideradas, sendo elas o uso de um projetor ou de um apontador laser. No caso do laser, o modelo seria RYS1230, e o controlo da sua orientação seria efetuado por dois servos SG90, montados num suporte *pan and tilt*, alimentados e programados através de um microcontrolador Arduino Uno. Para a tomada de decisão, foi efetuada uma análise hierárquica do processo (AHP). Esta análise é efetuada quando uma decisão tem de ser feita entre duas ou mais opções, e consiste na escolha de um conjunto de parâmetros de decisão, devendo estes ser importantes no caso em estudo, e na atribuição de um peso a cada um destes parâmetros,

dependendo da relevância do mesmo. A cada opção é associada uma classificação numérica em relação a cada parâmetro, e a classificação final é a soma pesada das classificações. A alternativa com maior classificação final é a escolhida [40].

Neste caso as opções foram, portanto, o uso de um projetor ou de um laser, e os parâmetros escolhidos foram:

- Volume ocupado. O volume ocupado é importante pois terá efeito no volume do suporte do Hardware, que deve ser tido em conta para cumprir o requisito de assegurar o conforto do operador. Quanto menor for o volume ocupado, maior a classificação.
- Custo total de posse. Sendo que o atuador não só terá de ser adquirido, mas também requerirá manutenção periódica para assegurar o seu bom funcionamento a longo prazo, esta despesa estará ao encargo da empresa que usufruirá do sistema. Posto isto, é importante contabilizar os custos diretos e indiretos da utilização de cada um. Quanto menor for o custo total de posse, maior a classificação.
- Tempo de vida. É também importante considerar o tempo que um atuador consegue durar até ter de ser substituído, pois esta substituição, para além do custo monetário associado, poderá implicar também a recalibração do componente. Quanto maior o tempo de vida, maior a classificação.
- Capacidade e performance. Este parâmetro avalia a capacidade do atuador no que diz respeito à manipulação dos contornos a projetar, bem como da precisão da projeção nas posições pretendidas. Quanto maior a capacidade e precisão mencionadas, maior a classificação.
- Complexidade de programação. Avalia a facilidade relativa à calibração e programação do componente. Quanto menor for a complexidade de programação, maior a classificação.

Na tabela 3.3 encontra-se o peso atribuído a cada parâmetro, bem como a classificação de cada opção, relativamente a cada parâmetro, ambos numa escala de 0 a 1. Por fim, a soma pesada é calculada.

Tabela 3.3: Tabela de decisão do atuador.

|                                   | <b>Projetor</b> | <b>Laser</b>  |
|-----------------------------------|-----------------|---------------|
| Volume ocupado (0.2)              | 0.1             | 0.7           |
| Custo total de posse (0.2)        | 0.25            | 0.7           |
| Tempo de vida (0.15)              | 0.6             | 0.15          |
| Capacidade e performance (0.35)   | 0.7             | 0.2           |
| Complexidade de programação (0.1) | 0.15            | 0.8           |
| <b>Classificação final</b>        | <b>0.42</b>     | <b>0.4525</b> |

Utilizou-se, portanto, um ponteiro laser como atuador para a projeção nos locais de colagem.

## 3.2 Ferramentas de Software

A crescente intensificação no estudo das capacidades de processamento de imagem, por parte de investigadores, em setores como a saúde (como na avaliação de exames de imagiologia) [41], a indústria (como no fornecimento de capacidades visuais a aplicações robóticas ou de controlo de qualidade) [42], ou segurança (como no reconhecimento de faces ou de pessoas) [43], levou ao rápido desenvolvimento de aplicações cada vez mais complexas e robustas, culminando na criação de várias bibliotecas para permitir aos programadores uma utilização mais direta dos variados algoritmos. Entre estas, duas foram utilizadas neste projeto: **Point Cloud Library (PCL)** e **OpenCV**. **Robotic Operative System (ROS)** foi também usado devido à sua arquitetura modular, por já possuir drivers para a Microsoft Kinect e para a Intel Realsense, e por ser facilmente utilizado em conjunto com PCL e OpenCV. **Meshlab** foi útil para a edição manual de nuvens de pontos, e o controlo do laser foi realizado através da biblioteca de **Arduino** para servos.

### 3.2.1 ROS (Robot Operating System)

ROS (Robot Operating System) consiste num conjunto de bibliotecas e ferramentas de Software para aplicações robóticas. O seu *design* distribuído e modular permite que cada utilizador necessite apenas de dominar as funcionalidades requeridas para a execução da aplicação pretendida, sendo por isso adequado para projetos de variados graus de complexidade.

A sua arquitetura é constituída por quatro componentes principais: nós, tópicos, mensagens e serviços [44].

- **Nós:** Um nó é um programa responsável pela execução de uma determinada função, podendo esta ser, por exemplo, o controlo ou a medição de um componente de Hardware, a deteção de uma região de interesse, o cálculo da pose de um robot, etc. A definição de um nó é a melhor representação da característica modular de ROS, visto que tipicamente uma aplicação possui múltiplos nós, sendo cada um responsável por uma componente do sistema. Pelo atrás exposto, o número de nós aumenta com a complexidade do sistema.
- **Tópicos:** Um tópico é um meio de comunicação entre dois nós. Por exemplo, numa aplicação em que um nó seja responsável pelo cálculo da trajetória a efetuar por um robot e um outro nó seja responsável pelo controlo dos motores para a movimentação do mesmo, é através de um tópico que ambos comunicam de forma a que o robot se movimente efetuando a trajetória pretendida. Cada tópico tem associado um tipo de dados.
- **Mensagens:** Um nó pode subscrever ou publicar uma mensagem num tópico. Ao publicar, o nó torna-se capaz de enviar mensagens através desse tópico, e ao subscrever o mesmo torna-se capaz de receber as mensagens que estão a ser publicadas nesse tópico por um outro nó. Uma mensagem é, portanto, a unidade de comunicação entre dois nós. Visto que todos os tópicos têm associado um tipo de dados, o tipo da mensagem que é transmitida através do mesmo tem de ser do mesmo tipo.

- **Serviços:** Serviços são um outro possível meio de comunicação entre nós que, ao invés de comunicação publicador/subscritor, característico dos tópicos, permitem comunicação pedido/resposta entre nós. Outra diferença importante entre tópicos e serviços é que, enquanto um tópico tem associado um único tipo de dados, que tem de ser respeitado quer pelo nó subscritor quer pelo nó publicador, no caso de um serviço o tipo de dados que efetua o pedido tem uma estrutura diferente em comparação com o tipo de dados que envia a resposta.

ROS possui também importantes ferramentas, como **RViz**, utilizada para visualização, e **rqt**, que contém *plug-ins* para visualização do sistema de nós, tópicos e respetivas ligações de subscrição/publicação (*rqt\_graph*) e para configuração dinâmica, que permite ao utilizador aceder e editar parâmetros de alguns componentes do sistema (*rqt\_reconfigure*), entre outros.

A decisão de utilizar ROS baseou-se no facto de este já possuir drivers para a Microsoft Kinect e para a Intel Realsense. Estas câmaras publicam quer a sua captação RGB quer a de profundidade em tópicos, pelo que basta instalar este driver para aceder às imagens através da subscrição ao tópico pretendido. É também possível subscrever um tópico que contém ambas captações de profundidade e RGB, calibradas uma em relação à outra. Outra importante característica é que, para além da imagem de profundidade, o nó da câmara publica também esta imagem em três dimensões, na forma de uma nuvem de pontos.

### 3.2.2 PCL (Point Cloud Library)

Uma nuvem de pontos é a representação espacial, em três dimensões, de uma cena através de um conjunto de pontos, tipicamente gerado por um scanner 3D a laser. Cada medida do scanner gera um ponto, guardando as respetivas coordenadas X, Y e Z, e após concluir todas as medições os pontos são agrupados, resultado numa nuvem de pontos [45]. Opcionalmente, o scanner pode também retornar a coloração RGB de cada ponto, se o sensor utilizado o permitir. A nuvem de pontos original da Microsoft Kinect V1 tem 307200 pontos, devido à sua resolução 640x480.

PCL é um projeto de uso livre para processamento de nuvens de pontos, que fornece aos seus utilizadores algoritmos padrão de filtragem, registo e reconhecimento, entre outros, estando estes organizados por módulos. Por outras palavras, algoritmos de filtragem encontram-se inseridos no módulo "filtering", e o mesmo se aplica aos restantes grupos mencionados. O projeto tem também módulos para conexão e leitura de sensores, e para visualização de nuvens de pontos [46].

PCL guarda a informação de cada nuvem num vetor de pontos, cada um contendo a respetiva posição segundo os eixos X, Y e Z no espaço, em relação a uma origem, e opcionalmente os respetivos valores RGB.

### 3.2.3 Meshlab

Meshlab é um Software gratuito para visualização e edição quer de malhas triangulares quer de nuvens de pontos. O programa contém ferramentas para alinhamento, reconstrução, processamento de cor, análise e medição, e texturização de malhas e nuvens. É também utilizado para a construção de modelos 3D para posterior impressão [47].

Com Meshlab, e guardando uma nuvem de pontos num ficheiro *.ply*, é possível abrir esse ficheiro no Software e manualmente remover os pontos necessários de forma a que, após a edição, fique na nuvem apenas os pontos de interesse. Através das classes de PCL *pcl::PLYWriter*, *pcl::PLYReader* e *pcl::PCDWriter* é possível guardar uma nuvem de pontos num ficheiro *.ply*, ler o conteúdo de um ficheiro *.ply* e guardar este num ficheiro *.pcd*, respetivamente.

### 3.2.4 OpenCV

OpenCV é uma das mais utilizadas e versáteis bibliotecas gratuitas para aplicações de visão por computador. A sua popularidade é explicada pela quantidade de algoritmos implementados, permitindo o seu uso em programas complexos, sem a necessidade de perceber toda a teoria por trás de todos os passos interínsecos de cada algoritmo, juntamente com a constante manutenção e atualização da biblioteca e qualidade da organização da sua documentação. OpenCV foi desenvolvido em C++, mas pode também ser utilizada em aplicações programadas em Python, Java e Matlab [48].

OpenCV guarda a informação de cada imagem num objeto *Mat*, que possui um cabeçalho contendo informação geral relevante da imagem, como a sua codificação, dimensão, etc., e uma matriz cujos elementos contêm o valor de cada píxel da imagem propriamente dita. A biblioteca foca-se em aplicações de processamento de imagens 2D, sendo que é também possível obter a reconstrução 3D ao combinar múltiplos objetos *Mat* num vetor, cada um representando uma "fatia" 2D da imagem 3D.

### 3.2.5 Arduino

Arduino é uma linha de placas eletrónicas que possuem geralmente um microcontrolador Atmel AVR, um conjunto de pinos de entradas e saídas, para a ligação deste a outros componentes eletrónicos, pinos de alimentação, entre outros. Possui também uma linguagem de programação característica, baseada em C/C++ simplificado, bem como um ambiente de desenvolvimento integrado de uso gratuito, e um conjunto vasto de bibliotecas para facilitar a leitura e controlo dos valores dos componentes ligados à placa [49].

Um programa executável em Arduino, denominado de "sketch", necessita de, no mínimo, duas funções para ser funcional: *setup()*, executada uma única vez, onde é efetuada a configuração dos componentes e a respetiva associação às portas de entrada e saída da placa; e *loop()*, corrida repetidamente enquanto a placa estiver eletricamente alimentada, na qual as instruções de controlo dos componentes são executadas [50].

## 3.3 Sumário

Neste capítulo, foram apresentadas as opções de Hardware consideradas para uso durante a implementação, bem como o método de escolha destes componentes. Foram ainda mencionadas e brevemente descritas as principais ferramentas de Software utilizadas.



## Capítulo 4

# Implementação

O trabalho conduzido nesta dissertação será apresentado neste capítulo. Inicialmente, será apresentada uma lista de definições, que será respeitada ao longo do resto do relatório. Depois, será feita a decomposição da implementação nas suas partes principais. Por último, cada uma destas partes é descrita e explicada.

### 4.1 Definições

Antes da explicação relativa à decomposição e ao funcionamento da implementação desenvolvida, foi feita uma lista de palavras-chave, de forma a manter a coerência ao longo do documento:

1. **Plano de colagem:** plano onde o operador cola um número estipulado de peças.
2. **Tronco de colagem:** estrutura em cera cujos planos são os planos de colagem.
3. **Suporte:** estrutura metálica que suporta o tronco de colagem
4. **Bacia:** estrutura em cerâmica que une o tronco de colagem ao seu suporte.
5. **Ponto de colagem:** ponto onde se pretende que o operador cole uma peça.
6. **Ponto de projeção:** ponto onde se encontra a projeção do ponteiro laser, no plano da imagem da câmara.
7. **Ciclo de colagem:** ciclo que começa quando o operador cola a primeira peça num plano de colagem e termina quando o mesmo cola a última peça nesse plano.
8. **Servo superior:** Servo que roda verticalmente, variando verticalmente a projeção do apontador laser.
9. **Servo inferior:** Servo que roda horizontalmente, variando horizontalmente a projeção do apontador laser.
10. **Peça:** Equivalente a molde.

## 4.2 Arquitetura do Sistema

Como mencionado no capítulo 3, o sistema consiste num par câmara-apontador. O sistema é composto por duas partes principais: a localização dos pontos de colagem no plano de colagem, que tem também de ser identificado, e o controlo dos servos no sentido de projetar o ponteiro laser nestes pontos, que inclui a deteção do ponto de projeção naquele instante, e a realimentação dos servos em malha fechada, para a convergência deste ponto no ponto de colagem pretendido.

O código desenvolvido encontra-se dividido em quatro nós, estando os três primeiros inseridos num mesmo pacote em ROS, e o último num *sketch* em Arduino:

- **processing\_node**: recebe como entrada a nuvem de pontos da câmara, com informação RGB, e é responsável pelo processamento necessário à identificação dos pontos de colagem e do número de peças já coladas, publicando o ponto de colagem da peça a ser colada.
- **laser\_node**: recebe a imagem RGB da câmara, em duas dimensões e sem pré-processamento, e tem como função o cálculo do ponto de projeção nesse instante. Este nó publica, então, o valor desse ponto.
- **servos\_node**: subscreve os valores publicados pelos nós *processing\_node* e *laser\_node*, e compara os valores de largura e comprimento do ponto de colagem com o ponto de projeção. Subscreve também a estimativa dos ângulos atuais dos servos, e após os atualizar publica os novos valores, de forma a que o ponto de projeção convirja no ponto de colagem.
- **arduino\_node**: responsável por publicar a estimativa dos ângulos atuais dos servos, e após atualização destes subscreve os novos ângulos e envia-os para os servos, provocando a rotação física destes.

Na figura 4.1 encontram-se representados os nós mencionados, por círculos. Estão também presentes os tópicos através dos quais foram trocadas mensagens entre os nós, representados por retângulos, o tipo de dados destes, por baixo do retângulo respetivo, e as relações de subscrição/publicação com cada nó, representadas por setas.

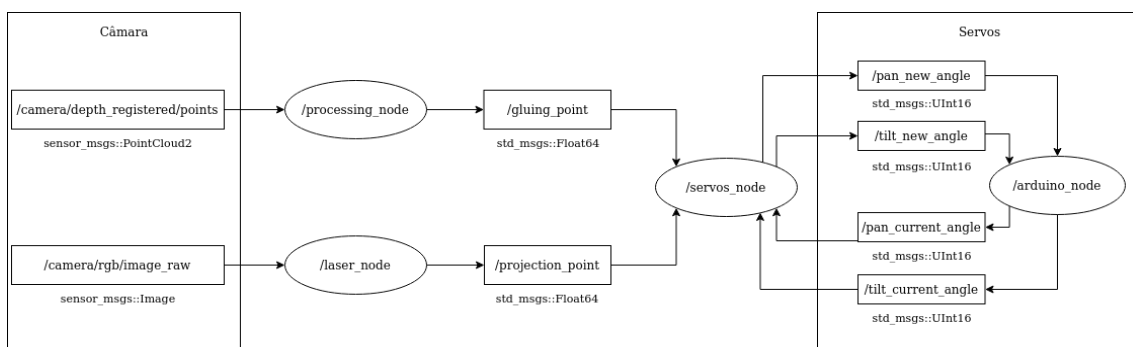


Figura 4.1: Arquitetura do sistema.

Para demonstrar as interações entre os diferentes nós, bem como as funções principais de cada um, foi feito um diagrama de sequência UML, apresentado na figura 4.2.

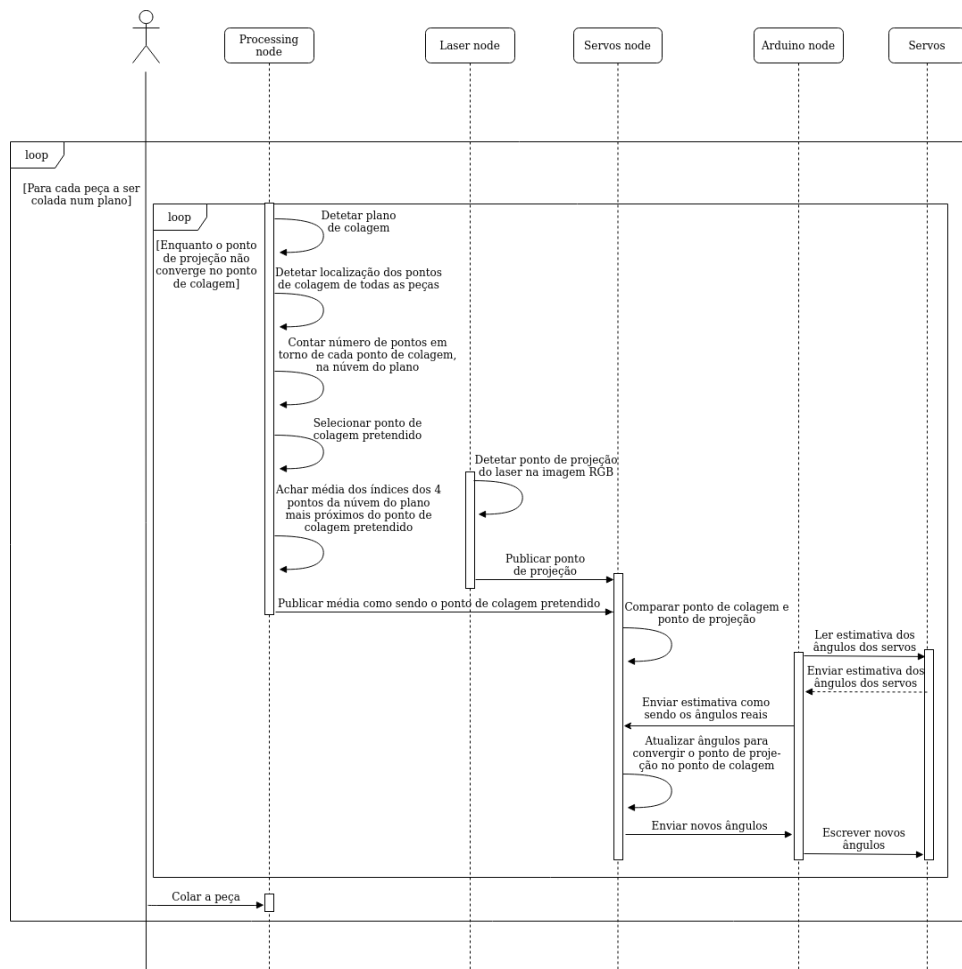


Figura 4.2: Diagrama de sequência UML do sistema.

De seguida, serão descritas e explicadas as diferentes partes do sistema.

### 4.3 Processamento da nuvem de pontos da câmara

Esta primeira parte recebeu, como entrada, a nuvem de pontos que contém a informação captada pela câmara de profundidade, com informação RGB, e engloba todo o processamento desta nuvem com a finalidade de publicar o índice do ponto da nuvem que corresponde ao ponto de colagem pretendido. Esta representação é justificada pelo facto de que o índice de cada ponto corresponde à sua posição na imagem de profundidade, em duas dimensões. Tal verifica-se pois o número de píxeis de uma imagem 2D publicada pela câmara de profundidade é igual ao número de pontos da nuvem original (sem processamento) publicada pela mesma câmara. Por exemplo, o ponto com índice 2 da nuvem de pontos corresponde ao píxel [0,2] da imagem de profundidade

correspondente. A saída deste sub-sistema é, portanto, o índice do ponto da nuvem que corresponde ao ponto de colagem da peça seguinte.

Para atingir este fim, inicialmente foi feita uma filtragem de forma a que a nuvem de pontos seja constituída unicamente pelo tronco de colagem, para permitir a posterior deteção do plano de colagem. Depois, foi criada uma nuvem contendo apenas o ponto de colagem de todas as peças. Analisando o número de pontos presentes na nuvem do plano de colagem, em torno dos pontos de colagem já identificados, foi possível obter o número de peças já coladas, permitindo detetar qual a próxima peça a ser colada pelo operador, e o respetivo ponto de colagem. Por fim, e tendo em conta que durante este processo foi necessário efetuar transformações no sistema de coordenadas, as mesmas foram invertidas, para que seja publicado o ponto de colagem da peça no sistema de coordenadas da câmara. Todo o código desenvolvido encontra-se no nó *processing\_node*, estando o algoritmo desenvolvido apresentado na figura 4.3.

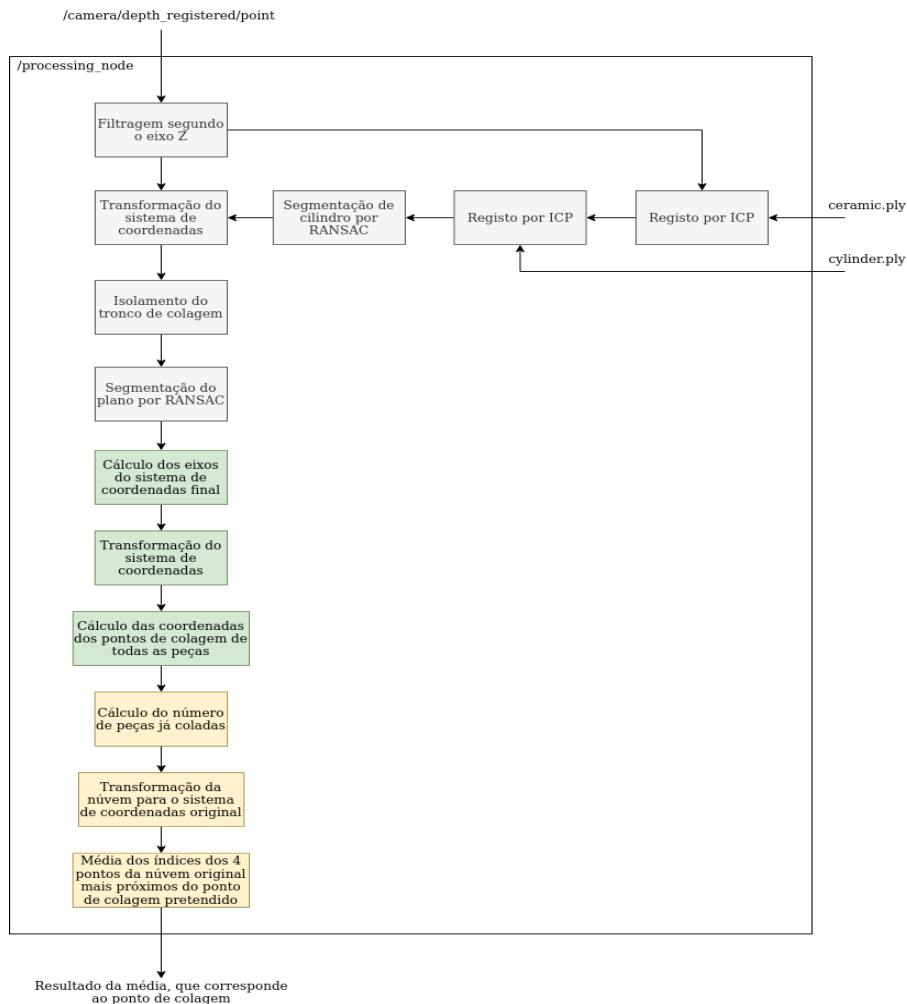


Figura 4.3: Algoritmo de processamento da nuvem de pontos.

### 4.3.1 Detecção do plano de colagem

Ao subscrever o tópico `/camera/depth_registered/points`, o nó subscritor recebe mensagens `sensor_msgs::PointCloud2`, que é o mais recente tipo de mensagens ROS que representam uma nuvem de pontos. Através da função `pcl::fromPCLPointCloud2()` a mensagem recebida é convertida num objeto da classe `pcl::PointCloud`, podendo depois ser processado através de funções da biblioteca PCL.

Após a subscrição e conversão, procedeu-se à filtragem da nuvem para isolamento do tronco. Sendo que a distância do tronco à câmara é conhecida (aproximadamente 85 centímetros), aplicou-se um filtro passa-banda segundo o eixo Z, com limites mínimo e máximo de 0.6 metros e 1 metro, respetivamente. Desta forma, todos os objetos que estejam a uma distância, relativamente à câmara, inferior a 0.6 metros foram removidos, tal como a mesa onde o suporte está pousado e o restante fundo, cuja distância à câmara é superior a 1 metro.

Visto que o tronco pode não estar alinhado quer com os eixos das ordenadas quer com o das abcissas, foi determinado o eixo de orientação do tronco. Para isso, observou-se que o tronco encontra-se fixado na bacia, que pode ser facilmente reconhecida através de um algoritmo de registo apropriado. Esta estrutura foi útil pois contém uma secção com forma cilíndrica, cujo eixo coincide com o eixo de orientação do tronco. Ou seja, obtendo-se o seu eixo, obtém-se também a orientação do tronco de colagem. Para tal, gravou-se a nuvem de pontos filtrada no ficheiro `ceramic.ply`, e através do Software Meshlab foram manualmente removidos todos os pontos que não fizessem parte da bacia. O resultado final da edição encontra-se na figura 4.4.

Seguidamente aplicou-se o algoritmo de registo ICP, *Iterative Closest Point*, que consiste no cálculo da matriz de transformação que converge uma nuvem de pontos (de origem), numa outra nuvem (de referência), que se mantém fixa, de forma a minimizar a distância de todos os pontos da primeira na segunda. O algoritmo recebe então, como entrada, as nuvens de origem e de referência, e opcionalmente uma estimativa inicial da transformação, e associa todos os pontos da nuvem de origem aos da nuvem de referência com menor distância, calculando a matriz de transformação respetiva, repetindo este processo iterativamente [51]. Neste caso, a nuvem de pontos filtrada, captada em tempo real, foi usada como nuvem de referência, e a nuvem do ficheiro `ceramic.ply`, contendo já apenas a bacia, foi usada como nuvem de origem. Finalmente, para a obtenção do eixo do cilindro, o ficheiro foi novamente editado, até que restassem apenas os pontos do cilindro, e a nuvem resultante foi gravada no ficheiro `cylinder.ply`. Aplicou-se novamente o algoritmo ICP, com a nuvem da bacia (já transformada pela matriz de transformação retornada pelo algoritmo) como nuvem de referência, e a nuvem do cilindro como nuvem de origem. Nas figuras 4.5 e 4.6 estão presentes os resultados do alinhamento por ICP mencionados.

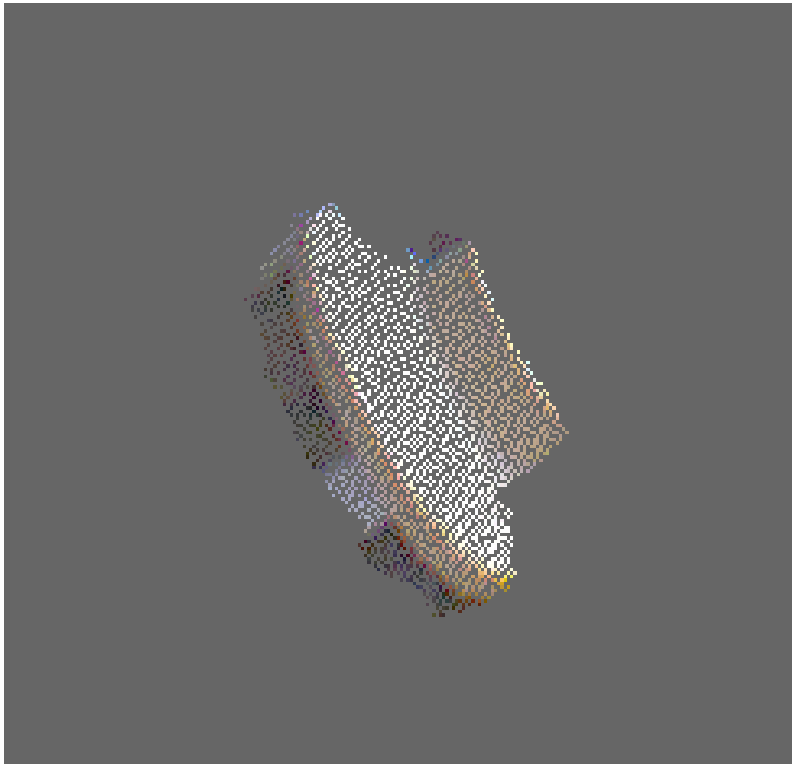


Figura 4.4: nuvem de pontos contendo apenas a bacia.

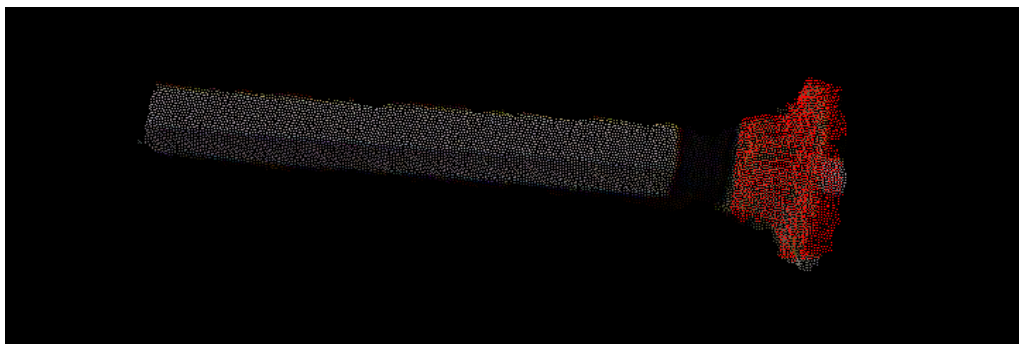


Figura 4.5: Alinhamento da bacia na nuvem de pontos filtrada.

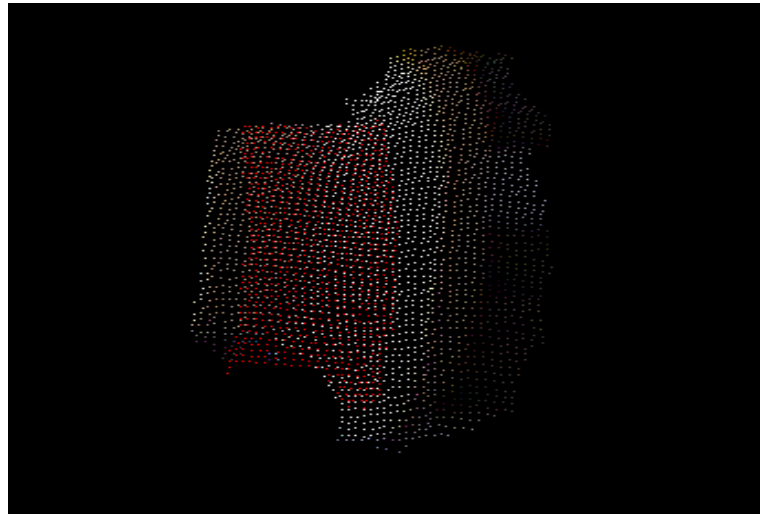


Figura 4.6: Alinhamento do cilindro na bacia.

Por fim, aplicou-se o algoritmo de segmentação RANSAC, *Random Sample Consensus*, para a detecção do cilindro. RANSAC é um método iterativo para a obtenção dos parâmetros de, por exemplo, um plano ou um cilindro. Inicialmente, este método seleciona aleatoriamente um conjunto de pontos, e calcula os parâmetros da equação matemática que representa a forma geométrica que se pretende obter, com base nesse conjunto. Depois, todos os restantes pontos da nuvem são avaliados como fazendo ou não parte desse modelo, com uma tolerância pré-estipulada. Caso o número de pontos pertencentes resultante seja superior a um limite, este modelo é considerado como válido, caso contrário é rejeitado. Este algoritmo é repetido um número fixo de vezes e, por último, são retornados os parâmetros do modelo válido cujos pontos pertencentes apresentem menor erro [52].

Sendo já conhecido o eixo pretendido, o sistema de coordenadas da nuvem foi transformado de forma a que o tronco fique orientado segundo o eixo Y. Esta transformação é útil pois, devido à presença do operador perto do tronco, é importante garantir que a nuvem de pontos seja maioritariamente composta por pontos pertencentes ao tronco de colagem. Assim, garante-se que o algoritmo RANSAC retorna os planos deste, em vez de um eventual plano detetado no corpo do operador. Para isso, e tendo em conta que as dimensões do tronco são conhecidas, a nuvem transformada foi filtrada segundo os eixos X e Y. Foi, no entanto, dada uma tolerância aos valores mínimo e máximo do filtro em cada eixo, no sentido de não comprometer o funcionamento do sistema caso o cálculo da orientação do tronco contenha algum erro.

Finalmente, foi aplicada a segmentação por RANSAC para detecção dos dois planos do tronco. Não tendo a garantia de que o primeiro plano detetado seja aquele em que o operador vai efetuar a colagem, o programa guarda inicialmente o primeiro plano num objeto e remove-o da nuvem, e guarda de seguida o segundo plano num outro objeto. Depois, é feita a comparação dos coeficientes de ambos os planos, e é selecionado o pretendido. Na figura 4.7, a vermelho, encontra-se o plano de colagem, e a azul o outro plano, também visível do ponto de vista da câmara.

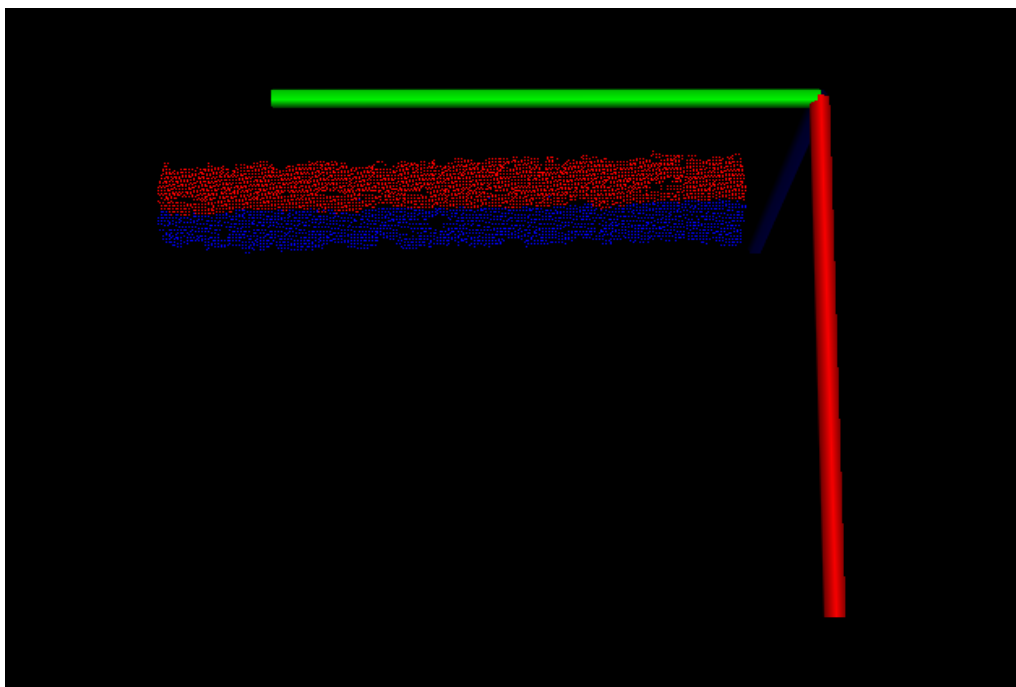


Figura 4.7: Planos do tronco detetados pela câmara.

### 4.3.2 Localização dos pontos de colagem

Para achar a localização dos pontos de colagem, foi efetuada uma nova transformação do sistema de coordenadas, de forma a que os eixos Y e Z do sistema transformado coincidam com duas arestas do plano de colagem. Para isso, e com a função *get\_planes\_intersection()*, foi calculada a linha da interseção destes planos, que contém uma das arestas do plano de colagem. A outra aresta necessária à transformação foi definida com a construção do polígono convexo que delimita todos os pontos da nuvem do plano de colagem, através de *pcl::ConvexHull.reconstruct()*, que retorna uma nuvem com os vértices que formam o polígono. Com esta última, e calculando o comprimento de cada aresta que une dois pontos do polígono, foi possível extrair o vetor correspondente à aresta em falta, da seguinte forma: idealmente, e visto que o plano de colagem tem forma retangular, bastaria extrair a segunda aresta com maior comprimento (visto que a aresta de maior comprimento seria a de interseção dos dois planos, já obtida). No entanto, a resolução limitada da câmara e o erro desta no cálculo da distância de cada ponto fazem com que o modelo de segmentação rejeite alguns pontos que pertencem ao plano, resultando numa nuvem com menos pontos do que idealmente teria. Assim, esta não possui formato perfeitamente retangular. Posto isto, o polígono convexo resultante contém mais do que os quatro vértices esperados. Uma vez que é conhecida a distância da origem do referencial à aresta do plano mais próxima a esta, optou-se então por uma filtragem inicial da nuvem de pontos do polígono, ficando apenas os segmentos que ligam dois pontos próximos da aresta mencionada, e entre estes foi escolhido o que tem maior comprimento. A figura 4.8 apresenta o resultado obtido na identificação das arestas do plano.

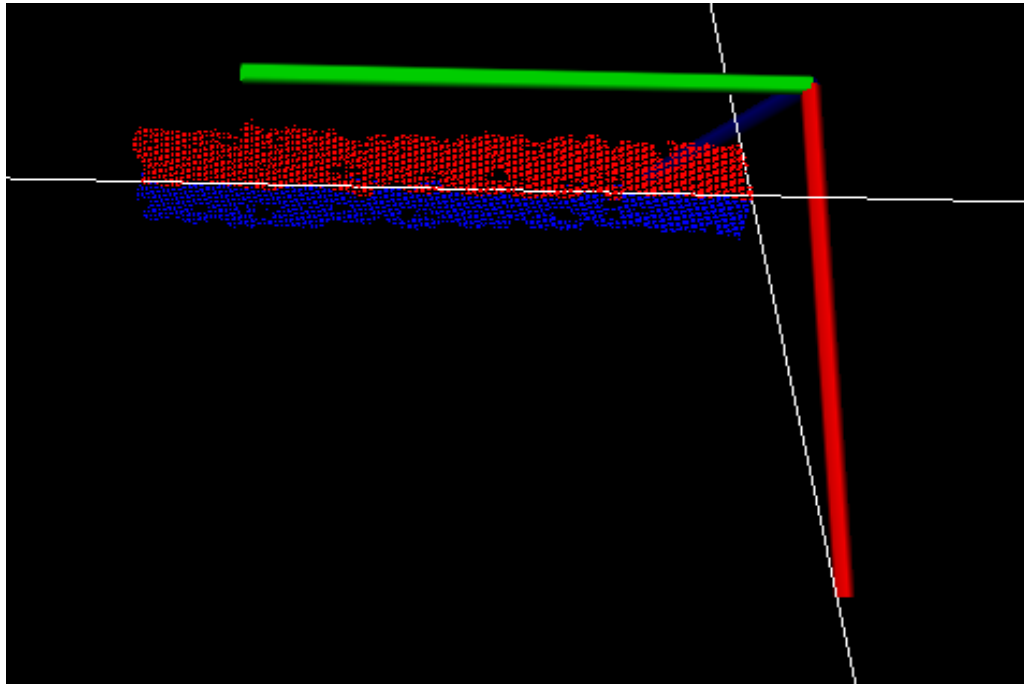


Figura 4.8: Eixos formados pelas arestas do plano de colagem.

Conhecendo-se já as arestas pretendidas, procedeu-se à transformação do sistema de coordenadas. A origem do novo sistema foi a interseção das duas arestas, calculada por *pcl::lineWithLineIntersection()*. Na figura 4.9 encontra-se o plano já alinhado com o novo sistema de coordenadas.

Para definir a localização dos pontos de colagem no plano, obteve-se as dimensões do plano de colagem, com *pcl::getMinMax3D*. Além disso, sabe-se que são coladas 10 peças por plano, e que estas devem estar equidistantes. Pretende-se também que o ponteiro laser projete sempre no centro da largura do plano. Não havendo informação sobre a distância entre o início do plano e o ponto onde a primeira peça deve ser colada, definiu-se que as coordenadas do ponto de colagem de cada peça  $i$  seriam definidas pelas equações:

$$gluing\_point.x = \frac{max.x + min.x}{2} \quad (4.1)$$

$$gluing\_point.y = \frac{max.y + min.y}{2} \quad (4.2)$$

$$gluing\_point.z = \frac{i * (max.z + min.z)}{10} - \frac{max.z + min.z}{10 * 2} \quad (4.3)$$

Onde as variáveis *min* e *max* são os objetos retornados pela função de PCL utilizada, contendo os valores mínimo e máximo, respetivamente, de X (em *min.x* e *max.x*), Y (em *min.y* e *max.y*) e Z (em *min.z* e *max.z*).

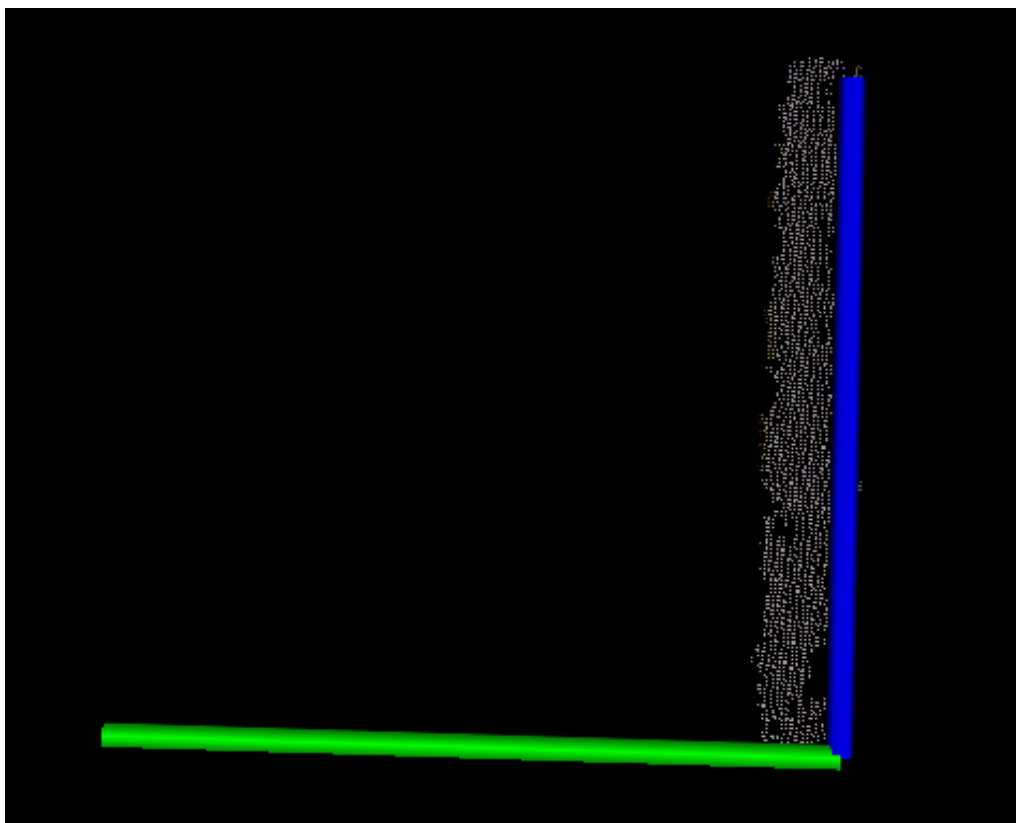


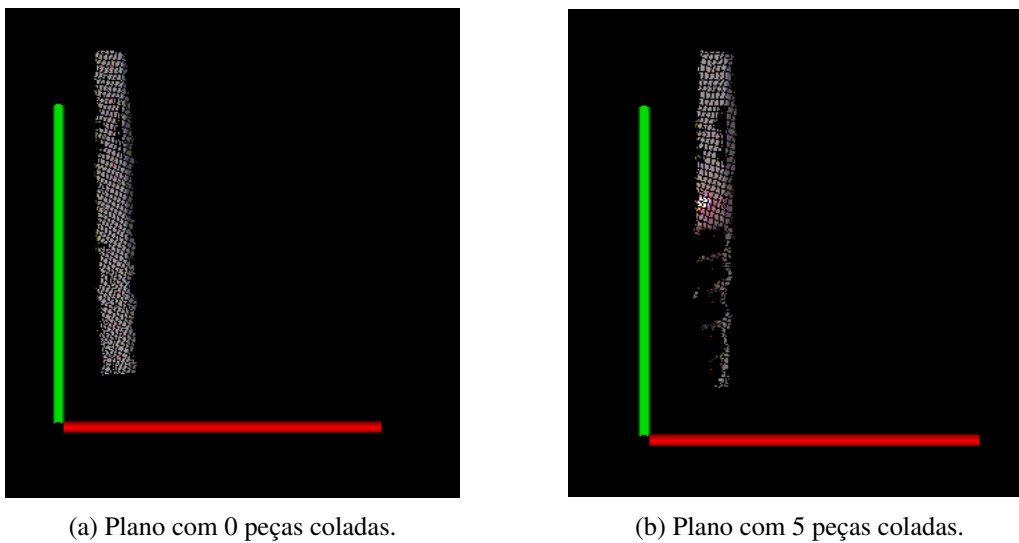
Figura 4.9: Plano de colagem transformado pelo novo sistema de coordenadas.

### 4.3.3 Cálculo da próxima peça a ser colada

Para determinar qual a próxima peça a ser colada, analisou-se o número de pontos em torno de cada ponto de colagem. Como a nuvem só contém pontos pertencentes ao plano, caso uma peça tenha sido já colada, verifica-se um buraco em torno do ponto de colagem respectivo, pois a presença da peça obstrui a detecção do plano nessa região. Posto isto, percorreu-se todos os pontos de colagem e calculou-se o número de pontos do plano dentro de uma esfera de raio 1 centímetro, cujo centro é o próprio ponto. Se o número calculado for menor que um limite, considera-se que já existe uma peça. A próxima peça a ser colada é, portanto, o primeiro ponto de colagem cuja esfera contenha mais pontos do que o limite estipulado. Nas figuras 4.10a e 4.10b encontra-se o plano de colagem com 0 e 5 peças, respetivamente.

Para obter a localização do ponto de colagem no sistema de coordenadas da câmara, necessária para a posterior comparação com o ponto de projeção do ponteiro laser, inverteu-se as duas transformações do sistema de coordenadas efetuadas anteriormente.

Por fim, foi preciso encontrar o ponto da nuvem original mais próximo das coordenadas resultantes de 4.1, 4.2 e 4.3 para a peça a ser colada. Para isso, identificou-se 4 pontos da nuvem original: os dois mais próximos segundo o eixo das abcissas, um no sentido positivo e outro no sentido negativo, e os dois mais próximos segundo o eixo das ordenadas, um no sentido positivo



(a) Plano com 0 peças coladas.

(b) Plano com 5 peças coladas.

Figura 4.10: Nuvem do plano de colagem, com 0 e 5 peças coladas

e outro no sentido negativo. Calculou-se a média dos índices desses pontos, e o resultado inteiro dessa média foi então publicado como sendo o índice do ponto de colagem.

## 4.4 Projeção do laser nos pontos de colagem

### 4.4.1 Detecção do ponto de projeção do laser

A imagem 4.11 resume o algoritmo desenvolvido para a detecção do ponto de projeção do ponteiro laser, em cada instante.

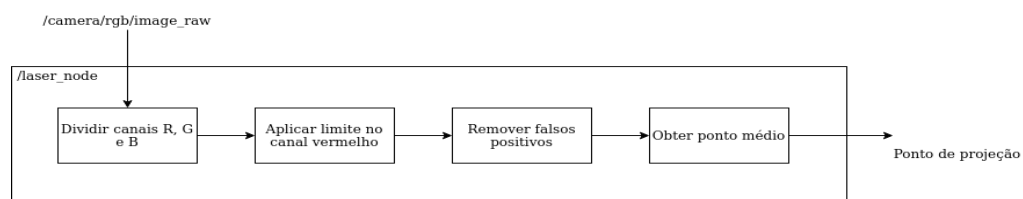


Figura 4.11: Algoritmo para detecção do ponto de projeção atual do ponteiro laser.

Todo o código desenvolvido nesta parte encontra-se no nó *laser\_node*, que subscreve o tópico */camera/rgb/image\_raw*, recebendo assim mensagens *sensor\_msgs::Image*. Estas correspondem à imagem RGB, em tempo real e em duas dimensões, da captação da câmara.

Com a função *cv\_bridge::toCvCopy()*, a mensagem recebida foi convertida num objeto *Mat*, podendo assim ser processada diretamente utilizando a biblioteca OpenCV. Após a conversão, e como o ponteiro laser possui uma cor predominantemente vermelha, com a função *split()* os três canais (vermelho, verde e azul) foram divididos, dando origem a um vetor de três objetos *Mat*, cada um contendo um único canal. Por outras palavras, cada canal constitui uma imagem em tons

de cinzento, cuja intensidade de cinza de cada píxel corresponde à intensidade do canal respetivo da imagem RGB.

Foi depois aplicado um filtro binário com `cv::threshold()`, limitando inferiormente a intensidade dos píxeis do objeto correspondente ao canal vermelho, de forma a que permaneçam na imagem apenas os pontos correspondentes à projeção do laser. Para evitar possíveis falsos positivos, bem como para reduzir ligeiramente o número de píxeis correspondentes à projeção, foi aplicada a função `cv::erode()` na imagem binária filtrada.

Por fim, foi calculada a média dos píxeis da imagem resultante, e esta foi publicada como sendo o ponto de projeção atual do atuador, para a posterior comparação com o ponto de colagem.

#### 4.4.2 Controlo dos servos

Sendo já conhecidos o ponto de colagem e o ponto de projeção, efetuou-se então o controlo dos servos. O código correspondente encontra-se dividido em dois nós: `servos_node`, escrito na linguagem C++, e `arduino_node`, escrito num `sketch` Arduino. A figura 4.12 apresenta o que é feito em cada nó, desta última parte do projeto.

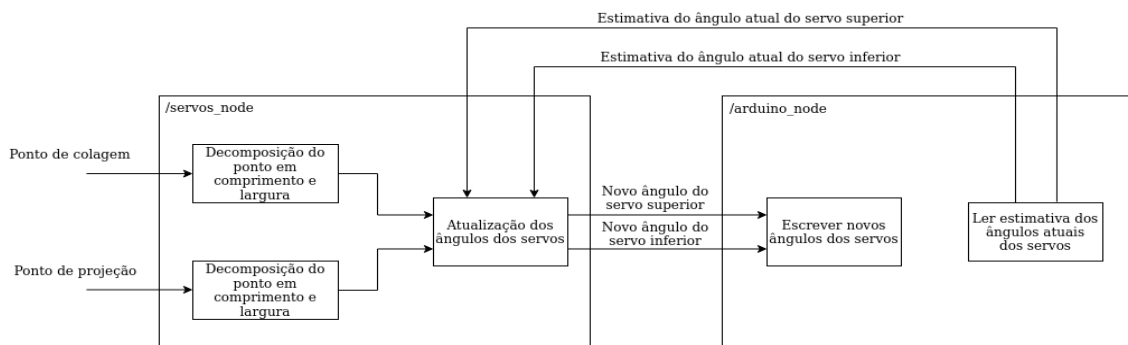


Figura 4.12: Algoritmo para controlo dos servos.

- `servos_node`

Este nó subscreve os tópicos `/pan_current_angle` e `/tilt_current_angle`, onde as mensagens que contêm o ponto de colagem e o ponto de projeção do ponteiro laser são publicadas, e os tópicos nos quais são publicadas as estimativas dos ângulos atuais de cada servo.

Inicialmente, foi feita a decomposição de cada um dos pontos subscritos nos valores de comprimento e largura correspondentes na matriz que representa a imagem 2D. Aqui, "comprimento" e "largura" referem-se à posição do píxel dentro da matriz de píxeis: o píxel com índice 1000, numa imagem com resolução 620x480, encontra-se na posição da matriz [1, 380], ou seja, o seu comprimento é 380 e a sua largura é 1. As fórmulas utilizadas para este cálculo foram:

$$pixel\_width = pixel\_index \% 640 \quad pixel\_length = pixel\_index / 640 \quad (4.4)$$

exceto se o píxel tiver índice 0. Nesse caso, as variáveis foram diretamente igualadas a esse valor. Calculou-se depois a distância, separadamente em comprimento e em largura, do ponto de projeção ao ponto de colagem.

$$width\_diff = gluing\_point\_width - projection\_point\_width \quad (4.5)$$

$$length\_diff = gluing\_point\_length - projection\_point\_length \quad (4.6)$$

Por fim, procedeu-se à atualização dos ângulos. Em 1, encontra-se o algoritmo utilizado, no qual *inf\_servo* representa o servo inferior, recebido pelo tópico */pan\_current\_angle*, e *supp\_servo* representa o servo superior, recebido pelo tópico */tilt\_current\_angle*.

---

**Algorithm 1:** Algoritmo para atualização dos ângulos dos servos

---

**Data:** *inf\_servo* e *supp\_servo* atuais

**Result:** *inf\_servo* e *supp\_servo* atualizados

```

if  $abs(length\_diff) \leq abs(width\_diff)$  and  $3 < abs(width\_diff)$  then
  if  $30 \leq abs(width\_diff)$  then
    if  $width\_diff \leq 0$  then
       $inf\_servo \leftarrow inf\_servo + 3;$ 
    else
       $inf\_servo \leftarrow inf\_servo - 3;$ 
  else
    if  $width\_diff \leq 0$  then
       $inf\_servo \leftarrow inf\_servo + 1;$ 
    else
       $inf\_servo \leftarrow inf\_servo - 1;$ 
    if  $length\_diff < -5$  then
       $supp\_servo \leftarrow supp\_servo - 1;$ 
    else if  $length\_diff > 5$  then
       $supp\_servo \leftarrow supp\_servo + 1;$ 
  else if  $abs(width\_diff) < abs(length\_diff)$  and  $4 < abs(length\_diff)$  then
    if  $length\_diff \leq 0$  then
       $supp\_servo \leftarrow supp\_servo - 1;$ 
    else
       $supp\_servo \leftarrow supp\_servo + 1;$ 

```

---

Primeiro, verifica-se se a diferença em comprimento dos pontos é superior à distância em largura. Caso seja, e se a distância em comprimento for superior a 30, considera-se que os pontos estão bastante afastados, pelo que o servo inferior é incrementado ou decrementado em 3 graus, dependendo se a diferença em 4.5 for negativa ou positiva, respetivamente. Por outro lado, se a diferença for inferior a 30, o ângulo atual é variado em 1 grau, seguindo a mesma lógica. Caso a condição inicial não se verifique, ou seja, se a distância em largura for superior à distância em comprimento, o servo superior é incrementado ou decrementado, em uma unidade, dependendo se a diferença em 4.6 for positiva ou negativa, respetivamente. Foi também definido um valor para cada ângulo, de forma a que caso a câmara não consiga captar a projeção do laser, este seja direcionado para uma posição no tronco visível do ponto de vista da câmara.

Atendendo ao facto de que os servos não se encontram perfeitamente alinhados com a câmara, a rotação do servo superior provoca também uma variação horizontal do ponto de projeção, e o mesmo se verifica em relação ao servo inferior. Além disso, como o apontador laser não está adjacente à câmara, a sua projeção pode dirigir-se para uma zona que não seja captada pela câmara. Caso esta situação ocorra regularmente, o sistema pode tornar-se lento, ou até ficar preso num ciclo infinito. Para evitar isto, é feita uma pequena compensação vertical da projeção: se o módulo da diferença em 4.5 for superior ao módulo da diferença em 4.6, e caso a diferença vertical calculada em 4.6 seja superior a 5 ou inferior a -5, o ângulo do servo superior é incrementado ou decrementado em uma unidade, respetivamente. Após a atualização dos ângulos, os novos valores foram separadamente publicados nos tópicos */pan\_new\_angle* e */tilt\_new\_angle*.

Caso a diferença, em comprimento, dos pontos seja inferior ou igual a 3, e a diferença em largura inferior ou igual a 4, considera-se que os pontos coincidem e, portanto, não é efetuada nenhuma variação nos ângulos dos servos até que o operador cole a peça respetiva. Verificou-se que, quando a projeção atravessa uma peça ao dirigir-se para o ponto de colagem, esta não passa no filtro binário. O mesmo se verifica no momento de colagem de uma peça. Como mencionado anteriormente, quando isto acontece, o projetor laser é automaticamente redirecionado para uma posição pré-estipulada do tronco. No entanto, não é desejável que isto aconteça sempre que uma peça é colada. Para resolver este problema, cada vez que a câmara deteta a projeção, o valor do ponto de projeção é guardado num vetor, que armazena os últimos cinco valores recebidos. Caso num dado instante a câmara não consiga detetar a projeção, o vetor é percorrido e o valor mais recente diferente de 0 é usado para a comparação com o ponto de colagem. Apenas se todos os valores do vetor forem iguais a 0 é que o ponteiro é redirecionado.

- *arduino\_node*

Este nó lê, através da função *servo.read()*, uma estimativa dos valores atuais do ângulo de ambos os servos, e publica nos tópicos */pan\_current\_angle* e */tilt\_current\_angle* estes valores, tópicos esses subscritos pelo nó *servos\_node*. Paralelamente, subscrive também

os tópicos */pan\_new\_angle* e */tilt\_new\_angle*, nos quais as mensagens contendo os novos valores dos ângulos são publicadas, e envia-os para os servos com a função *servo.write()*, provocando a variação angular física correspondente.

## 4.5 Sumário

Neste capítulo, a implementação do estudo realizado foi dividida nas suas partes principais, e cada uma destas foi descrita e explicada. Inicialmente, foi feita uma lista de palavras-chave, de forma a definir alguns termos que foram utilizados ao longo da descrição. Depois, foi estruturada a arquitetura da implementação, bem como uma breve descrição das funcionalidades de cada nó e a forma como estes comunicam. Depois, a implementação propriamente dita foi descrita, desde o processamento da nuvem de pontos captada pelo sensor até o controlo dos servos de forma a obter o correto funcionamento do atuador.



## Capítulo 5

# Resultados

Neste capítulo serão apresentados e comentados os resultados obtidos para avaliação da implementação.

### 5.1 Processamento da nuvem de pontos da câmara

A câmara escolhida para o teste da implementação feita, a primeira das duas versões da Microsoft Kinect, possui uma resolução bastante limitada quando comparada com outras do mesmo tipo. A tabela 5.1 compara as especificações da câmara de profundidade utilizada com as da versão 2 da mesma linha e da câmara Intel Realsense D435 [38][39].

Tabela 5.1: Especificações de diferentes câmaras de profundidade

|                | <b>Microsoft Kinect V1</b> | <b>Microsoft Kinect V2</b> | <b>Intel Realsense D435</b> |
|----------------|----------------------------|----------------------------|-----------------------------|
| Tipo de sensor | Structured light           | Time of flight             | Active IR Stereo            |
| Resolução      | 640x480                    | 1820x1040                  | 1280x720                    |
| Campo de visão | 57°x43°                    | 70°x60°                    | 86°x57°                     |
| Profundidade   | 0.4m-4m                    | 0.5m-4.5m                  | 0.11m-10m                   |

Como mencionado no capítulo 4, para obter a orientação do tronco de colagem foi utilizado o algoritmo Iterative Closest Point (ICP) para identificar a bacia na nuvem captada pela câmara. Foi também utilizado Random Sample Consensus (RANSAC) para o cálculo do eixo que define essa mesma orientação, bem como para a identificação e distinção dos dois planos de colagem. Na tabela 5.2 encontram-se as constantes definidas na configuração dos métodos ICP e RANSAC:

Tabela 5.2: Configuração dos algoritmos de registo e segmentação utilizados

| Objetivo (Algoritmo utilizado)   | Funções                                   | Constantes |
|----------------------------------|---|------------|
| Registo da bacia na nuvem (ICP)  | <i>setMaxCorrespondenceDistance</i>       | 0.0025     |
|                                  | <i>setRANSACOutlierRejectionThreshold</i> | 1          |
|                                  | <i>setTransformationEpsilon</i>           | $1e^{-8}$  |
|                                  | <i>setMaximumIterations</i>               | 50         |
| Segmentação do cilindro (RANSAC) | <i>setNormalDistanceWeight</i>            | 0.1        |
|                                  | <i>setMaxIterations</i>                   | 1000       |
|                                  | <i>setDistanceThreshold</i>               | 0.1        |
|                                  | <i>setRadiusLimits</i>                    | (0, 0.1)   |
| Segmentação do plano (RANSAC)    | <i>setMaxIterations</i>                   | 1000       |
|                                  | <i>setDistanceThreshold</i>               | 0.0025     |

A resolução limitada da câmara fez-se notar no cálculo da distância de cada ponto à câmara. Idealmente, as coordenadas de todos os pontos pertencentes a cada plano de colagem seriam solução da equação matemática do plano geométrico respetivo. Na configuração do algoritmo RANSAC, para segmentação dos planos, foi definida uma tolerância de 0.0025m para a aceitação ou rejeição de um ponto como fazendo parte do plano, como apresentado na tabela 5.2. Esta constante deve ser cuidadosamente escolhida pois um valor demasiado baixo pode resultar na rejeição de muitos pontos que na verdade pertencem ao plano, e um valor demasiado elevado pode resultar na aceitação de pontos que na verdade não pertencem ao plano. Transpondo estas propriedades para a implementação feita, uma constante demasiado pequena poderia provocar ausência de pontos em redor de pontos de colagem, pelo que a estratégia de detetar se uma peça foi já colada nessa região teria de ser alterada; e uma constante demasiado elevada poderia culminar num cálculo errado das dimensões do plano, introduzindo algum erro na localização dos pontos de colagem.

Ao aplicar-se o algoritmo RANSAC para a deteção de planos, no tronco de cera, observou-se que esta deteção não se mantinha estável. Primeiro variou-se o valor da tolerância, mas foi visível que mesmo para valores inferiores a 0.0025m o algoritmo, para além de considerar como *outliers* pontos que pertencem ao plano, ainda considerava como *inliers* pontos que não pertencem ao plano. Posto isto, caso esta constante fosse ainda mais reduzida, o primeiro erro seria ainda mais notável, e caso a mesma fosse aumentada, o segundo erro agravar-se-ia. Foi então cortada uma folha A3 com as dimensões dos planos de colagem, e foi testado novamente o algoritmo, tendo-se obtido uma segmentação consideravelmente mais estável. Extraiu-se uma amostra de 60 contagens do número de pontos da nuvem resultante do RANSAC para segmentação do plano de colagem, quer na superfície em papel, quer na superfície de cera. Os resultados obtidos encontram-se nos histogramas 5.1 e 5.2.

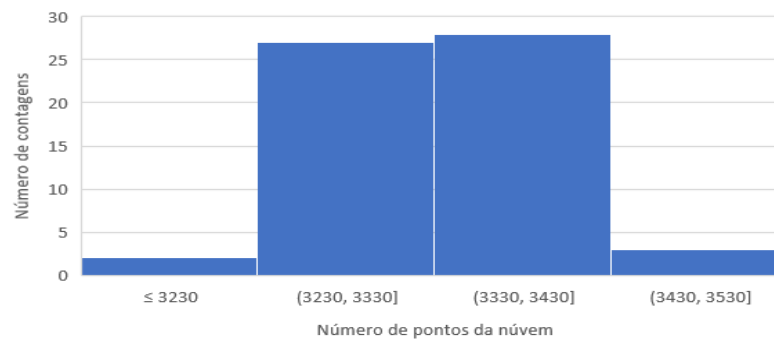


Figura 5.1: Número de pontos da nuvem do plano de colagem, com papel.

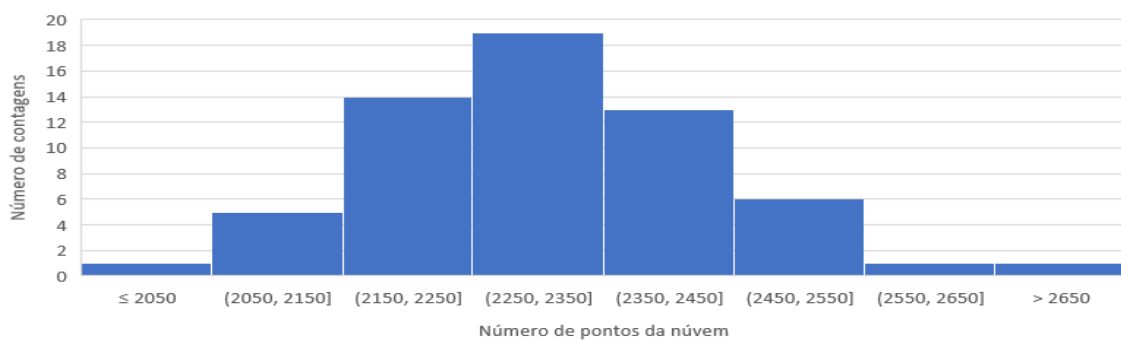


Figura 5.2: Número de pontos da nuvem do plano de colagem, sem papel (superfície de cera).

Nos histogramas acima apresentados é notória a diferença das amplitudes de valores registrada: no caso da amostra em que foi utilizado papel, 92% das contagens extraídas encontra-se no intervalo [3230,3430], ou seja, numa amplitude de 200 pontos. Por outro lado, quando o algoritmo foi aplicado diretamente na superfície em cera, a gama de valores foi mais larga, estando 93% das contagens no intervalo [2050,2550], e portanto numa amplitude 2,5 vezes superior à do papel. É de destacar também a diferença do número de pontos detetado em ambos os casos: o número máximo de pontos em 5.2 é inferior ao número mínimo de pontos em 5.1. Devido à instabilidade da deteção do plano, toda a implementação foi feita usando a folha de papel a envolver os planos do tronco.

Para estudar o efeito do posicionamento da câmara em relação ao tronco na qualidade e consistência da deteção do plano de colagem, variou-se o ângulo do vetor normal deste plano em relação ao eixo Z do sistema de coordenadas do mundo entre três valores: 0 graus (figura 5.3), ou seja, paralelo ao eixo; 25 graus, tendo-se rodado o tronco segundo o seu eixo no sentido da câmara (figura 5.4a); e 25 graus, tendo-se rodado o tronco segundo o seu eixo no sentido oposto à câmara (figura 5.4b). Extraíu-se uma amostra de 100 medições para cada uma das orientações mencionadas.

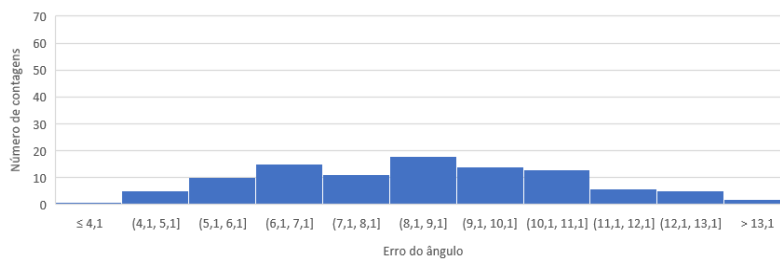
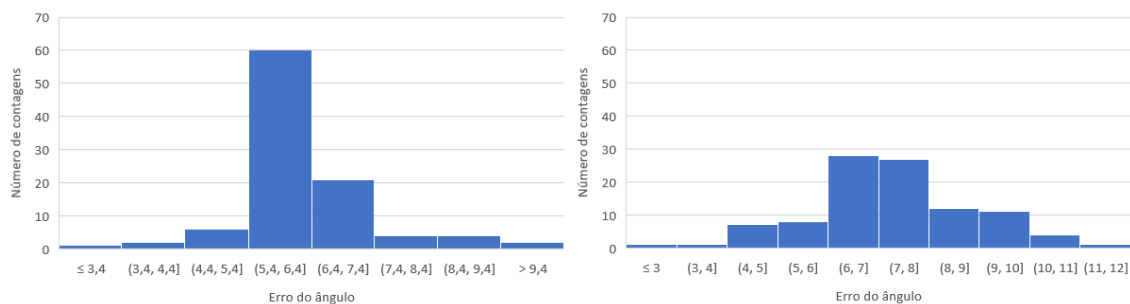


Figura 5.3: Erro angular do vetor normal do plano, a 0 graus.



(a) Erro angular do vetor normal do plano, a 25 graus no sentido da câmara. (b) Erro angular do vetor normal do plano, a 25 graus no sentido oposto à câmara.

Figura 5.4: Erro angular do vetor normal do plano, com este inclinado a 25 graus nos dois sentidos

Como esperado, observa-se que quando o eixo normal do plano de colagem se aproxima do eixo Z do sistema de coordenadas da câmara (histograma 5.4a), o cálculo do primeiro contém erros consideravelmente mais constantes do que nos restantes casos, sendo que 80% dos erros medidos está no intervalo [5,4,7,4] graus. O facto de não se ter verificado nenhuma medição inferior a 3 graus pode ser justificado pela existência de erro no cálculo do eixo do cilindro que foi utilizado como eixo Z do sistema de coordenadas do mundo, bem como pela rotação do tronco, que foi feita manualmente. Foi também interessante observar que, no histograma 5.3, o número de medições com erro compreendido no intervalo [5,1,11,1] graus manteve-se aproximadamente constante, entre 10 a 20 medições por unidade de erro dentro deste intervalo. Por outro lado, no histograma 5.4b, apesar de se ter observado uma gama de valores idêntica ao registado em 5.3, cerca de 50% das medições encontram-se no intervalo [6,8].

Avaliou-se também a evolução do erro relativo à localização dos pontos de colagem, à medida que vão sendo colocadas peças sobre este. Para isso, foi cortada uma folha de papel milimétrico, para que esta forme dois planos com as dimensões dos planos de colagem. Neste, assinalou-se o ponto exato de colagem da décima peça, e apontou-se o apontador laser para este ponto. A projecção foi então detetada e o respetivo ponto de projecção, no plano da imagem, foi publicado num tópico pelo nó *laser\_node*. O nó *processing\_node* subscreveu depois este tópico, e mediu a distância entre o ponto de projecção com o ponto de colagem localizado em 4.3.2. Primeiro, mediu-se esta distância sem peças no plano. Depois, foi adicionada uma peça de cada vez, começando

com 5 peças, até que o plano deixasse de ser detetado em todas as medições. Para cada caso, foram recolhidas 80 medições, e calculou-se a média e o desvio padrão das mesmas, bem como o número de vezes que o erro medido foi superior a 1,5 centímetros. Este último cálculo corresponde à falha na deteção do plano. Os resultados obtidos encontram-se na tabela 5.3.

Tabela 5.3: Erro na localização do ponto de colagem da última peça

| Nº de peças | Média    | Desvio padrão | Nº de medições > 1,5cm |
|-------------|----------|---------------|------------------------|
| 0           | 2,8403   | 1,1267        | 0                      |
| 5           | 3,1197   | 1,3166        | 0                      |
| 6           | 3,3717   | 1,4213        | 0                      |
| 7           | 3,7166   | 1,4060        | 0                      |
| 8           | 3,3899   | 3,2813        | 1                      |
| 9           | 4,2954   | 3,5152        | 1                      |
| 10          | 3,7514   | 1,4023        | 0                      |
| 11          | 3,8912   | 1,9790        | 1                      |
| 12          | 5,3492   | 9,5234        | 2                      |
| 13          | 10,4718  | 16,0205       | 19                     |
| 14          | 43,8282  | 45,9356       | 43                     |
| 15          | 478,1294 | 155,4991      | 80                     |

Verificou-se, portanto, que até 12 peças o erro aumentou pouco (cerca de 1 milímetro), tendo-se registado valores ligeiramente mais elevados nos casos em que ocorreu pelo menos uma falha na deteção do plano, sendo que nestas medições o valor de erro registado foi consideravelmente superior, afetando a média da amostra respetiva. A partir de 13 peças, o número de falhas na deteção do plano aumentou consideravelmente, levando à conclusão que o plano só consegue ser detetado caso este contenha 12 ou menos peças. Visto que na aplicação real são apenas coladas 10 peças no plano, esta experiência prova que o erro não se degradará durante a execução da colagem. Analisando a implementação, é de destacar que a localização dos pontos de colagem é feita através da medição das dimensões do plano. Para isso, é necessário que nenhuma das peças coladas oclua as arestas do plano de colagem. De facto, após colocar a 15ª peça no plano, uma das arestas deixou de ser captada pela câmara, pelo que as dimensões do plano foram erradamente calculadas em todas as medições.

## 5.2 Projeção do laser nos pontos pretendidos

Uma dificuldade encontrada nesta parte do projeto foi o facto de, tendo sido utilizados servos de baixa gama, por vezes a força mecânica destes ser inferior à inércia. Por outras palavras, verificou-se que quando o Arduino envia um impulso aos servos SG90 para estes rodarem 1 ou 2 graus, estes não fazem o movimento correspondente. Nestas situações, como quer o ponto de colagem quer o ponto de projeção do apontador laser não se alteram, o Arduino volta a enviar o mesmo impulso. Só quando a diferença entre o ângulo que o servo deveria ter, se rodasse sempre que um impulso lhe é enviado, e o ângulo real do servo é superior a 3 ou 4 graus é que o servo

efetua a rotação acumulada. Este problema é mais acentuado nas zonas em volta dos pontos de colagem, pois se a disparidade entre os pontos a ser comparados for de poucos píxeis, pretende-se que os servos rodem lentamente até que o ponto de projeção convirja no ponto de colagem pretendido. No entanto, com os servos utilizados, por vezes o apontador fica estático, os graus de rotação acumulam-se, e quando os servos finalmente rodam a projeção atravessa o ponto de colagem e o apontador roda mais um pouco, em sentido contrário ao pretendido.

Para avaliar o controlo dos servos e a eficácia global do sistema, foi usado o papel milimétrico cortado anteriormente. Neste papel, foram assinaladas as posições exatas onde se pretende que cada peça seja colada. O sistema foi posto em funcionamento durante 10 ciclos de colagem e, após o ponto de projeção convergir em cada ponto de colagem, foi medido o erro entre o ponto de projeção e o ponto de colagem assinalado, em píxeis e em milímetros. Na tabela 5.4 encontram-se as médias dos erros obtidos em cada peça, juntamente com a média global.

Tabela 5.4: Erro após convergência do ponto de projeção em cada ponto de colagem

| Peça   | Erro comp. (píxeis) | Erro largura (píxeis) | Erro (mm) |
|--------|---------------------|-----------------------|-----------|
| 1      | 1,4                 | 0,2                   | 5,567     |
| 2      | -0,2                | 1,6                   | 4,355     |
| 3      | 0,2                 | 0,7                   | 4,993     |
| 4      | 0,7                 | 1,3                   | 4,543     |
| 5      | 0,2                 | 1,3                   | 4,885     |
| 6      | -0,3                | 0,6                   | 2,037     |
| 7      | 0,6                 | 0,1                   | 3,163     |
| 8      | 0,5                 | -0,1                  | 3,094     |
| 9      | 0,7                 | 1,6                   | 5,033     |
| 10     | 0                   | 0,7                   | 4,035     |
| Global | 0,38                | 0,8                   | 4,171     |

Os erros medidos em píxeis foram calculados pelas fórmulas 4.5 e 4.6. Segundo estas, a diferença é positiva quando o valor de comprimento ou de largura do ponto de colagem, na imagem, é superior ao valor respetivo do ponto de projeção. Como as peças são coladas da extremidade do plano mais próxima à bacia até a extremidade oposta, e tendo em conta a orientação do tronco do ponto de vista da câmara, os pontos de colagem têm progressivamente menor comprimento. Por outro lado, como os servos não se encontram alinhados com o tronco, a rotação do servo inferior (responsável pela variação lateral do ponto de projeção) provoca também uma variação vertical do mesmo. Quando uma peça é colada, o servo inferior roda de forma a reduzir o comprimento do ponto de projeção, aumentando também ligeiramente o valor de largura do píxel correspondente. Posto isto, quando o ponto de projeção se aproxima de um ponto de colagem, o primeiro é detetado num ponto com maior comprimento e maior largura do que o ponto pretendido. Devido ao comportamento dos servos quando estes devem mover-se poucos graus, a projeção acaba por se mover mais um pouco do que o pretendido, justificando-se assim que quer o erro em comprimento quer o erro em largura apresentem valores médios positivos.

### 5.3 Comparação com Intel Realsense D435

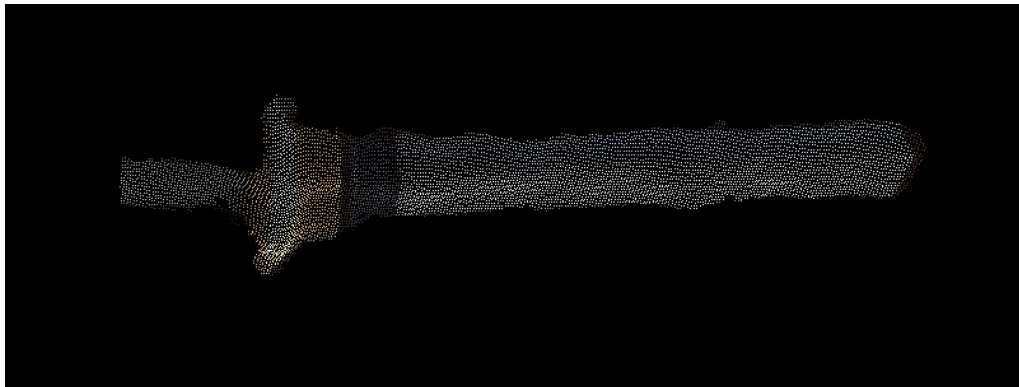
Com o tempo ainda disponível, foi também feita uma tentativa no sentido de comparar a performance global do sistema, usando como sensor uma câmara com melhor resolução do que a Microsoft Kinect V1. Entre as disponíveis, destacou-se a câmara RGB-D da empresa Intel, Realsense D435. Esta, tal como a Microsoft Kinect V1, possui também um driver já implementado para ser usada em conjunto com ROS: *realsense2*. Assim, basta também instalar este driver para ser possível o acesso aos diferentes tipos de imagem (RGB, de profundidade, e de infra-vermelhos) da câmara, através da subscrição ao tópico respetivo.

Era esperado que, devido à resolução 1280x720 da sua câmara de profundidade, a robustez e qualidade ao nível do pormenor (por exemplo, deteção de arestas, cantos, exatidão no cálculo da profundidade dos pontos, etc.) da nuvem de pontos fosse consideravelmente superior à da Kinect. No entanto, ao visualizar a nuvem no Software de visualização da Intel, *Realsense Viewer*, verificou-se que a câmara não captava vários pontos, além de que os planos apresentavam algumas ondulações e as arestas que separavam duas superfícies, em vez de formarem ângulos retângulos, possuíam forma curvilínea. Este comportamento pode justificar-se pelo facto de que esta câmara determina a distância de cada ponto através de cálculos, ao contrário da Kinect, cuja medição é feita através do tempo de voo de um sinal de luz artificial. Assim, a qualidade do cálculo da distância de cada ponto está muito dependente da qualidade da calibração quer da câmara, quer das constantes presentes nas fórmulas.

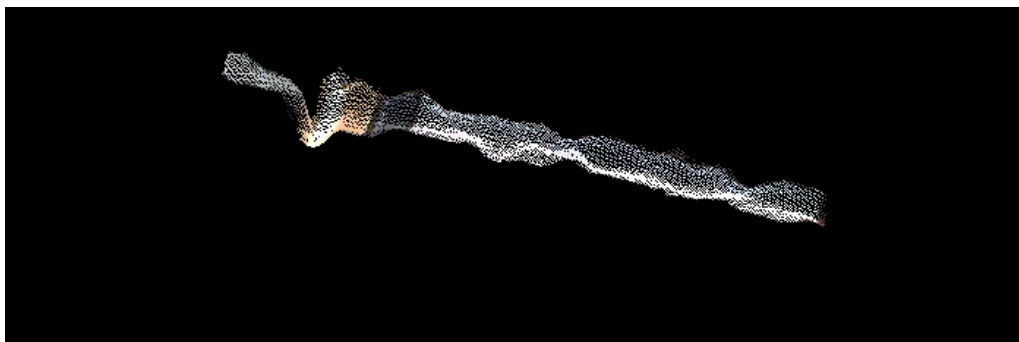
Procedeu-se, então, à tentativa de calibrar a Realsense D435. Primeiro, utilizou-se a ferramenta de auto calibração do *Realsense Viewer*. Com isto, deixou de haver buracos de pontos na nuvem, mas as ondulações permaneceram. Variou-se então manualmente os diversos parâmetros disponibilizados no Software, juntamente com as constantes de alguns algoritmos de pós-calibração (como, por exemplo, de média temporal, filtro espacial, redução da resolução, etc.). Testou-se, também, diferentes combinações destes algoritmos. Conseguiu-se uma ligeira redução nas ondulações, mas o resultado final não foi, mesmo assim, satisfatório. As figuras 5.5a, 5.5b e 5.5c mostram, em três perspectivas diferentes, a nuvem de pontos do tronco captada após a calibração.

Depois, testou-se o algoritmo de deteção do plano de colagem na nuvem do tronco com diferentes valores de tolerância. Verificou-se que, devido às ondulações, independentemente da tolerância definida o plano retornado pelo RANSAC não apresentava qualquer estabilidade, contendo pontos de ambos os planos, e não contabilizando pontos pertencentes ao plano que se pretendia detetar. O resultado obtido encontra-se na figura 5.6.

Para melhorar o resultado obtido, foi aplicado o método *Moving Least Squares* na nuvem do plano. Tendo-se obtido o resultado presente na figura 5.7. Nesta, a vermelho encontra-se o plano detetado pelo método RANSAC, e a azul os pontos retornados por MLS.



(a) Tronco de colagem, visto da câmara



(b) Tronco de colagem, visto na vertical



(c) Tronco de colagem, visto da altura deste

Figura 5.5: Tronco de colagem, visto de diferentes perspectivas

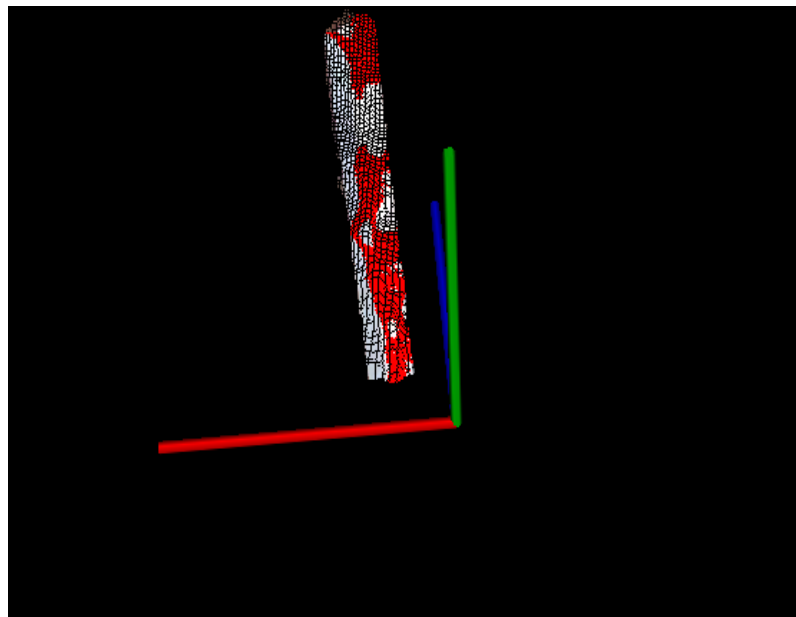


Figura 5.6: Detecção do plano de colagem, utilizando a câmara Realsense D435.

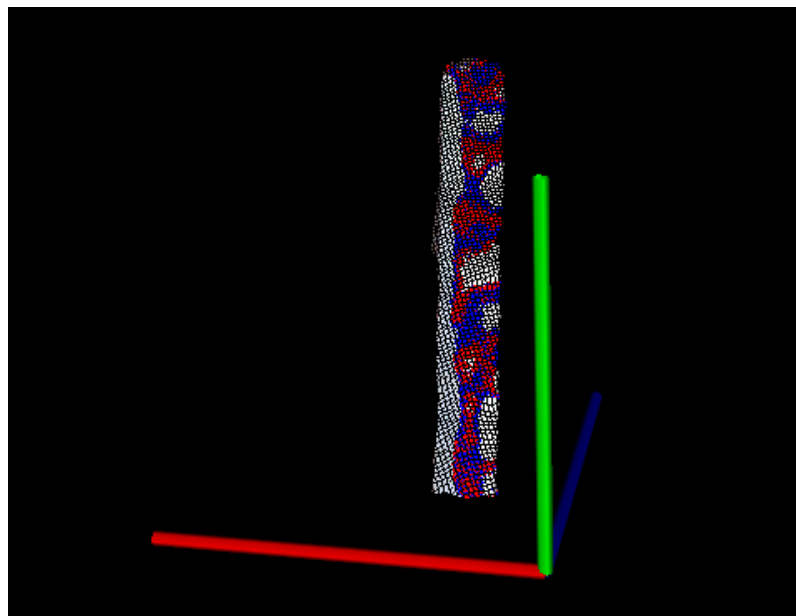


Figura 5.7: Detecção do plano de colagem, utilizando a câmara Realsense D435.

Para comparar, por fim, a qualidade na detecção do plano utilizando a câmara Microsoft Kinect V1 com a detecção do mesmo utilizando a Intel Realsense D435, mediu-se a consistência da localização dos 4 cantos do plano com ambas as câmaras. Para isso, colocou-se o plano de colagem em 4 posições diferentes, de forma a variar o ângulo entre o seu vetor normal e o eixo perpendicular ao plano da imagem. Os ângulos escolhidos foram  $0^\circ$ ,  $15^\circ$ ,  $30^\circ$  e  $40^\circ$ . Foram recolhidas 3000 medições da localização dos 4 cantos do plano, para cada posição deste, e foram calculadas a média, o desvio padrão e a variância das amostras. Os resultados obtidos encontram-se nas tabelas 5.5, 5.6, 5.7 e 5.8.

Na figura 5.8 encontram-se identificados os cantos do plano real, identificação essa feita para organizar a informação das tabelas mais abaixo, que contêm os resultados obtidos. Para cada caso de estudo, está também presente nas figuras 5.9a, 5.9b, 5.9c e 5.9d a inclinação efetuada, do ponto de vista do operador.

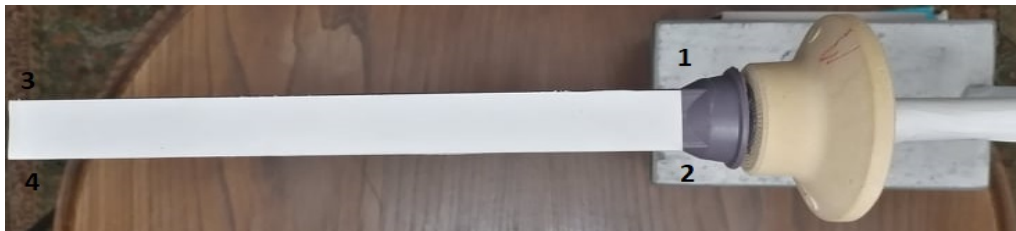
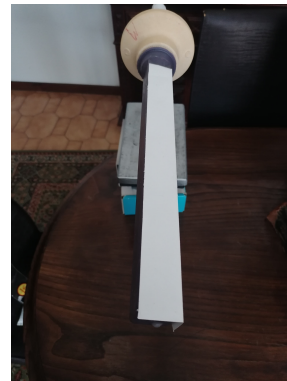


Figura 5.8: Enumeração dos cantos do plano de colagem.



(a) Plano de colagem, com normal a fazer ângulo de 0 graus.



(b) Plano de colagem, com normal a fazer ângulo de 15 graus.



(c) Plano de colagem, com normal a fazer ângulo de 30 graus.



(d) Plano de colagem, com normal a fazer ângulo de 40 graus.

Figura 5.9: Orientações do plano de colagem para medição dos cantos

- 0 graus:

Tabela 5.5: Comparação dos dois sensores na detecção dos cantos do plano de colagem, a 0 graus

| câmara         | Canto | Média (m)<br>(x,y,z)    | desvio padrão (mm)<br>(x,y,z) | variância ( $mm^2$ )<br>(x,y,z) |
|----------------|-------|-------------------------|-------------------------------|---------------------------------|
| Kinect V1      | 1     | (0.070, -0.062, 0.865)  | (1.794, 3.196, 1.375)         | (3.219, 10.219, 1.892)          |
|                | 2     | (0.072, -0.028, 0.865)  | (1.672, 1.391, 0.772)         | (2.797, 1.935, 0.595)           |
|                | 3     | (-0.238, -0.037, 0.869) | (2.432, 1.636, 1.165)         | (5.914, 2.677, 1.358)           |
|                | 4     | (-0.233, -0.005, 0.870) | (2.976, 2.439, 1.008)         | (8.855, 5.948, 1.016)           |
| Realsense D435 | 1     | (0.032, -0.008, 0.676)  | (4.745, 7.475, 1.863)         | (22.518, 55.875, 3.472)         |
|                | 2     | (0.038, 0.016, 0.675)   | (4.997, 6.891, 1.914)         | (24.969, 47.491, 3.664)         |
|                | 3     | (-0.272, 0.026, 0.726)  | (13.011, 9.139, 2.277)        | (169.293, 83.520, 5.184)        |
|                | 4     | (-0.266, 0.051, 0.727)  | (14.387, 8.915, 2.655)        | (206.987, 79.482, 7.048)        |

- 15 graus:

Tabela 5.6: Comparação dos dois sensores na detecção dos cantos do plano de colagem, a 15 graus

| câmara         | Canto | Média (m)<br>(x,y,z)    | desvio padrão (mm)<br>(x,y,z) | variância ( $mm^2$ )<br>(x,y,z) |
|----------------|-------|-------------------------|-------------------------------|---------------------------------|
| Kinect V1      | 1     | (0.067, -0.069, 0.870)  | (1.738, 3.663, 1.764)         | (3.021, 13.417, 3.112)          |
|                | 2     | (0.071, -0.038, 0.864)  | (1.652, 2.509, 2.468)         | (2.728, 6.293, 6.091)           |
|                | 3     | (-0.238, -0.040, 0.873) | (2.610, 4.606, 1.607)         | (6.812, 21.213, 2.584)          |
|                | 4     | (-0.235, -0.009, 0.868) | (2.547, 1.698, 2.114)         | (6.490, 2.884, 4.471)           |
| Realsense D435 | 1     | (0.034, -0.010, 0.680)  | (2.685, 6.159, 2.268)         | (7.207, 37.937, 5.142)          |
|                | 2     | (0.041, 0.018, 0.678)   | (1.890, 5.555, 3.172)         | (3.573, 30.858, 10.059)         |
|                | 3     | (-0.273, 0.027, 0.733)  | (8.431, 10.920, 3.095)        | (71.086, 119.256, 9.579)        |
|                | 4     | (-0.270, 0.057, 0.731)  | (8.238, 4.469, 3.353)         | (67.857, 19.968, 11.244)        |

- 30 graus:

Tabela 5.7: Comparação dos dois sensores na detecção dos cantos do plano de colagem, a 30 graus

| câmara         | Canto | Média (m)<br>(x,y,z)    | desvio padrão (mm)<br>(x,y,z) | variância ( $mm^2$ )<br>(x,y,z) |
|----------------|-------|-------------------------|-------------------------------|---------------------------------|
| Kinect V1      | 1     | (0.070, -0.072, 0.880)  | (2.080, 2.267, 2.217)         | (4.325, 5.139, 4.914)           |
|                | 2     | (0.071, -0.049, 0.860)  | (1.930, 4.060, 1.909)         | (3.725, 16.480, 3.646)          |
|                | 3     | (-0.235, -0.042, 0.880) | (14.314, 2.961, 1.697)        | (204.888, 8.767, 2.879)         |
|                | 4     | (-0.237, -0.022, 0.867) | (9.408, 3.206, 1.673)         | (88.518, 10.281, 2.798)         |
| Realsense D435 | 1     | (0.039, 0.001, 0.677)   | (2.748, 4.296, 1.794)         | (7.552, 18.451, 3.218)          |
|                | 2     | (0.043, 0.025, 0.681)   | (2.676, 5.408, 1.776)         | (7.164, 29.250, 3.153)          |
|                | 3     | (-0.279, 0.034, 0.733)  | (13.557, 4.946, 2.816)        | (183.800, 24.467, 7.928)        |
|                | 4     | (-0.270, 0.062, 0.737)  | (10.579, 8.138, 3.250)        | (111.925, 66.226, 10.565)       |

- 40 graus:

Tabela 5.8: Comparação dos dois sensores na detecção dos cantos do plano de colagem, a 40 graus

| câmara         | Canto | Média (m)<br>(x,y,z)    | desvio padrão (mm)<br>(x,y,z) | variância ( $mm^2$ )<br>(x,y,z) |
|----------------|-------|-------------------------|-------------------------------|---------------------------------|
| Kinect V1      | 1     | (0.092, -0.069, 0.878)  | (2.264, 2.408, 4.127)         | (5.127, 5.799, 17.033)          |
|                | 2     | (0.092, -0.056, 0.862)  | (1.957, 2.201, 1.478)         | (3.831, 4.845, 2.184)           |
|                | 3     | (-0.199, -0.051, 0.886) | (11.073, 10.626, 10.979)      | (122.628, 112.916, 120.535)     |
|                | 4     | (-0.208, -0.036, 0.871) | (8.456, 1.973, 10.658)        | (71.500, 3.894, 113.602)        |
| Realsense D435 | 1     | (0.140, 0.030, 0.674)   | (8.048, 9.873, 7.176)         | (64.777, 97.477, 51.496)        |
|                | 2     | (0.143, 0.054, 0.669)   | (7.741, 12.365, 3.898)        | (59.930, 152.883, 15.196)       |
|                | 3     | (-0.148, 0.074, 0.777)  | (18.152, 9.782, 7.940)        | (329.501, 95.690, 63.036)       |
|                | 4     | (-0.147, 0.095, 0.777)  | (19.533, 14.233, 8.538)       | (381.528, 202.583, 72.903)      |

Pelas tabelas, conclui-se que a Kinect apresenta resultados consideravelmente mais consistentes, quando em comparação com a Realsense D435. Observa-se também que as variâncias na detecção dos cantos mais afastados da bacia são maiores do que as variâncias na detecção dos cantos mais próximos. Com base nisto, seria necessário efetuar bastante mais processamento sobre a nuvem de pontos publicada pela câmara Realsense, até que o uso desta apresentasse melhores resultados do que os da Kinect V1.

## 5.4 Sumário

Neste capítulo, foram apresentados os principais resultados obtidos da implementação feita, utilizando os sensores Microsoft Kinect V1 e Intel Realsense D435.

Com o teste relativo à contagem do número de pontos retornado pelo RANSAC para detecção do plano de colagem, verificou-se que, usando a Kinect, seria necessário utilizar-se um papel branco a tapar os planos, devido á inconsistência da detecção registada sem este papel. Mediu-se também o efeito do aumento ou redução do ângulo formado entre o vetor normal ao plano de colagem e o eixo Z da câmara, para avaliar a importância do posicionamento do sensor em relação ao plano, no que diz respeito à correta e consistente identificação do plano de colagem. Avaliou-se também a degradação do erro na detecção do ponto de colagem da décima peça, tendo-se verificado que esta só é notável quando uma das arestas do plano fica oclusa, do ponto de vista da câmara. No que diz respeito ao atuador, mediu-se a distância entre o ponto de projeção e o ponto de colagem, após estes pontos convergirem, tendo-se verificado que, apesar da dificuldade de controlo dos servos mencionada, o algoritmo calculava sempre corretamente a distância relativa entre os pontos, enviando aos servos instruções no sentido de reduzir essa distância.

Por fim, abordou-se a tentativa de teste da implementação utilizando o sensor Intel Realsense D435, tendo-se observado que a calibração deste precisava de bastantes melhorias. Ainda assim, mediu-se a média, desvio padrão e variância da detecção dos 4 cantos do plano de colagem com ambos os sensores, de forma a comparar a performance dos mesmos.



## Capítulo 6

### Conclusões

A produção de peças em aço pelo método de fundição por cera perdida baseia-se num processo demorado, de cerca de 2 a 3 semanas. Na empresa Zollern e Comandita Portugal, algumas das etapas deste processo estão ainda pouco automatizadas, pelo que a qualidade do resultado final está consideravelmente dependente da performance e da experiência do operador. Uma destas tarefas é a de colagem de moldes, feitos em cera, num tronco, tarefa na qual o operador tem de colar um número fixo de peças num plano, sem qualquer apoio ou suporte. A rígida exigência na precisão desta colagem, aliada ao elevado período de aprendizagem requerido para que um operador seja considerado apto para a execução da mesma, justifica que seja desenvolvido um sistema que apoie os operadores durante a execução desta tarefa.

Já foi feita bastante investigação no desenvolvimento de sistemas de apoio ao operador, em diferentes tarefas de processos de manufatura, como corte e montagem, entre outros. As diferentes tecnologias estudadas no capítulo 2 permitiram agrupá-las em 3 grupos, de forma a relacionar as principais características e vantagens de cada um com as aplicações às quais melhor se adequa. Assim, foi possível comparar as características e exigências do processo de colagem com os sistemas já desenvolvidos, tendo-se chegado à conclusão que o mais vantajoso seria um que prestasse a assistência diretamente no plano de colagem, em tempo real. Apesar de se ter decidido, mais tarde, utilizar um apontador laser como atuador, foi ainda feito um estudo sobre diferentes técnicas de calibração de uma câmara e de um projetor. Este estudo foi útil no sentido em que, caso se pretenda substituir o apontador por um projetor, pode servir como referência na calibração do componente respetivo. Além disso, serviu também para explicitar a importância de considerar os erros não lineares das lentes, no processo de calibração, para aplicações nas quais a precisão da projeção seja crucial para a aplicabilidade do sistema. O estudo permitiu também perceber a vantagem de calibrar o projetor usando uma câmara descalibrada, em comparação a efetuar esta com uma câmara calibrada.

O trabalho desenvolvido consistiu no estudo de um sistema de apoio ao operador na tarefa de colagem, e respetiva implementação. No que diz respeito à câmara, o indicador mais claro que mostra que a Microsoft Kinect V1 não é a apropriada para o caso de estudo está presente nos gráficos 5.1 e 5.2, pois a inconsistência no número de pontos detetados como fazendo parte

do plano de colagem levou à necessidade de se utilizar uma folha de papel com as dimensões deste. Uma possível justificação para este comportamento é que a cera reflete os raios de luz de forma diferente do papel, pelo que a Microsoft Kinect tem maior dificuldade em receber a reflexão destes raios no caso da cera. Os resultados presentes na tabela 5.3 permitem concluir que o erro na localização do ponto de colagem da décima peça aumenta ligeiramente à medida que vão sendo coladas peças no plano: entre a primeira medição, com 0 peças no plano, até a medição com 10 peças, registou-se um aumento no erro de 0,9111 milímetros. Atualmente, os operadores da Zollern e Comandita Portugal medem a distância entre peças pela grossura do dedo utilizado para pegar na peça. Assim sendo, a implementação do sistema proposta permite uma acumulação do erro reduzida, à medida que as peças vão sendo coladas. Ainda assim, este erro poderia ser melhorado com uma câmara de melhor resolução. A limitação referente à resolução fez-se notar também nos resultados presentes nas tabelas 5.5, 5.6, 5.7 e 5.8, nas quais é notável a diferença entre os desvios padrão na localização dos cantos do plano de colagem mais próximos da bacia e os desvios padrão dos cantos opostos. Apesar de esta característica não ter comprometido o processamento da nuvem de pontos, nem a estabilidade do sistema, notou-se especial dificuldade na identificação do cilindro da bacia. Visto que este cilindro tem largura curta, a nuvem de pontos deste contém poucos pontos. Por isso, o cálculo do eixo do cilindro não era estável. Para contornar este problema, foi dada uma tolerância à filtragem feita de seguida, para isolamento do tronco. Com esta tolerância, caso o tronco não ficasse perfeitamente alinhado com o eixo das ordenadas, este não seria cortado pelo filtro. Para contornar a inconsistência na localização dos cantos do plano de colagem, o cálculo das coordenadas dos pontos de colagem foi feito medindo os valores máximo e mínimo, da nuvem do plano, quer em comprimento quer em largura. Assim, mesmo que os cantos não sejam detetados, desde que pelo menos um ponto de cada aresta real do plano seja detetada, as coordenadas dos pontos de colagem não se alterarão. Conclui-se, portanto, que apesar de a implementação apresentar um resultado global robusto, os erros registados poderiam ser reduzidos com um sensor com melhor resolução. Com a câmara Intel Realsense D435, não foi possível testar-se toda a implementação, devido à descalibração do sensor. Foram tentadas várias abordagens para melhorar a nuvem de pontos captada, tendo-se ainda comparado a deteção dos 4 cantos do plano de colagem entre esta câmara e a Microsoft Kinect V1. Nesta comparação, chegou-se à conclusão que a última câmara apresenta variâncias consideravelmente inferiores às registadas com a Realsense D435.

Relativamente ao apontador laser, os servos escolhidos para o controlo da projecção terão também de ser substituídos, principalmente devido ao problema da variação dos ângulos mencionado em 5.2. Devido à instabilidade do comportamento dos servos quando se pretende variar os ângulos em 1 ou 2 graus, apesar de o ponteiro convergir sempre para os pontos pretendidos, por vezes estes possuem alguma dificuldade em estabilizar no ponto de colagem pretendido. Para reduzir essa dificuldade, foi necessário dar-se uma tolerância em torno do ponto de colagem, no sentido de mesmo caso o ponto de projecção não coincida totalmente no ponto de colagem, o sistema considera que estes já coincidem. Isto resulta em erros superiores ao desejado, no intervalo [3.094, 5.567] milímetros, como registado na tabela 5.4.

O estudo efetuado culminou numa implementação compatível com qualquer câmara RGB-D. O mesmo cumpre os requisitos estipulados inicialmente, pois:

- Como os pontos são localizados através de fórmulas, e com base nas dimensões do plano de colagem, o erro entre o ponto de colagem e o ponto onde o operador efetivamente cola cada peça não está dependente das características físicas de cada operador;
- Sendo o Hardware constituído por uma câmara e por um apontador, o operador não usa qualquer dispositivo no seu corpo, pelo que a presença deste não causa cansaço adicional;
- O apontador projeta diretamente no plano de colagem. Assim, o operador não tem a necessidade de estar constantemente a alternar o olhar entre o plano e um ecrã ou tablet;
- Apesar de não se ter chegado a colocar o Hardware numa estrutura, de forma a que a sua presença não limite os movimentos do operador ou o incomode, a escolha deste teve em conta esse requisito, pelo que as reduzidas dimensões quer da câmara quer do apontador permitem que esta estrutura tenha também dimensões reduzidas.

Para o teste da implementação, foram utilizados dois sensores, sendo que entre elas obteve-se resultados consideravelmente melhores utilizando a Microsoft Kinect V1. Apesar disso, mesmo esta não pode ser aplicada na fábrica, devido aos maus resultados obtidos na deteção do plano de colagem, sem o uso da folha de papel.

## 6.1 Trabalho Futuro

O estudo conduzido nesta dissertação, podendo constituir um ponto de partida para o desenvolvimento de um sistema de apoio ao operador a ser implementado na fábrica da empresa Zollern e Comandita Portugal, necessita ainda de melhorias até estar operacional para ser aplicado na fábrica real.

Durante o desenvolvimento da dissertação, não foi possível testar o sistema em contexto real, com operadores da fábrica. Posto isto, a primeira proposta de trabalho futuro é precisamente testar o sistema nesse contexto.

Adicionalmente, e como mencionado acima, o Hardware necessita de ser substituído por outro com melhores especificações. Uma câmara com melhor resolução eliminaria a necessidade do uso da folha de papel, que por si só já impossibilita a usabilidade da Microsoft Kinect V1 na fábrica real. Além disso, a sua substituição permitiria também uma deteção mais consistente da secção cilíndrica da bacia, bem como dos contornos do plano de colagem. A substituição dos servos permitiria também um controlo mais preciso do apontador laser, acelerando a transição do ponto de projeção de um ponto de colagem para o ponto seguinte.

Uma outra proposta para trabalho futuro é combinar várias câmaras RGB-D, colocadas em posições diferentes. Com isso, seria possível obter nuvens de pontos de diferentes orientações, obtendo também informação que resultaria na redução de oclusões.

Uma última proposta relativa ao Hardware é a de substituir o apontador laser por um projetor. Esta alteração permitiria um sistema consideravelmente mais rápido, visto que seria possível programar o projetor de forma a projetar de uma só vez sobre todos os pontos de colagem, em contraste com a projeção por ponteiro laser, no qual a projeção tem de ser progressivamente atualizada cada vez que uma peça é colada. Uma outra vantagem do uso do projetor seria a possibilidade de customizar os contornos a ser projetados. Assim, estes podem conter a forma da base de cada peça, dando uma indicação mais precisa da zona a colar cada peça.

No que diz respeito ao Software, uma melhoria importante seria o desenvolvimento de um algoritmo mais robusto de localização do tronco de colagem no volume de trabalho. Sendo que a câmara capta toda a mesa de trabalho, o algoritmo atual não suporta movimentações do tronco, pois se o operador mudar este de posição o ICP deixa de reconhecer a bacia. Com um algoritmo mais robusto, cada operador poderia ajustar o tronco, conforme o deixe mais confortável durante a execução da tarefa.

Visto que a empresa Zollern e Comandita Portugal produz um conjunto bastante variado de objetos em aço, esta utiliza também moldes em cera com diferentes formatos, sendo estas coladas em troncos apropriados para cada tipo de molde. O sistema proposto foca-se no tipo de moldes proposto pela empresa. Propõe-se, então, como trabalho futuro a expansão da solução para ser compatível com diferentes formatos do tronco.

# Bibliografia

- [1] C. Flavián, S. Ibáñez-Sánchez, and C. Orús, “The impact of virtual, augmented and mixed reality technologies on the customer experience,” *Journal of Business Research*, vol. 100, pp. 547–560, 2019. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0148296318305319>
- [2] “Grupo Zollern,” disponível em <https://www.zollern.com/br/empresa/zollern-mundialmente/zollern-porto.html>, Acedido pela última vez em 2020-01-05.
- [3] D. Gorecky, M. Khamis, and K. Mura, “Introduction and establishment of virtual training in the factory of the future,” *International Journal of Computer Integrated Manufacturing*, vol. 30, no. 1, pp. 182–190, 2017. [Online]. Available: <https://doi.org/10.1080/0951192X.2015.1067918>
- [4] J. Roldán Gómez, E. Crespo, A. Martín-Barrio, E. Peña-Tapia, and A. Barrientos, “A training system for Industry 4.0 operators in complex assemblies based on virtual reality and process mining,” *Robotics and Computer-Integrated Manufacturing*, vol. 59, pp. 305–316, 2019.
- [5] F. Loch, F. Quint, and I. Brishtel, “Comparing video and augmented reality assistance in manual assembly,” in *2016 12th International Conference on Intelligent Environments (IE)*, 2016, pp. 147–150.
- [6] T. Masood and J. Egger, “Adopting augmented reality in the age of industrial digitalisation,” *Computers in Industry*, vol. 115, p. 103112, 02 2020.
- [7] L. Rodriguez, F. Quint, D. Gorecky, D. Romero, and H. Siller, “Developing a mixed reality assistance system based on projection mapping technology for manual operations at assembly workstations,” vol. 75, 11 2015.
- [8] H. Alvarez, I. Lajas, A. Larrañaga, L. Amozarrain, and I. Barandiaran, “Augmented reality system to guide operators in the setup of die cutters,” *The International Journal of Advanced Manufacturing Technology*, vol. 103, 07 2019.
- [9] “What is projection mapping? - Projection Mapping Central,” disponível em <http://projection-mapping.org/what-is-projection-mapping/>, Acedido pela última vez em 2020-01-18.

- [10] “What is projection mapping? | Holofiction,” <https://www.holofiction.hr/en/company/news/what-is-projection-mapping.html>, Acedido pela última vez em 2020-01-10.
- [11] H. Anwar, I. Din, and K. Park, “Projector calibration for 3d scanning using virtual target images,” *International Journal of Precision Engineering and Manufacturing*, vol. 13, 01 2012.
- [12] D. Moreno and G. Taubin, “Simple, accurate, and robust projector-camera calibration,” in *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization Transmission*, 2012, pp. 464–471.
- [13] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.
- [14] I. Martynov, J.-K. Kamarainen, and L. Lensu, “Projector calibration by “inverse camera calibration”,” vol. 6688, 05 2011, pp. 536–544.
- [15] S. Audet and M. Okutomi, “A user-friendly method to geometrically calibrate projector-camera systems,” in *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2009, pp. 47–54.
- [16] Jiarui Liao and Lilong Cai, “A calibration method for uncoupling projector and camera of a structured light system,” in *2008 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, 2008, pp. 770–774.
- [17] “Augmented and Virtual Reality Solutions - EON Reality,” disponível em <https://eonreality.com>, Acedido pela última vez em 2020-01-05.
- [18] “(2) EON LIAA Virtual HoloPartner - YouTube,” disponível em <https://www.youtube.com/watch?v=TTNIQizyT6M>, Acedido pela última vez em 2020-01-05.
- [19] “AVEVA - Global Leader in Industrial Software,” disponível em <https://www.aveva.com/>, Acedido pela última vez em 2020-01-05.
- [20] “Operator Training - Advanced Simulation Software from AVEVA,” disponível em <https://sw.aveva.com/engineer-procure-construct/operator-training>, Acedido pela última vez em 2020-01-05.
- [21] “appsbow,” disponível em <http://appsbow.com/#1>, Acedido pela última vez em 2020-01-05.
- [22] “Virtual Training Environment for Manufacturing Industry | Appsbow,” disponível em <http://appsbow.com/virtual-training-environment/#5~>, Acedido pela última vez em 2020-01-05.
- [23] “Laser Projection, Laser Measurement and Quality Assurance | LAP,” disponível em <https://www.lap-laser.com/>, Acedido pela última vez em 2020-01-05.

- [24] “Laser Projection& Quality Inspection Systems | Virtek,” disponível em <https://virtektion.com/>, Acedido pela última vez em 2020-01-07.
- [25] “Laser-based worker guidance | LAP,” disponível em <https://www.lap-laser.com/industry/worker-guidance>, Acedido pela última vez em 2020-01-05.
- [26] “HOMAG | HOMAG,” disponível em <https://www.homag.com/pt/>, Acedido pela última vez em 2020-01-08.
- [27] “Sistema LED de assistência ao operador | HOMAG,” disponível em <https://www.homag.com/pt/produtos/inovacoes/sistema-led-de-assistencia-ao-operador-1>, Acedido pela última vez em 2020-01-07.
- [28] “Assembly Solutions GmbH | Assembly Assistance Systems | Assembly System Planning,” disponível em [https://www.assemblysolutions.de/index\\_en.html](https://www.assemblysolutions.de/index_en.html), Acedido pela última vez em 2020-01-07.
- [29] “Home | Global | Siemens,” disponível em <https://new.siemens.com/global/en.html>, Acedido pela última vez em 2020-01-10.
- [30] “Virtual reality and field operator training | Plant Engineering Software COMOS | Siemens,” disponível em <https://new.siemens.com/global/en/products/automation/industry-software/plant-engineering-software-comos/virtual-reality-training.html>, Acedido pela última vez em 2020-01-10.
- [31] “Rexroth | Bosch Rexroth. WE MOVE. YOU WIN.” disponível em <https://www.boschrexroth.com/en/xc/home/index>, Acedido pela última vez em 2020-01-10.
- [32] “ActiveAssist assistance system | Bosch Rexroth AG,” disponível em <https://www.boschrexroth.com/en/xc/products/product-groups/assembly-technology/news/activeassist-assistance-system/index>, Acedido pela última vez em 2020-01-10.
- [33] “Institute for Process Engineering and Packaging - Fraunhofer IVV,” disponível em <https://www.ivv.fraunhofer.de/en.html>, Acedido pela última vez em 2020-01-10.
- [34] “(2) Fraunhofer E3 – Assistance Systems for Production - YouTube,” disponível em <https://www.youtube.com/watch?v=6giSUMKJ6io>, Acedido pela última vez em 2020-01-10.
- [35] Y. Zhang, G. Gibson, R. Hay, R. Bowman, M. Padgett, and M. Edgar, “A fast 3d reconstruction system with a low-cost camera accessory,” *Scientific Reports*, vol. 5, p. 10909, 06 2015.
- [36] C. Tomasi, “3 d reconstruction with two calibrated cameras,” 2013.
- [37] “Kinect - Wikipedia,” disponível em <https://en.wikipedia.org/wiki/Kinect>, Acedido pela última vez em 2020-06-02.

- [38] A. A. Al-Naji, K. Gibson, S.-H. Lee, and J. Chahl, “Real time apnoea monitoring of children using the microsoft kinect sensor: A pilot study,” *Sensors*, vol. 17, p. 286, 02 2017.
- [39] “Depth Camera D435 – Intel® RealSense™ Depth and Tracking Cameras,” disponível em <https://www.intelrealsense.com/depth-camera-d435/> Acedido em.
- [40] R. Ford and C. Coulston, *Design for Electrical and Computer Engineers*, 1st ed. USA: McGraw-Hill, Inc., 2007.
- [41] S. Naseera, G. Rajini, B. Venkateswarlu, and J. pemeena priyadarsini, “A review on image processing applications in medical field,” *Research Journal of Pharmacy and Technology*, vol. 10, pp. 3456–3460, 10 2017.
- [42] R. Ciora and C. Simion, “Industrial applications of image processing,” *ACTA Universitatis Cibiniensis*, vol. 64, 12 2014.
- [43] F. Duchoň, P. Bučka, M. Dekanová, M. Dekan, P. Beňo, and M. Tolgyessy, “Image processing of motion for security applications,” *European Scientific Journal*, vol. 13, 09 2017.
- [44] “ROS.org | Core Components,” disponível em <https://www.ros.org/core-components/>, Acedido pela última vez em 2020-06-06.
- [45] “Point cloud - Wikipedia,” disponível em [https://en.wikipedia.org/wiki/Point\\_cloud](https://en.wikipedia.org/wiki/Point_cloud), Acedido pela última vez em 2020-06-06.
- [46] “Point Cloud Library | The Point Cloud Library (PCL) is a standalone, large scale, open project for 2D/3D image and point cloud processing.” disponível em <https://pointclouds.org/>, Acedido pela última vez em 2020-06-06.
- [47] “MeshLab,” disponível em <http://www.meshlab.net/>, Acedido pela última vez em 2020-06-07.
- [48] “OpenCV,” disponível em <https://opencv.org/>, Acedido pela última vez em 2020-06-08.
- [49] “Arduino - Wikipedia,” disponível em <https://en.wikipedia.org/wiki/Arduino>, Acedido pela última vez em 2020-06-08.
- [50] “What is Arduino? - Definition from Techopedia,” disponível em <https://www.techopedia.com/definition/27874/arduino>, Acedido pela última vez em 2020-06-08.
- [51] “Iterative closest point - Wikipedia,” disponível em [https://en.wikipedia.org/wiki/Iterative\\_closest\\_point?fbclid=IwAR3zsTroCF11AclGv50qImbTIk26o4yH4-7p8ZzwLzuVUuIEwZgf--MoEI0](https://en.wikipedia.org/wiki/Iterative_closest_point?fbclid=IwAR3zsTroCF11AclGv50qImbTIk26o4yH4-7p8ZzwLzuVUuIEwZgf--MoEI0), Acedido pela última vez em 2020-06-11.

- [52] “How to use Random Sample Consensus model — Point Cloud Library 1.11.0-dev documentation,” disponível em [http://pointclouds.org/documentation/tutorials/random\\_sample\\_consensus.html?fbclid=IwAR0T1vKMrGi3QUYo52xzxEqpZD2xosKpnUhwRnFdu8xKuCrDjYMUfvs9\\_g](http://pointclouds.org/documentation/tutorials/random_sample_consensus.html?fbclid=IwAR0T1vKMrGi3QUYo52xzxEqpZD2xosKpnUhwRnFdu8xKuCrDjYMUfvs9_g), Acedido pela última vez em 2020-06-11.
- [53] “Process Investment casting,” disponível em <https://www.zollern.com/br/fundicao-forjado/fundicao-por-cera-perdida/process-investment-casting.html>, Acedido pela última vez em 2020-01-05.