



**Frederico Emanuel
Marques Moreira**

**Sistema de Determinação da Evolução do Relevo
Costeiro**

**System for Determining the Evolution of the Coastal
Morphology**



Universidade de Aveiro
2022

**Frederico Emanuel
Marques Moreira**

**Sistema de Determinação da Evolução do Relevo
Costeiro**

**System for Determining the Evolution of the Coastal
Morphology**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Eletrónica e Telecomunicações, realizada sob a orientação científica do Doutor Telmo Reis Cunha, Professor associado do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro, e do Doutor Paulo Renato Enes Baganha Baptista, Investigador CESAM, Universidade de Aveiro.

o júri / the jury

presidente / president

Prof. Doutor José Manuel Matos Moreira
Professor Auxiliar da Universidade de Aveiro

vogais / examiners committee

Prof. Doutor Daniel Filipe Albuquerque
Professor Adjunto do Instituto Politécnico de Viseu - Escola Superior de Tecnologia

Prof. Doutor Telmo Reis Cunha
Professor Associado da Universidade de Aveiro

**agradecimentos /
acknowledgements**

Esta dissertação significa o fim de uns longos cinco anos repletos de coisas boas. Pretendo assim, referir alguns nomes que mais me foram importantes ao longo desta jornada. Começo por agradecer ao Professor Telmo Cunha, meu orientador de Dissertação, uma vez que sem o apoio e a constante presença dele, este trabalho não teria o mesmo resultado final. Agradeço, também, aos meus pais por estarem sempre ao meu lado, principalmente quando mais precisei, e aos meus sobrinhos por me proporcionarem momentos de felicidade que contribuíram para o meu bem-estar. Um obrigado especial aos meus amigos Bárbara, Joana, Hugo, Gustavo, Inês Rocha e Bruno Pereira por terem feito parte desta aventura. Foram a minha segunda família. Por último, mas não menos importante, quero agradecer ao Instituto de Telecomunicações pelos meios colocados à disposição deste trabalho.

Palavras Chave

morfologia, 3D, estereoscopia, distanciómetro laser.

Resumo

Nos últimos anos foram feitos vários estudos sobre o processo de erosão costeira, tendo-se notado a necessidade de monitorizar a evolução da morfologia de determinadas zonas – zonas de estudo – e apresentá-la ao longo do tempo. Para tal, foram estudados vários métodos de determinação da forma 3D das superfícies, todos eles demonstrando necessidade de um grande esforço, dedicação e tempo, por parte da pessoa que está envolvida nas observações. Pretendendo minimizar o problema anterior, surge esta dissertação que consiste na conceção de um sistema protótipo para determinação da morfologia – mapeamento 3D da superfície observada – de uma determinada zona, com base em técnicas de estereoscopia com imagem digital, auxiliada por medidas de um distanciómetro laser. Para tal, foi necessário integrar neste sistema uma plataforma de suporte mecânico com dois graus de orientação angular (controlados por motores de passo) e desenvolver um software que permita o controlo da orientação dessa plataforma. O objetivo final da dissertação é utilizar as imagens adquiridas pela câmara e os dados adquiridos pelo distanciómetro de forma síncrona, sendo estes processados por um algoritmo dedicado, desenvolvido para o efeito, que permite gerar o mapa 3D da superfície do terreno.

Keywords

morphology, 3D, stereoscopy, laser distance sensor.

Abstract

In recent years, several studies on coastal erosion processes have been performed. The need to follow the evolution of the morphology of certain areas - study areas - and to present it over time was noticed. For this purpose, various methods for determining the 3D shape of surfaces have been studied, all of which have demonstrated the need for great effort, dedication and time on the part of the person involved in the observations. In order to minimize the previous problem, this dissertation consists in designing a prototype system to determine the morphology - 3D mapping of the observed surface - of a given area, based on stereoscopy techniques with digital imaging, aided by laser measurements. For this purpose, the developed system needed to integrate a mechanical support platform with two degrees of angular orientation (controlled by stepper motors) with a dedicated software tool to control the orientation of this platform. The final goal of the dissertation is to use the images synchronously acquired by the camera and the data acquired by the laser distance sensor generate a 3D map of the terrain surface, produced by a processing algorithm designed for this purpose.

Conteúdo

Conteúdo	i
Lista de Figuras	iii
Lista de Tabelas	iv
Glossário	v
1 Introdução	1
1.1 Objetivos	1
1.2 Estrutura	2
2 Metodologias para determinação da morfologia de regiões costeiras	3
3 Arquitetura do sistema	9
3.1 Distanciômetro Laser	10
3.2 Motores de passo e Rodas dentadas	13
3.2.1 Driver do motor	14
3.2.2 Desenho das rodas dentadas	18
3.3 Suporte	21
3.4 Câmaras	22
3.5 Protótipo Final	22
4 Desenvolvimento da plataforma de software	25
4.1 Configurações	26
4.2 Modo Manual	30
4.3 Modo automático	33
4.4 Plot 3D	35
4.4.1 Detecção do ponteiro laser nas imagens capturadas pelas câmaras ((w_{1P}, h_{1P}) e (w_{2P}, h_{2P}))	36
4.4.2 Retas r1 e r2	37

5	Simulação e Resultados	39
6	Conclusão e Trabalho Futuro	45
	Appendices	47
A	Simulação e Resultados	47
	Referências	49

Lista de Figuras

2.1	Sistema de levantamento INSHORE instalado numa moto 4 [1].	4
2.2	Relação entre um ponto em 3D com um ponto de uma imagem 2D [5]	5
2.3	Grelha para aquisição de imagens [6]	5
2.4	Erro obtido no cálculo das coordenadas 3D na simulação efetuada ao método da estereoscopia.	7
2.5	Erro obtido no cálculo das coordenadas 3D com a junção da informação da distância real	8
3.1	Maquete idealizada inicialmente (1- Distanciómetro Laser, 2- Motores de passo, 3- rodas dentadas, 4- plataforma de suporte do distanciómetro laser).	10
3.2	Peças mecânicas já adquiridas previamente.	10
3.3	Distanciómetro laser OptoNCDT ILR 1191 [7].	11
3.4	Ligações do laser ao D-sub9 fêmea.	11
3.5	PCB construído para a ligação entre o laser e o computador.	12
3.6	Ligações TTL-232R.	13
3.7	Necessidade de 2 eixos de rotação do laser.	13
3.8	Resolução do passo do motor vs pontos por medir.	14
3.9	Motor de passo FIT0278 [8].	14
3.10	Driver do motor - SparkFun ProDriver [9]	15
3.11	Terminais de entrada (esquerda) e de saída (direita) do SparkFun ProDriver.	15
3.12	Modos de configuração do SparkFun ProDriver e consequentes resoluções.	17
3.13	Sistema de controlo dos motores de passo.	18
3.14	Duas rodas dentadas com o mesmo raio.	18
3.15	Duas rodas dentadas com diferentes raios (1:4)..	19
3.16	Projeto de duas rodas dentadas com diferentes raios.	19
3.17	Passos para o desenho de uma roda dentada.	20
3.18	Roda dentada final de raio R.	20
3.19	Rodas dentadas utilizadas no protótipo	21
3.20	Suporte construído	21
3.21	WebCam Logitech C922 Pro Stream 1080p [10].	22

3.22	Protótipo final, (1- Laser e circuito de comunicação entre o laser e o computador, 2- Motores de passo, 3- PIC32 e drivers do motor, 4- Suporte).	23
4.1	Página inicial da plataforma.	26
4.2	Página inicial após configurações feitas e câmaras ligadas.	30
4.3	Modo Manual.	31
4.4	Modo automático.	33
4.5	Exemplo prático do algoritmo usado no modo automático.	34
4.6	Imagens capturadas pela câmara esquerda e direita respetivamente.	35
4.7	Determinação das coordenadas de um ponto 3D Real.	36
4.8	Imagem da câmara direita após aplicar o filtro BGR2GRAY.	37
4.9	Imagem da câmara após aplicar um <i>Threshold</i> e obter uma imagem binária.	37
5.1	Imagem da câmara direita.	39
5.2	<i>setup</i> montado para o primeiro teste.	40
5.3	Pré-calibração das câmaras.	41
5.4	Erro das coordenadas em metros	42
5.5	Vista do sistema montado num anfiteatro, para a realização dos testes..	43
5.6	Levantamento da morfologia 3D do anfiteatro.	44
5.7	Planos 2D do anfiteatro.	44
A.1	Dados do primeiro teste.	47

Lista de Tabelas

3.1	Descrição dos pinos do SparkFun ProDriver.	16
4.1	Resoluções de passo disponíveis.	32

Glossário

GUI graphical user interface

RGB Red Green Blue

INSHORE INtegrated System for High Operational
REsolution

DEMS Digital Elevation Models

GPS Global Positioning System

CCD Charge-Coupled Device

Introdução

Os problemas da erosão, principalmente ao nível da costa, têm-se mostrado muito preocupantes ao longo dos últimos anos, pondo em risco terrenos, bens e populações em diferentes zonas do país.

Dada a sua importância, vários estudos sobre os processos de erosão costeira têm vindo a ser conduzidos nas últimas décadas, baseando-se na medição de diversos fatores. Um dos elementos cuja monitorização é, naturalmente, importante, é a evolução da morfologia que uma determinada zona de estudo apresenta ao longo do tempo.

Para tal são considerados vários métodos de determinação da forma 3D da superfície dessa zona de estudo, envolvendo estas observações um tempo e esforço significativo (uma vez que requerem, habitualmente, que um operador percorra (várias vezes por ano) a zona em questão com um equipamento específico que vai registando as posições 3D dos pontos por onde este vai passando). Um exemplo deste tipo de métodos é o sistema de levantamento INSHORE instalado numa moto 4 [1], onde o utilizador necessita então de percorrer todo o areal o que implica um gasto de tempo e de recursos considerável. Pretende-se assim otimizar e automatizar o processo de levantamento de pontos de uma superfície.

1.1 OBJETIVOS

Recentemente está a ser explorada uma nova técnica de monitorização das mudanças da morfologia de determinadas zonas costeiras, com recurso a fotografia digital de elevada resolução. Um par de câmaras de vídeo de elevada resolução foi instalado na Praia de Mira, captando imagens síncronas da mesma zona do areal, mas de pontos distintos.

Utilizando técnicas de estereoscopia, pretende-se, assim, extrair o relevo do areal fotografado a partir das imagens digitais, no entanto, devido à distância reduzida entre as duas câmaras, uma das dimensões determinadas pelo processamento estereoscópico das imagens não tem elevada precisão o que dificulta a obtenção dos pontos 3D necessários para a correta obtenção da forma 3D de uma superfície.

O principal objetivo desta dissertação é, assim, criar um protótipo funcional capaz de obter a morfologia de uma superfície sem que seja necessário alguém deslocar-se ao local e despende de tempo várias vezes ao ano a obter dados ponto a ponto, e em vez disso, conseguir obter os mesmos dados mas remotamente.

1.2 ESTRUTURA

Este capítulo em questão, retrata a Introdução , os principais objetivos e a estrutura desta dissertação.

No segundo capítulo são dados exemplos de metodologias que existem para fazer a morfologia da erosão costeira e é feita uma demonstração, por simulação, da falta de precisão em profundidade que se observa através de processamento estereoscópico de imagens.

O terceiro capítulo diz respeito à arquitetura do sistema, onde são detalhadas todas as peças mecânicas utilizadas para construir o protótipo final de modo a que fosse possível interagir com este á distância e ainda que o sistema fosse preciso.

O capítulo 4 apresenta a plataforma de software desenvolvida, capaz de controlar o protótipo de uma maneira intuitiva. Neste capítulo, o utilizador aprende a utilizar sem qualquer problema ou ambiguidade a plataforma, e também, a familiarizar-se com alguns algoritmos usados, caso pretenda aprimorar ou adicionar novos recursos e/ou funcionalidades à plataforma.

No capítulo 5 são apresentados os testes realizados com o protótipo e apresentados e analisados os respetivos resultados.

O capítulo 6, diz respeito à conclusão e ao trabalho futuro.

Metodologias para determinação da morfologia de regiões costeiras

O estudo das zonas costeiras tem interesse em termos científicos em relação à sua proteção e gestão, uma vez que as zonas costeiras possuem um papel importante no desenvolvimento da humanidade, tanto em termos sociais como em termos económicos, além de que são zonas que dispõem de ecossistemas com elevado valor ambiental, que podem ser definidas como a ligação entre a terra e mar, que estão sujeitas a diversos tipos de eventos extremos, podendo originar inundações, destruição de infra-estruturas (ocupação humana) e de dunas e erosão.

Assim, a monitorização das zonas costeiras é essencial para a observação e previsão dos processos atuais e futuros, especialmente nas frentes urbanas, uma vez que muitos países enfrentam problemas de segurança, sendo necessária proteção contra a erosão costeira [2]. A monitorização é assim, importante para prever e controlar o impacto de intervenções humanas, tais como dragagens, realimentação de praias e construções de obras de defesa.

Os estudos idealizados na zona costeira são, normalmente, baseados em medições de ondas, correntes, transporte de sedimentos e de alterações da morfologia sendo que neste trabalho vai ser dada importância unicamente a este último ponto, ou seja, à morfologia de regiões costeiras, onde foram já feitos estudos para a sua determinação.

Dependendo do objetivo de cada caso de estudo, várias técnicas e métodos de pesquisa estão disponíveis, podendo ser divididas em técnicas aéreas e terrestres. A capacidade de cada técnica em adquirir dados de um modo eficiente é importante para gerar uma representação morfológica do terreno confiável.

Um dos métodos terrestres conhecidos é o sistema INtegrated System for High Operational REsolution (INSHORE) [1], visível na figura 2.1, que tem como principais objetivos conseguir obter coordenadas 3D altamente precisas da superfície (na ordem de 0,02 a 0,03 m) e ser eficiente no levantamento de uma praia com uma extensão de vários quilómetros. Este sistema consegue determinar Digital Elevation Models (DEMS) essencialmente através de um sensor laser e de recetores Global Positioning System (GPS) de elevada precisão.

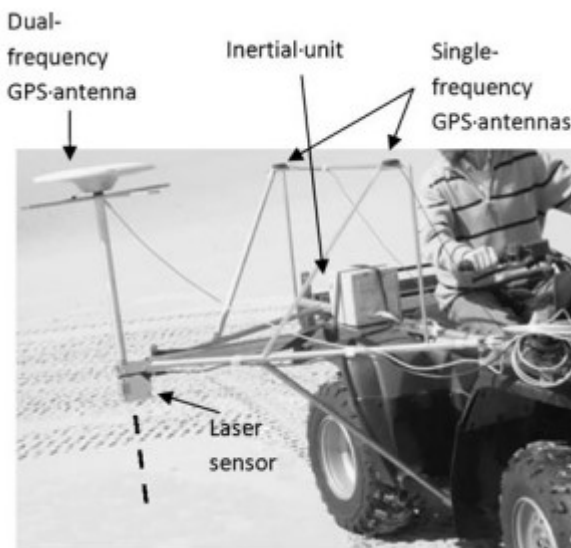


Figura 2.1: Sistema de levantamento INSHORE instalado numa moto 4 [1].

Este sistema demonstrou grande aplicabilidade com resultados satisfatórios mas que implicam a necessidade de um utilizador andar pela praia para poderem ser medidos pontos, o que implica gasto de tempo (e várias vezes ao ano), e de recursos.

Outras técnicas possíveis incluem o uso de plataformas aéreas e espaciais [3] que oferecem uma boa resolução espacial e temporal, contudo, o seu uso gera elevados custos. Para colmatar este problema, tem vindo a ser explorado nas últimas décadas sistemas com recurso a câmaras de vídeo, uma vez que é possível adquirir dados continuamente durante um intervalo de tempo tão longo quanto se queira, praticamente sem haver a necessidade de um operador se deslocar ao terreno para efetuar o levantamento de dados.

Em meados dos anos 90, foi na Oregon State University (EUA) que foram feitas as primeiras investigações sendo criado o programa de monitorização Argus [4] onde existem várias estações de monitorização de vídeo (constituídas por uma ou mais câmaras e um computador para armazenar os vídeos), que fornecem imagens costeiras de todo o mundo.

Como nem tudo é perfeito, as desvantagens destes sistemas dizem respeito à impossibilidade de obter dados durante a noite e principalmente ao problema da precisão das imagens decrescer com o aumento da distância entre a estação de vídeo e a praia, devido ao decréscimo da resolução obtida (cada pixel observa uma área que aumenta com a distância à câmara). No entanto, estes sistemas continuam a ser investigados e aprimorados cada vez mais.

Recentemente (em 2019), foi feito um estudo de obtenção da morfologia de uma praia usando relações matemáticas que correlacionam coordenadas do mundo real com coordenadas de uma imagem (plano com coordenadas e 2D) como mostra a figura 2.2 através de *surfcams* [5], que são câmaras de vídeo instaladas na costa com o objetivo principal de fornecer remotamente informações visuais da praia e do estado do mar para banhistas e surfistas transmitindo quase continuamente, através da Internet.

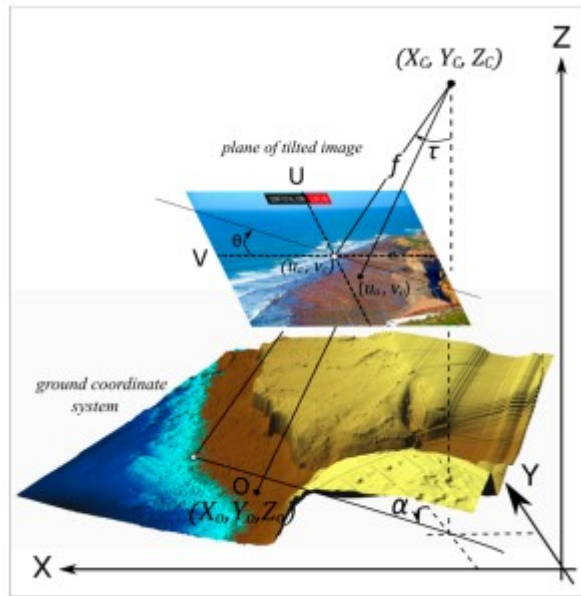


Figura 2.2: Relação entre um ponto em 3D com um ponto de uma imagem 2D [5]

Métodos baseados em fotografia digital referenciada (técnica que permite o estudo e a definição de formas, dimensões e posições de objetos no espaço, utilizando medições obtidas a partir de fotografias ou imagens digitais), para a reconstrução 3D têm-se tornado cada vez mais populares, devido ao baixo custo de implementação e à sua flexibilidade, tendo sido realizado um estudo em 2020 onde eram sobrepostas imagens bidimensionais (2D) de modo a reconstruir a geometria de uma forma 3D [6], com a ajuda de uma plataforma desenhada e desenvolvida com o objetivo de saber a posição onde a câmara estava a adquirir dados como mostra a figura 2.3.

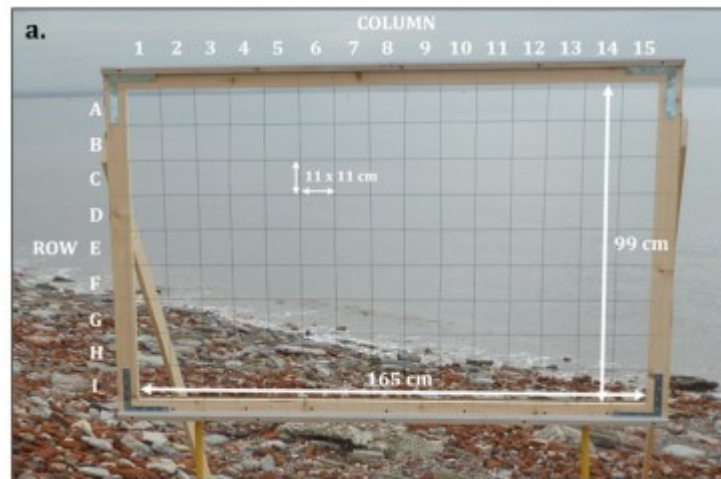


Figura 2.3: Grelha para aquisição de imagens [6]

No entanto, como se pode ver pelo tamanho da grelha, verifica-se que esta solução não é muito prática e que necessita de estar alguém presencialmente na colheita de dados,

conseguindo-se reconstruir apenas uma pequena escala da zona costeira.

Um outro método que pode ser utilizado para determinação da morfologia de regiões costeiras baseia-se na estereoscopia, que também consegue obter uma sensação de relevo dada pela fusão, numa única imagem, de duas fotografias do mesmo objeto tiradas de pontos de vista diferentes.

Para se analisar as potencialidades (e potenciais limitações) deste método foi feita uma demonstração, por simulação, da precisão em profundidade que se observa através de processamento estereoscópico de imagens. Foi implementado, através de um script em Matlab, um cenário virtual de duas câmaras digitais (com dimensões e características iguais às câmaras que posteriormente foram utilizadas neste trabalho) que são colocadas lado a lado no mesmo plano, distando horizontalmente de uma distância que foi especificada em 2 metros (por semelhança a um sistema de monitorização existente na Praia de Mira, para o qual este trabalho visa contribuir). Seguidamente simulou-se a aquisição de fotografias de um ponto cujas coordenadas (relativamente ao ponto situado a meio das câmaras) são de 20 metros para a direita (coordenada x_p), 300 metros em profundidade (coordenada y_p), e 30 metros na vertical (coordenada z_p), para baixo (i.e., as coordenadas reais do ponto visado são (20;300;-30)). Com base numa análise da geometria deste sistema virtual, simulou-se uma imprecisão na identificação exata do pixel na imagem de cada câmara, tendo a estas coordenadas-imagem sido adicionado ruído aleatório (com magnitude até à distância entre pixels adjacentes). Com estes valores efetuou-se o cálculo estimado para as coordenadas do ponto visado no terreno, tendo-se observado (como esperado) uma enorme imprecisão associada à estimativa da profundidade desse ponto. A figura 2.4 apresenta o erro associado às coordenadas estimadas por este processo.

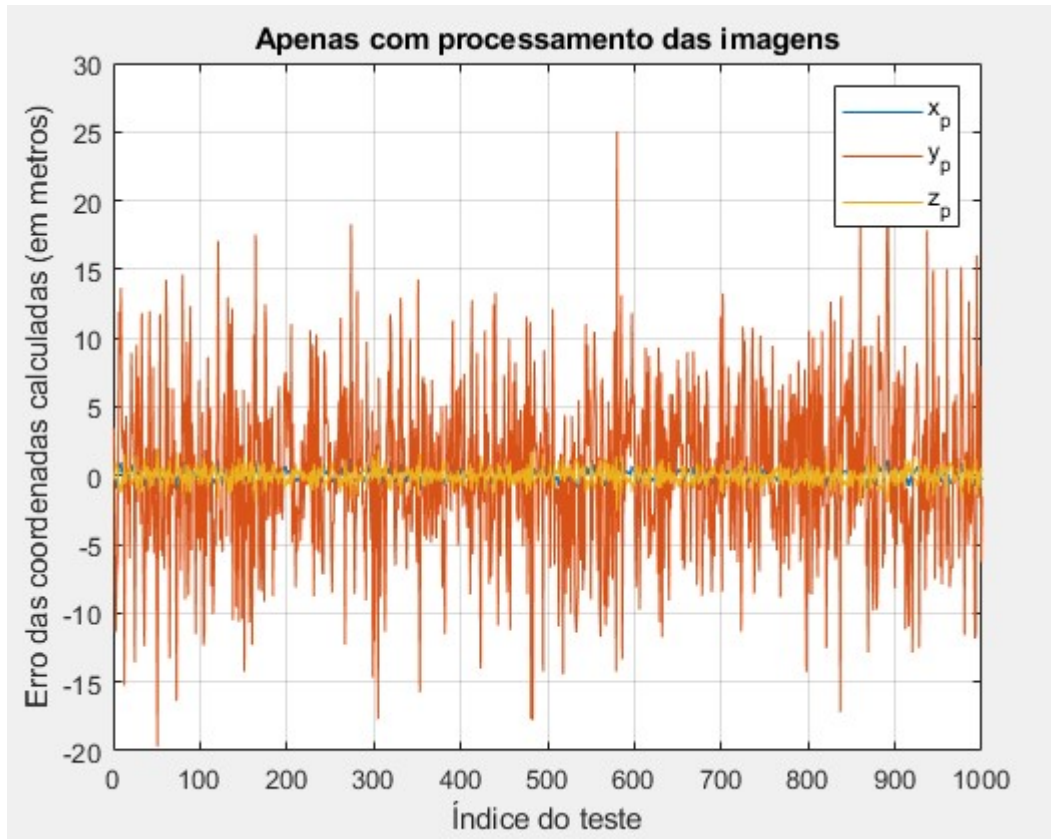


Figura 2.4: Erro obtido no cálculo das coordenadas 3D na simulação efetuada ao método da estereoscopia.

Pela análise desta figura podemos concluir a falta de precisão no cálculo da coordenada y_p por este método, ou seja, falta de precisão na profundidade. Em seguida, complementou-se esta simulação adicionando às equações geométricas que produzem o resultado por estereoscopia uma condição extra que assume que a medida da distância entre o ponto médio entre as câmaras e o ponto visado é conhecido. Com esta condição adicional (onde se considerou, também, um erro aleatório na medida dessa distância que seria típico de um distanciómetro laser de elevada precisão), calcularam-se novamente as coordenadas do ponto visado, tendo-se observado os erros indicados na figura 2.5.

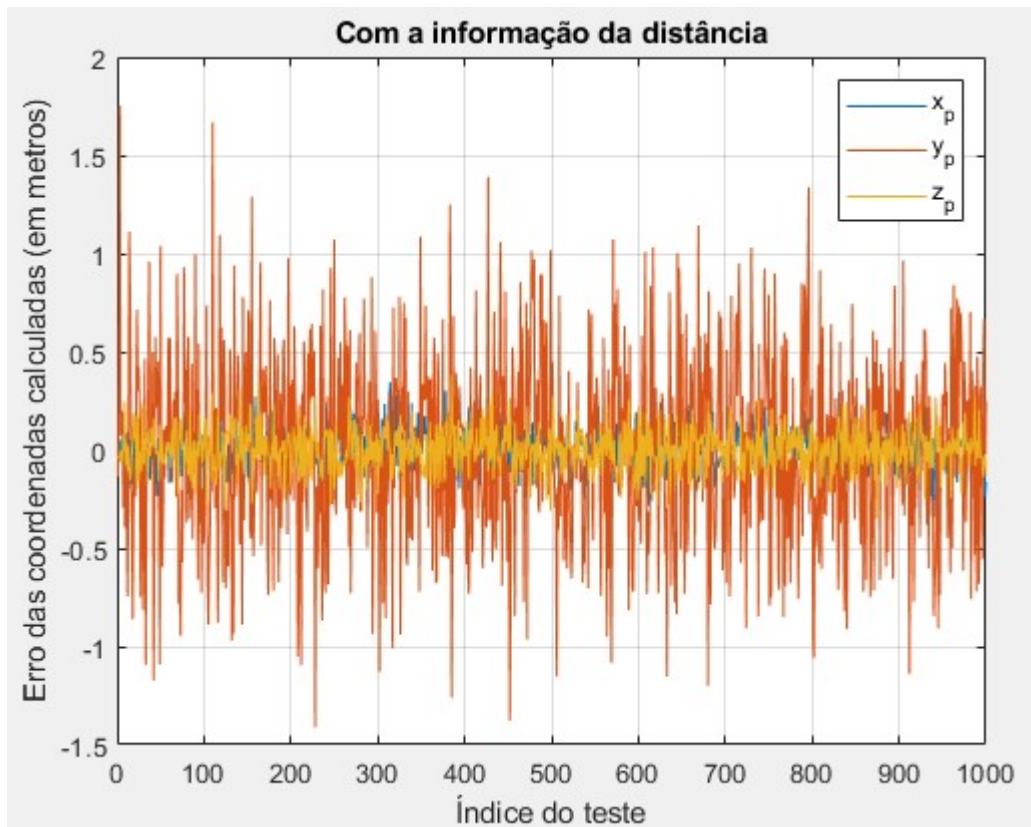


Figura 2.5: Erro obtido no cálculo das coordenadas 3D com a junção da informação da distância real

Por esta última figura observa-se que o erro na coordenada y_p é cerca de 10 vezes menor, comparado com os resultados da figura 2.4, e que o erro em x_p e z_p diminuiu também.

Arquitetura do sistema

O sistema dividiu-se em vários elementos, nomeadamente:

- A plataforma de posicionamento do distanciómetro laser necessária para o levantamento de uma grelha de pontos visados no solo, durante a noite (para garantir a visibilidade da mira do laser);
- A eletrónica de interface com o distanciómetro laser para existir comunicação entre o laser e o computador;
- A aplicação de software de controlo do posicionador e das câmaras de vídeo;
- O algoritmo de processamento que consegue a partir das coordenadas dos píxeis, em conjunto com a informação da distância do laser, obter um ponto de coordenadas reais em relação ao laser;
- A interface com o utilizador.

Neste capítulo é descrito o hardware (posicionador e eletrónica de interface com o distanciómetro laser), sendo que os elementos implementados em software são descritos no capítulo seguinte. São, assim, descritas todas as peças mecânicas necessárias para a construção do protótipo, tendo em conta alguns pré-requisitos como: i) ser possível interagir à distância; ii) o sistema ser possuidor de precisão suficiente.

Foi assim elaborada uma maquete, visível na figura 3.1, com base em material que já tinha sido adquirido anteriormente numa bolsa de investigação, como mostra a figura 3.2, tendo sido decidido reaproveitá-lo para a construção da plataforma de posicionamento do distanciómetro laser.

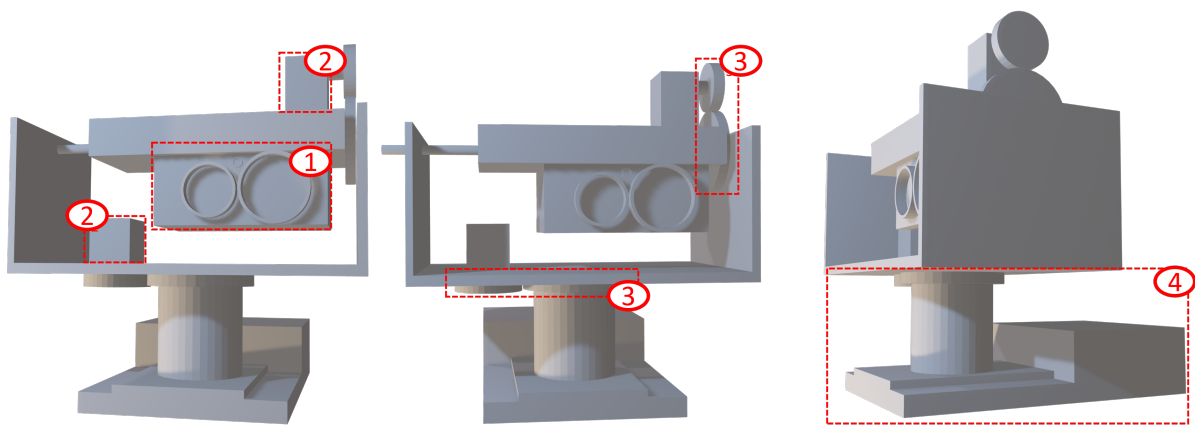


Figura 3.1: Maquete idealizada inicialmente (1- Distanciômetro Laser, 2- Motores de passo, 3- rodas dentadas, 4- plataforma de suporte do distanciômetro laser).

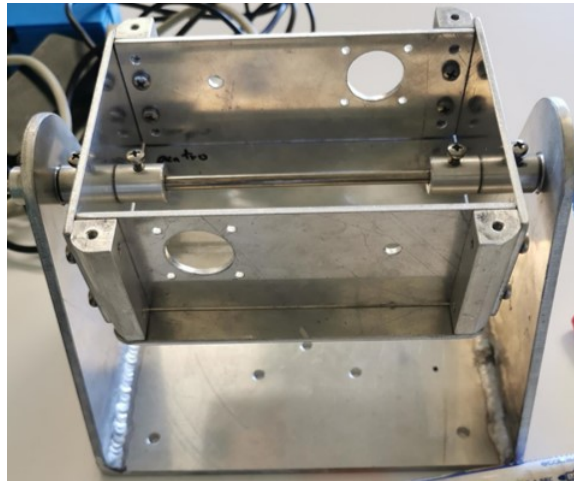


Figura 3.2: Peças mecânicas já adquiridas previamente.

3.1 DISTANCIÔMETRO LASER

Como já foi mencionado anteriormente, devido à distância reduzida entre as duas câmaras, uma das dimensões determinadas pelo processamento estereoscópico das imagens não tem elevada precisão, o que dificulta a obtenção dos pontos 3D necessários para a correta obtenção da forma 3D de uma superfície. Para resolver esta falta de precisão (no eixo do y) foi utilizado o distanciômetro laser **OptoNCDT ILR 1191** [7] como mostra a figura 3.3 capaz de alcançar até 300 metros se a superfície for refletora.



Figura 3.3: Distanciômetro laser OptoNCDT ILR 1191 [7].

Este laser vai ser necessário na medida em que, quando obtemos um par de fotografias digitais é possível identificar as coordenadas do ponteiro do laser, e saber através deste, a distância ao ponto a ser determinado. Com todos estes dados (distância obtida pelo laser e os pixels do ponteiro laser das 2 imagens digitais), é possível obter a precisão necessária em todas as dimensões.

Para uma correta conexão/ligação entre o laser e o computador foi necessário realizar um circuito de interface, uma vez que o laser possui uma interface RS232 assíncrona (configurado para uma ligação de 8 bits de dados, sem paridade e com 1 stop bit, usando uma cadência de 115.5 kBaud) com as ligações a uma ficha D-Sub 9 fêmea indicadas na figura 3.4 (de acordo com a informação indicada no datasheet do sensor laser) [7].

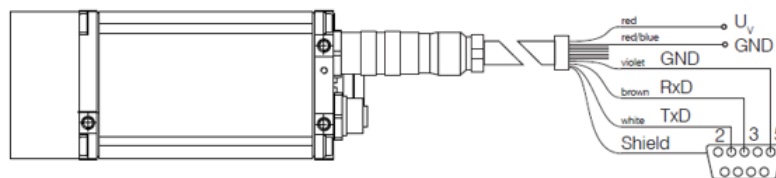


Figura 3.4: Ligações do laser ao D-sub9 fêmea.

A ideia é, assim, usar os pinos livres da ficha D (fêmea) para ligar os restantes condutores do cabo do laser (nomeadamente, a alimentação e sinais analógicos e digitais auxiliares, não usados neste trabalho), para que o sensor seja ligado ao restante equipamento por apenas um cabo. Contudo, isto faz com que a ligação deixe de seguir o estipulado no protocolo RS232, pelo que o cabo não poderá ser ligado a outros dispositivos que não o construído especificamente para o efeito.

Consideram-se assim as seguintes ligações:

Pino 1 (da ficha D fêmea a ligar ao cabo) – VCC 12V (red)

Pino 2 – TxD (white)

Pino 3 – RxD (brown)

Pino4 – Analog signal 4-20 mA (yellow) - A corrente que é injetada nesta linha é proporcional à distância medida dentro de um intervalo de distância programável;

Pino 5 – GND (violet and red/blue)

Pino 6 – N/C

Pino 7 – Q1 (blue) - Tem como função representar leituras de distância como dados de operação lógica;

Pino 8 – Q2 (grey/pink) - Tem como função representar leituras de distância como dados de operação lógica;

Pino 9 – Trigger (green) - Permite que uma única medição de distância seja acionada pela aplicação de um sinal externo.

Uma vez já feita a ligação do laser a uma ficha D (fêmea), foi montado o resto do circuito representado na figura 3.5 constituído essencialmente por um conector D (macho), um circuito integrado MAX3232, um conversor (TTL-232R) de USB para o protocolo série RS232, mas considerando as tensões ao nível do TTL, e não ao nível do especificado no protocolo RS232, um transformador 220 V AC para 12 V DC e um conversor DC-DC (LM1084) que converte a alimentação de 12 V para 3.3 V.

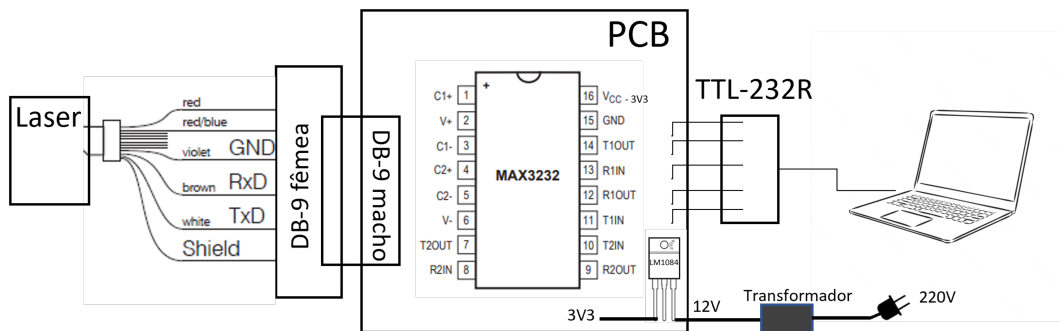


Figura 3.5: PCB construído para a ligação entre o laser e o computador.

O transformador para 12V é necessário porque é esta a alimentação do laser, no entanto é usado então um LM1084 que nos permite obter 3V3 a partir dos 12V porque é esta a alimentação do MAX3232.

O MAX3232 fornece a interface elétrica entre um controlador de comunicação assíncrona e o conector de porta série e o TTL-232R é um conversor série TTL para USB onde as ligações efetuadas são as apresentadas na figura 3.6

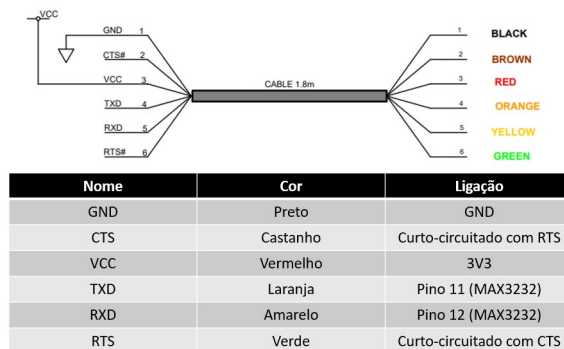


Figura 3.6: Ligações TTL-232R.

Estando tudo completamente funcional, é possível, através da porta série do computador, estabelecer ligação com o laser através de uma porta USB e enviar comandos através de um terminal que suporte ligações do tipo série, tal como o HyperTerminal ou equivalente, sendo o protocolo de comunicação baseado em curtas mensagens de texto.

3.2 MOTORES DE PASSO E RODAS DENTADAS

Sendo necessário controlar todo o sistema à distância, para obter todos os pontos da superfície, a plataforma de posicionamento do distanciômetro laser requer dois eixos de eixos de rotação que possam ser controlados por um micro-controlador, composto por motores e um sistema de rodas dentadas. O objetivo destes dois eixos é direcionar o feixe do distanciômetro laser por forma a que este observe diversos pontos ao longo da superfície da região costeira, tal como ilustrado na figura 3.7, para assim se determinar a morfologia dessa área observada.

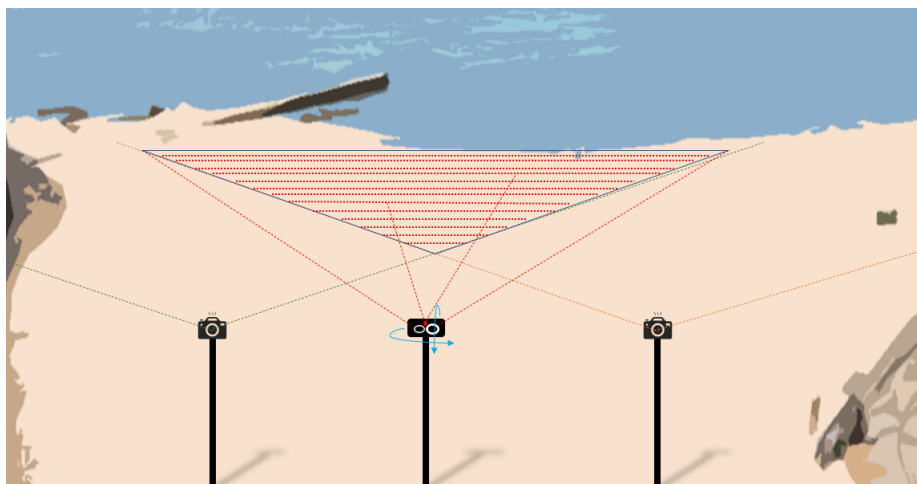


Figura 3.7: Necessidade de 2 eixos de rotação do laser.

Foi tido o cuidado de escolher motores com binário e resolução suficientemente elevada para que não se verificasse o problema visto na figura 3.8 uma vez que, se o passo não for muito pequeno, a resolução que se consegue obter no terreno, a longas distâncias, será insuficiente, tal como se pretende representar na figura 3.9 (a relação geométrica entre o distanciômetro laser e a superfície a observar impõe uma distância elevada entre os pontos visados (A e B)

para o passo angular indicado por α). Assim, quanto menor for o passo, maior será a quantidade de pontos que se consegue observar no terreno e mais informação irá ser obtida.

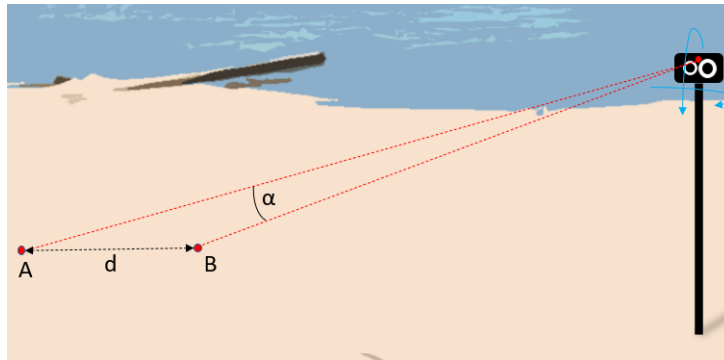


Figura 3.8: Resolução do passo do motor vs pontos por medir.

O motor de passo adquirido foi então o FIT0278 [8], como mostra a figura 3.9, que possui um passo de 1.8 graus, ou seja, em 360 graus, será possível realizar 200 passos.



Figura 3.9: Motor de passo FIT0278 [8].

No entanto, esta resolução ainda não é ideal para atingir os objetivos iniciais, e para colmatar este problema foram consideradas duas soluções:

- **1:** Usar um driver para os motores capaz de aumentar esta resolução, ou seja, aumentar o número de passos em 360°;
- **2:** Ganhar resolução de passo aquando do desenho das rodas dentadas a serem utilizadas.

3.2.1 Driver do motor

Para controlar os motores foi adquirido um SparkFun ProDriver [9], como mostra a figura 3.10, que é compatível tanto com os motores, como com o micro-controlador a utilizar que vai ser detalhado mais à frente neste documento.

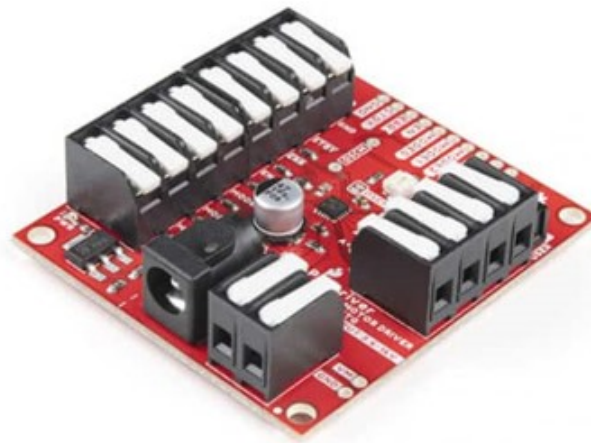


Figura 3.10: Driver do motor - SparkFun ProDriver [9] .

Este driver possui terminais de entrada visíveis do lado esquerdo da figura 3.11 que são descritos na tabela 3.1e os terminais de Output do lado direito, onde os terminais de entrada vão estar ligados ao micro-controlador e os terminais de saída aos motores.

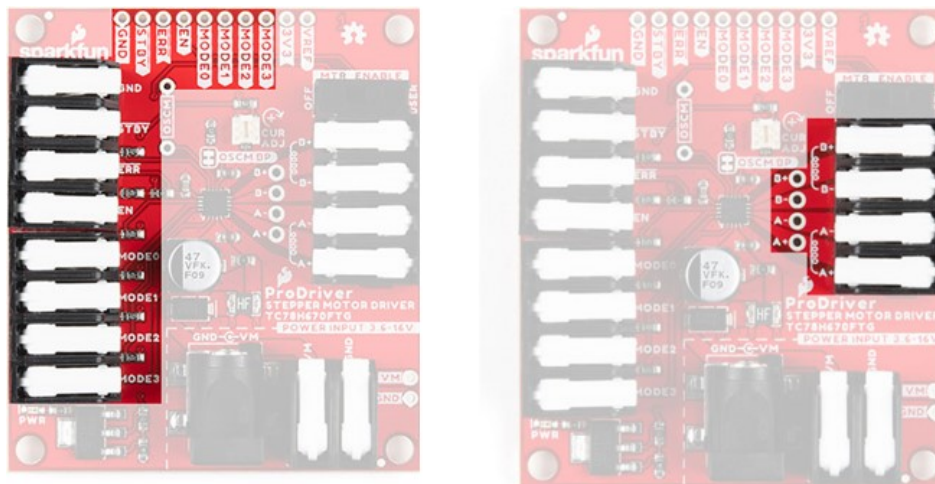


Figura 3.11: Terminais de entrada (esquerda) e de saída (direita) do SparkFun ProDriver.

Tabela 3.1: Descrição dos pinos do SparkFun ProDriver.

Nome do Terminal	Descrição	Possível configuração (V)
Ground (GND)	Referência 0V.	0V.
Standby (STBY)	Pino usado para iniciar um dos possíveis métodos de controlo: <ul style="list-style-type: none">• 0V → o motor está parado;• 3V3 → o driver fica configurado em modo “clock-in” ou em comunicação série, conforme a tensão nos pinos MODE0...MODE3.	0V a 3V3 <i>Default: 0V</i>
Enable (EN)	Ativa todo o sistema, só funciona quando está ativo.	0V a 3V3 <i>Default: 0V</i>
MODE0	Utilizado para a configuração do método a utilizar.	0V a 3V3 <i>Default: 3V3</i>
MODE1	Utilizado para a configuração do método a utilizar.	0V a 3V3 <i>Default: 3V3</i>
MODE2	Utilizado para a configuração do método a utilizar e para aplicar uma onda quadrada fazendo com que o motor se desloque passo a passo.	0V a 3V3 <i>Default: 3V3</i>
MODE3	MODE3 controla o sentido de rotação: <ul style="list-style-type: none">• 0V → Os motores vão rodar no sentido contrário aos ponteiros do relógio;• 3V3 → Os motores vão rodar no sentido dos ponteiros do relógio.	0V a 3V3 <i>Default: 3V3</i>

O driver possui dois modos distintos: O modo fixo, onde uma vez inicializado, a resolução é configurada e assegurada ao longo de toda a operação do motor; e o modo variável, onde o motor pode ser inicializado com uma certa configuração no início da operação dos motores, e mais tarde modificar a resolução através de uma nova configuração enquanto os motores estão a funcionar.

Estas configurações são conseguidas com recurso aos pinos MODE0 a MODE3 que podem estar no estado LOW (0V) ou HIGH (3V3) como mostra a seguinte figura 3.12.

MODE3 pin input	MODE2 pin input	MODE1 pin input	MODE0 pin input	Mode	Function
L	L	L	H	Variable Mode	Full step resolution <-> 1/2 step resolution (2-phase excitation) (1-2-phase excitation)
L	L	H	L		Full step resolution <-> 1/4 step resolution (2-phase excitation) (W1-2-phase excitation)
L	L	H	H		Full step resolution <-> 1/8 step resolution (2-phase excitation) (2W1-2-phase excitation)
L	H	L	L		Full step resolution <-> 1/16 step resolution (2-phase excitation) (4W1-2-phase excitation)
L	H	L	H		Full step resolution <-> 1/32 step resolution (2-phase excitation) (8W1-2-phase excitation)
L	H	H	L		Full step resolution <-> 1/64 step resolution (2-phase excitation) (16W1-2-phase excitation)
L	H	H	H		Full step resolution <-> 1/128 step resolution (2-phase excitation) (32W1-2-phase excitation)
H	L	L	L	Fixed Mode	Full step resolution (2-phase excitation)
H	L	L	H		1/2 step resolution (1-2-phase excitation)
H	L	H	L		1/4 step resolution (W1-2-phase excitation)
H	L	H	H		1/8 step resolution (2W1-2-phase excitation)
H	H	L	L		1/16 step resolution (4W1-2-phase excitation)
H	H	L	H		1/32 step resolution (8W1-2-phase excitation)
H	H	H	L		1/64 step resolution (16W1-2-phase excitation)
H	H	H	H		1/128 step resolution (32W1-2-phase excitation)

Figura 3.12: Modos de configuração do SparkFun ProDriver e consequentes resoluções.

De notar ainda que é através do pino MODE2 que se consegue fazer com que o motor se desloque passo a passo através da aplicação de uma onda quadrada, com a ajuda do micro-controlador, e o pino MODE3 controla o sentido de rotação, ou seja, quando este se encontra a LOW, os motores vão rodar no sentido contrário aos ponteiros do relógio, e, se estiver a HIGH, irá rodar no sentido dos ponteiros do relógio.

As ligações efetuadas no sistema de controlo dos motores de passo encontram-se apresentadas na figura 3.13.

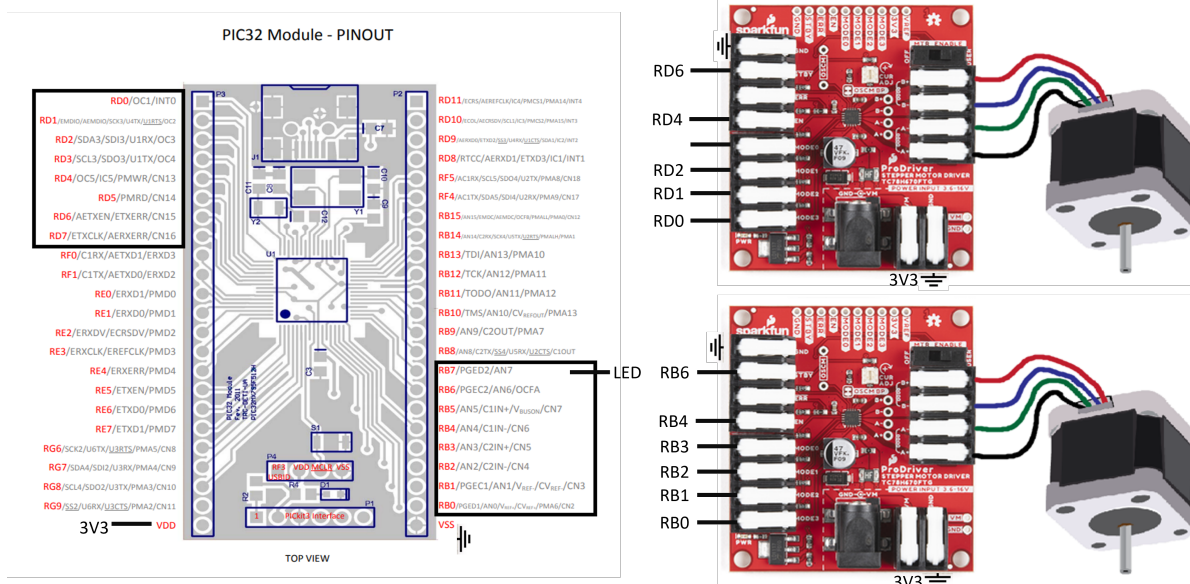


Figura 3.13: Sistema de controlo dos motores de passo.

O micro-controlador considerado na figura 3.13 é o PIC-32 (tendo sido utilizada uma placa existente previamente, que implementa as funcionalidades básicas de suporte ao funcionamento deste micro-controlador) e foram utilizados os pinos RB0 a RB6 para controlar um dos motores, o pino RB7 que está ligado apenas a um LED para verificar se o PIC-32 e os comandos utilizados estão a funcionar corretamente, e os pinos RD0 a RD7 foram usados para controlar o outro motor.

3.2.2 Desenho das rodas dentadas

Para se conseguir movimentar os eixos de rotação é necessário que uma das rodas dentadas esteja fixa a um eixo da plataforma, e a outra esteja fixa apenas ao motor. Se as rodas dentadas forem iguais como na figura 3.14, ou seja, a proporção for de 1:1, não existe ganho de resolução uma vez que a roda dentada móvel percorre uma única vez a roda dentada fixa, então o passo mantém-se.

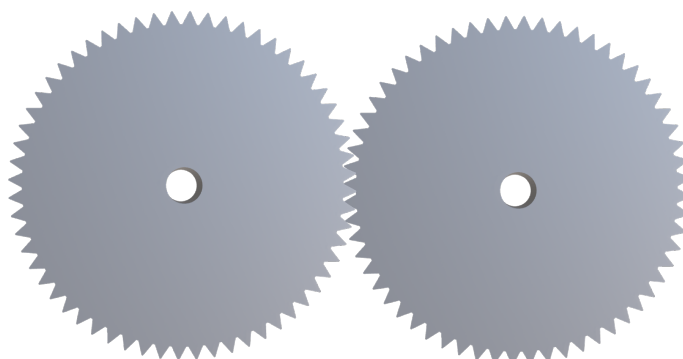


Figura 3.14: Duas rodas dentadas com o mesmo raio.

Caso o raio entre as duas rodas dentadas não seja o mesmo, e seja por exemplo o raio da roda móvel igual a 1/4 da roda dentada fixa, como mostra a figura 3.14, então agora a

resolução do passo aumentou 4 vezes uma vez que a roda pequena consegue dar 4 voltas completas enquanto a roldana maior percorre uma volta. A resolução do passo passa assim a ser $1/4$ da inicial.

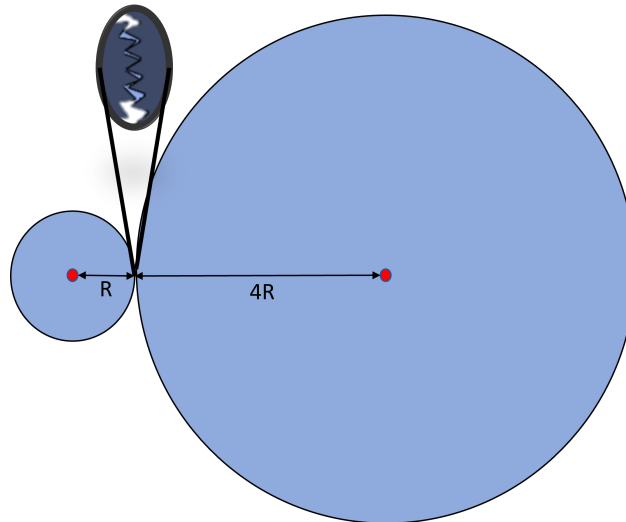


Figura 3.15: Duas rodas dentadas com diferentes raios (1:4).

Neste protótipo, uma vez que os eixos onde as rodas dentadas vão ser inseridas estão a uma distância específica e se quisermos uma determinada proporção entre estas, as rodas dentadas têm também de ser desenhadas de modo a obter o contacto adequado entre estas como mostra a figura 3.16

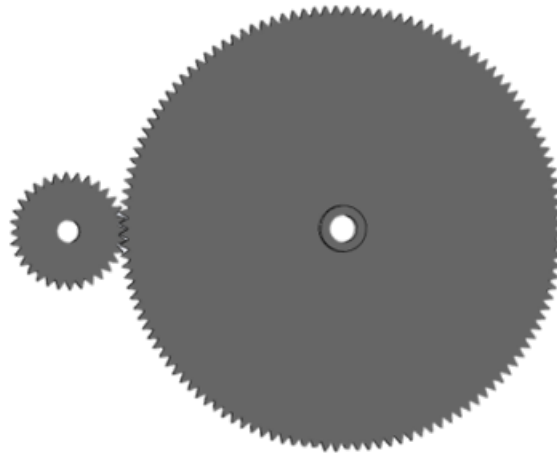


Figura 3.16: Projeto de duas rodas dentadas com diferentes raios.

Para que passemos da figura 3.15 para a figura 3.16, uma série de passos devem ser seguidos no desenho das rodas dentadas como mostra a próxima figura 3.17.

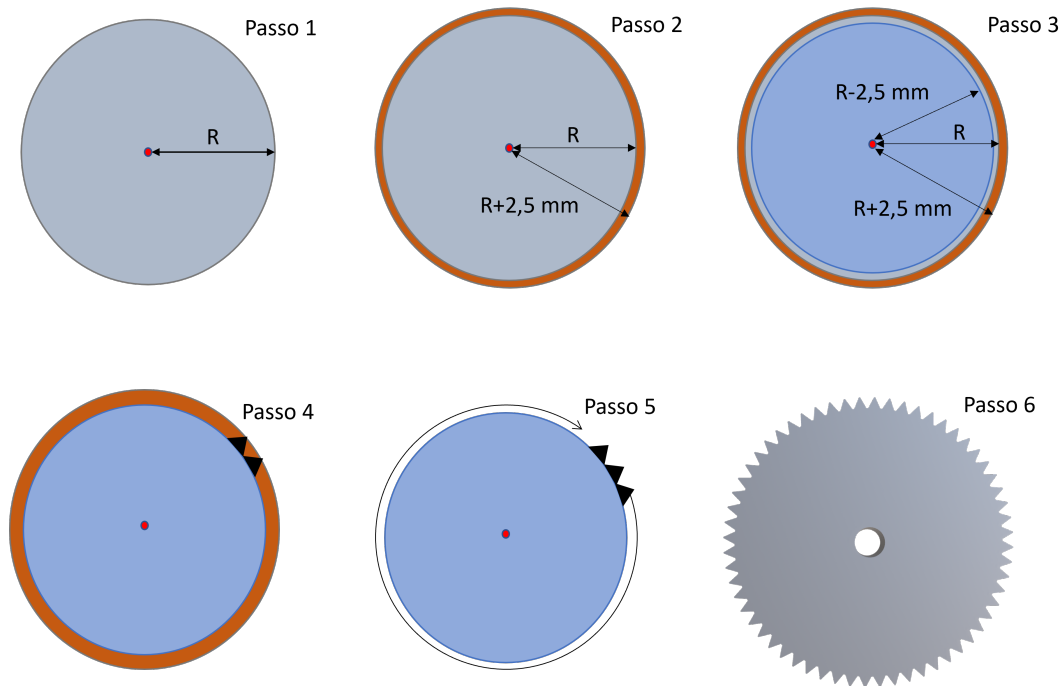


Figura 3.17: Passos para o desenho de uma roda dentada.

Primeiramente, deve ser conhecida a distância total entre os eixos onde as rodas dentadas vão ser inseridas, ou seja, entre os centros. No caso da figura 3.15, seria R mais $4R$ obtendo-se assim uma distancia total para dimensionar de $5R$. Imaginando que se quer dimensionar 2 rodas com uma proporção de 1:4, uma das rodas irá ter raio R e a outra $4R$ tal como mostra esta mesma figura. Para desenhar esta roda menor, cria-se um círculo de raio R , seguidamente um círculo de raio R mais $2,5$ mm e depois no passo 3, um círculo de raio R menos $2,5$ mm.

No passo 4 é eliminado o círculo central (de raio R), e vão sendo desenhados dentes até completar a roda dentada com um número de dentes par como mostra o passo 5 até se obter o resultado no passo 6. Nesta etapa basta apenas abrir o círculo central onde a roda dentada vai ser aplicada e atribuir uma certa espessura, alcançando o resultado da figura 3.18.

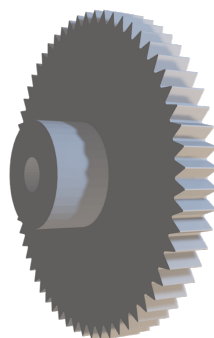


Figura 3.18: Roda dentada final de raio R .

Para os dois eixos independentes foram realizadas dois pares de rodas dentadas visíveis na figura 3.19, uma meia roda dentada com uma proporção de 1:2 para o eixo horizontal, e um par de rodas dentadas para o eixo vertical com uma proporção de 1:4 uma vez que já foi visto

que este eixo precisa de cuidado acrescido uma vez que é muito sensível e necessita de maior resolução.

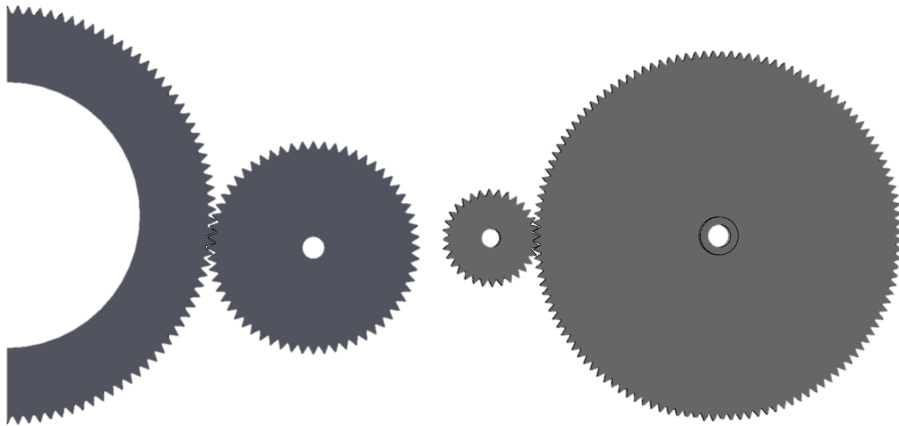


Figura 3.19: Rodas dentadas utilizadas no protótipo

3.3 SUPORTE

O suporte utilizado para este sistema de posicionamento é um pouco atípico, uma vez que é necessário no mesmo suporte, coexistir uma parte fixa onde vai estar integrada uma das rodas dentadas e uma parte móvel, onde vai estar o motor, para que o laser seja capaz de se movimentar e assim conseguir ser controlado conforme o desejado. Foi assim construído um suporte com rolamento como mostra a figura 3.20

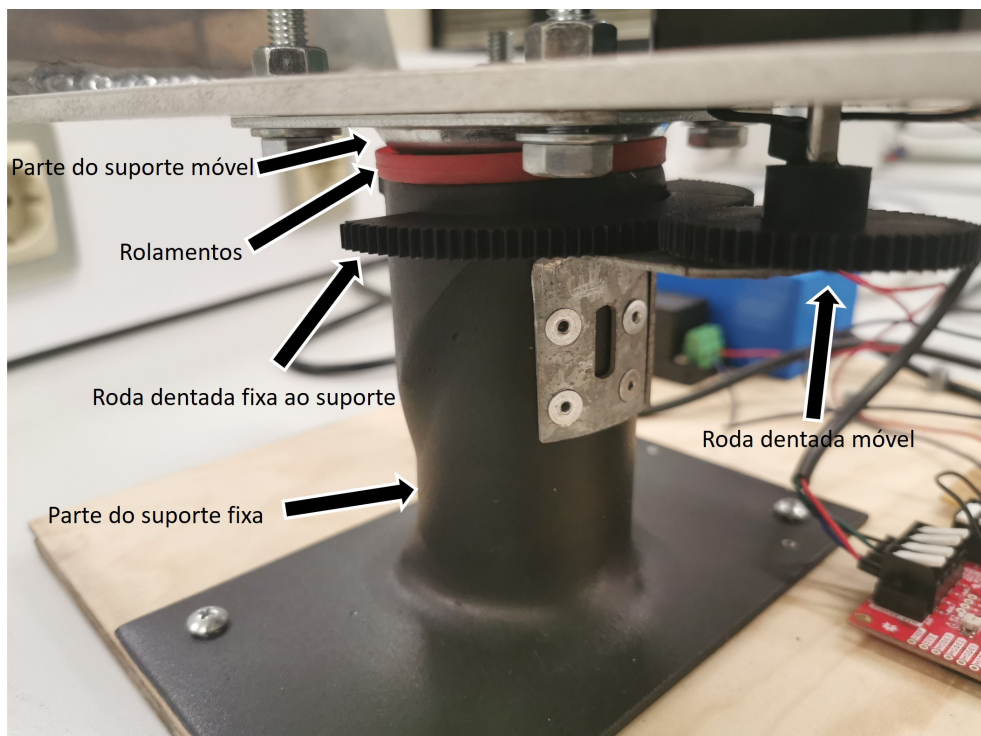


Figura 3.20: Suporte construído

3.4 CÂMARAS

Para este projeto iam ser utilizadas duas câmaras industriais de alta resolução mas não foi conseguida uma ligação entre estas e o computador para uma correta aquisição de imagens, pelo que, para prova de conceito, foram utilizadas duas WebCams Logitech C922 Pro Stream 1080p iguais [10] com ligações USB visíveis na figura 3.21.



Figura 3.21: WebCam Logitech C922 Pro Stream 1080p [10].

As câmaras são controladas pelo computador, que garante uma aquisição de imagens de forma síncrona, sendo também simultânea com a medição da distância efetuada pelo distanciómetro laser. Esta informação, em conjunto, será processada por forma a serem determinadas as coordenadas 3D dos pontos visados no terreno, gerando assim uma representação da morfologia da região observada.

3.5 PROTÓTIPO FINAL

A plataforma de posicionamento do distanciómetro laser, o próprio sensor laser e as duas câmaras de vídeo foram ligadas a um computador que controla todo os sistema e processa os dados captados, constituindo assim a totalidade do protótipo do sistema de mapeamento 3D que este trabalho visava elaborar. Este protótipo encontra-se apresentado na figura 3.22.

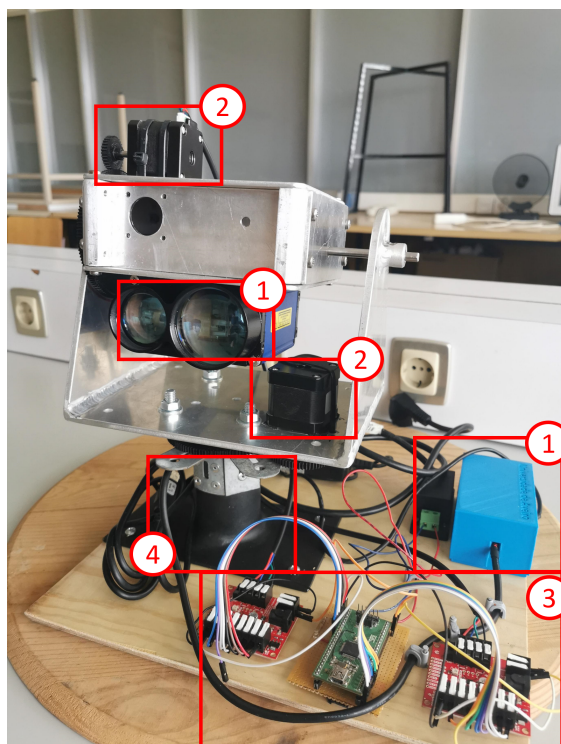


Figura 3.22: Protótipo final, (1- Laser e circuito de comunicação entre o laser e o computador, 2- Motores de passo, 3- PIC32 e drivers do motor, 4- Suporte).

Desenvolvimento da plataforma de software

Neste capítulo vai ser apresentada a plataforma de software desenvolvida de maneira a que utilizador não só seja capaz de a utilizar sem qualquer problema ou ambiguidade, mas também, como um manual para aqueles que pretendem aprimorar ou adicionar novos recursos e/ou funcionalidades.

A plataforma foi desenvolvida com recurso ao programa MATLAB, mais especificamente através da ferramenta App Designer. Esta ferramenta permite criar aplicativos de maneira interativa uma vez que possui um sistema de “Drag and drop“ de componentes visuais que vão fazer parte da graphical user interface (GUI) e usar o editor integrado para programar as funcionalidades destes posteriormente.

Com esta plataforma pretende-se conseguir controlar o protótipo à distância e, para isso, é necessário primeiramente conseguir configurar todos os componentes que constituem este protótipo, visualizar em tempo real as imagens das câmaras, tirar medidas de distância através do distanciómetro laser, e conseguir controlar os dois sentidos de orientação de uma maneira intuitiva.

A página inicial da plataforma está visível na figura 4.1, e esta é composta, assim, por um menu onde é possível escolher diferentes funcionalidades:

- **Configurações**- onde são feitas todas as configurações essenciais;
- **Modo manual**- onde é possível controlar o distanciómetro laser manualmente;
- **Modo Automático**- onde é possível mencionar o número de posições pretendido para o distanciómetro laser percorrer automaticamente;
- **Plot 3D**- onde é vista a figura 3D de todos os pontos tirados desde o início da sessão.

Ainda nesta página inicial, qualquer que seja o menu escolhido, é sempre possível ver as imagens das câmaras em tempo real, guardar as 2 imagens e a respetiva distância simultaneamente e apagar todos os dados caso seja feito um novo teste.

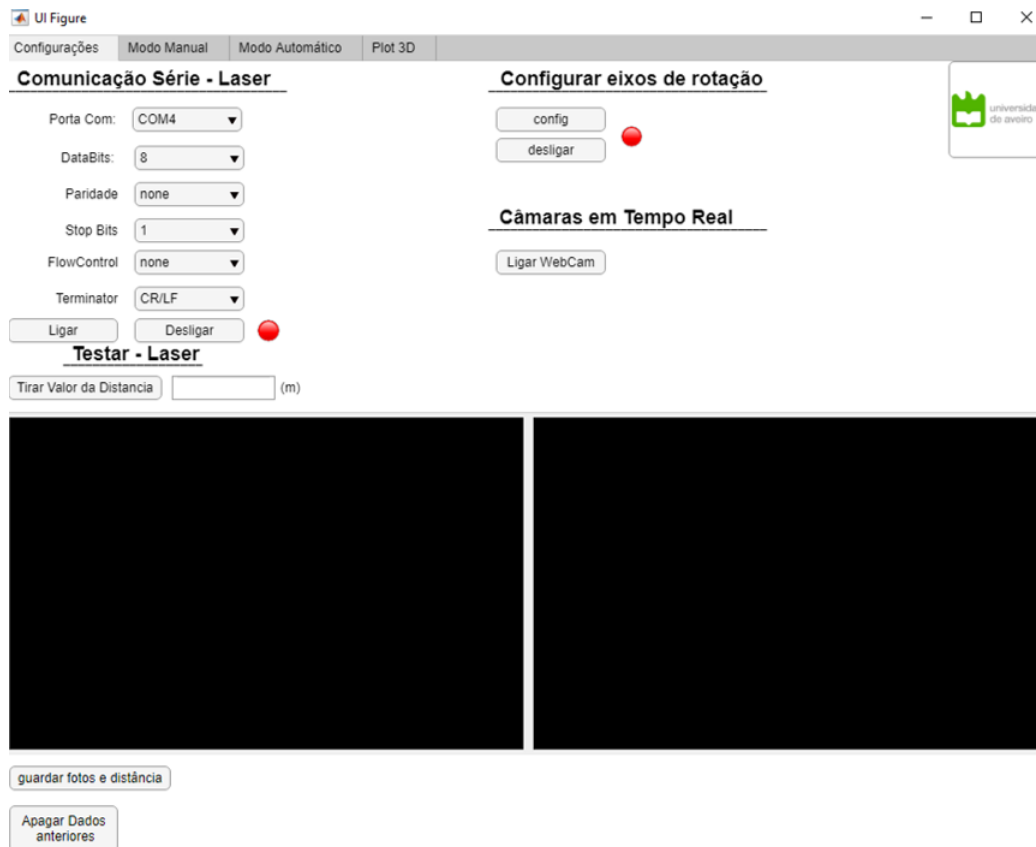


Figura 4.1: Página inicial da plataforma.

4.1 CONFIGURAÇÕES

Neste menu é possível realizar uma comunicação série com o laser, onde todas as configurações são apresentadas já por default para ser estabelecida uma ligação corretamente, onde deve ser mudado apenas a porta “COM“ que varia consoante o utilizador. Para descobrir qual é a porta correta devem ser seguidas as seguintes etapas:

- Na barra de busca procurar: Gestor de dispositivos;
- Selecionar Controladores USB (Universal Serial Bus);
- Verificar qual a porta COM que está a ser utilizada e trocar na GUI;

Uma vez concluído este processo, basta clicar no botão “Ligar“ e assim que o LED vermelho passar a verde, indica que a ligação foi estabelecida com sucesso. Para testar o laser temos o botão “Tirar Valor de Distância“ que sempre que é pressionado é devolvida a distância do ponteiro laser, sem que este valor seja guardado. Existem vários comandos para interagir com o laser, mas considera-se unicamente o comando “dm“ onde o laser devolve uma medida única, assim, o botão “Tirar Valor de Distância“ é composto pelo seguinte trecho de código:

```
fprintf(app.s,[char(27) 'dm' char(13)]); % equivalente a enviar
                                     % ESC -> dm -> Enter
app.a=fscanf(app.s); %ler o valor devolvido pelo laser
```

Ainda neste menu, é feita a configuração dos motores quando é pressionado o botão “config“, que à semelhança do anterior, apenas quando o LED passa a verde é que os motores estão prontos a serem utilizados. Como já foi mencionado anteriormente, existem vários modos de configuração possíveis para controlar os motores mencionados na figura 3.12, no entanto, foi escolhido o “Fixed mode 1/2 step resolution“ como configuração default.

Para comandar o módulo PIC32 através de comandos do MATLAB é necessário alimentar o módulo PIC32 (a 3.3 V) e ligá-lo a uma porta USB do PC através de um cabo (este liga à porta mini-USB do módulo PIC32).

Depois da ligação efetuada, é necessário colocar os ficheiros (pasta hidapi.zip) na pasta de trabalho do MATLAB.

Antes de comunicar com o PIC32 é necessário estabelecer a ligação USB. Tal é efetuado pelos comandos:

```
% Starting the USB link :
h=hidapi(0,hex2dec('04D8'),hex2dec('003F'),64,64);
h.open;
```

Uma vez estabelecida a ligação é possível enviar comandos através da função:

```
h.write_TRC()
```

E os comandos diponíveis são os seguintes:

```
h.write_TRC([hex2dec('0B') n s]);
h.write_TRC([hex2dec('0D') n s]);
```

O primeiro comando coloca o bit ‘n’ da porta B no estado indicado por ‘s’. O índice do bit, ‘n’, pode ser qualquer valor inteiro entre 0 e 7 (apenas o byte menos significativo da porta B). O estado ‘s’ pode ser 0 ou 1 (0 coloca a saída no nível baixo, e 1 coloca no nível alto). O segundo comando é igual ao primeiro, mas atua sobre os bits da porta D do PIC32.

No final do programa é conveniente fechar a ligação USB através dos comandos:

```
% Closing the USB link .
h.close;
h.delete;
```

Uma vez que já conseguida a interação entre o micro-controlador (que vai atuar sobre os drivers dos motores) e o MATLAB, pode-se então passar a explicar o código utilizado para a configuração dos eixos de rotação: Tal como explicado anteriormente, antes de comunicar com o PIC32 é necessário estabelecer a ligação USB pelos comandos já apresentados.

Seguidamente são configurados em “LOW“ todos os pinos que vão ser necessários, e sempre com uma pausa de 0.1 segundos para o micro-controlador não avançar nenhuma linha de código e comprometer a configuração desejada:

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% pin setup %%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%% Eixo vertical %%%%%%%%%%
```

```

ha.write_TRC ([hex2dec('0B') 4 0]); %Enable =0
pause(0.1)
ha.write_TRC ([hex2dec('0B') 6 0]); %sinal de Standby =0
pause(0.1)
ha.write_TRC ([hex2dec('0B') 3 0]); %Mode 0
pause(0.1)
ha.write_TRC ([hex2dec('0B') 2 0]); %Mode 1
pause(0.1)
ha.write_TRC ([hex2dec('0B') 1 0]); %Mode 2
pause(0.1)
ha.write_TRC ([hex2dec('0B') 0 0]); %Mode 3
pause(0.1)

```

%%%%%%%%%% Eixo Horizontal %%%%%%%%%%

```

ha.write_TRC ([hex2dec('0D') 4 0]); %Enable =0
pause(0.1)
ha.write_TRC ([hex2dec('0D') 6 0]); %sinal de Standby =0
pause(0.1)
ha.write_TRC ([hex2dec('0D') 3 0]); %Mode 0
pause(0.1)
ha.write_TRC ([hex2dec('0D') 2 0]); %Mode 1
pause(0.1)
ha.write_TRC ([hex2dec('0D') 1 0]); %Mode 2
pause(0.1)
ha.write_TRC ([hex2dec('0D') 0 0]); %Mode 3
pause(0.1)

```

Sendo depois estes pinos configurados conforme o“Fixed mode 1/2 step resolution“:

%%%%%%%%%% Eixo vertical %%%%%%%%%%

```

%CONTROL MODE SELECT-----%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% MODE 0, MODE 1, MODE 2 , MODE 3 %
% H          L          L          H %
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

ha.write_TRC ([hex2dec('0B') 3 1]); %Mode 0
pause(0.1)
ha.write_TRC ([hex2dec('0B') 2 0]); %Mode 1
pause(0.1)

```

```

ha.write_TRC([hex2dec('0B') 1 0]); %Mode 2
pause(0.1)
ha.write_TRC([hex2dec('0B') 0 1]); %Mode 3
pause(0.1)

```

%%%%%%%% Eixo Horizontal %%%%%%%%%

```

%modo fixo full step resolucio n
ha.write_TRC([hex2dec('0D') 3 1]); %Mode 0
pause(0.1)
ha.write_TRC([hex2dec('0D') 2 0]); %Mode 1
pause(0.1)
ha.write_TRC([hex2dec('0D') 1 0]); %Mode 2
pause(0.1)
ha.write_TRC([hex2dec('0D') 0 1]); %Mode 3
pause(0.1)

```

A configuração é finalizada quando é ativado o sinal de Standby:

```

ha.write_TRC([hex2dec('0D') 6 1]); %sinal de Standby =1
pause(0.1)
ha.write_TRC([hex2dec('0B') 6 1]); %sinal de Standby =1
pause(0.1)

```

```

app.Lamp_2.Color= 'g' % sucesso

```

Finalmente, para serem obtidas imagens em tempo real, basta clicar no botão “Ligar WebCam“, e as câmara ligam e passam a fazer stream ao longo de todo o processo de aquisição de dados. Com todas as configurações efetuadas e as câmaras inicializadas obtém-se um resultado semelhante ao da figura 4.2, estando pronto o cenário para recolher todos os dados necessários para obtenção da morfologia de uma superfície.

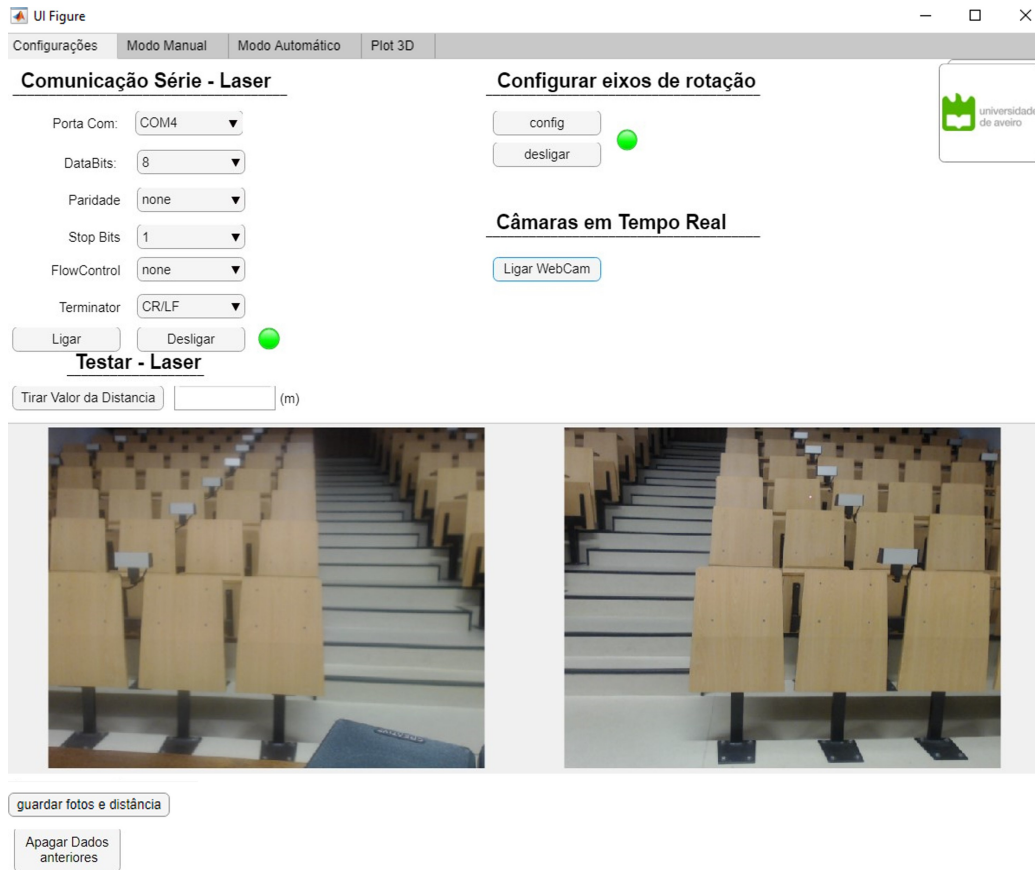


Figura 4.2: Página inicial após configurações feitas e câmaras ligadas.

4.2 MODO MANUAL

Quando em modo manual, o principal objetivo é conseguir colocar manualmente o ponteiro laser em qualquer pixel das imagens, com auxílio dos botões das setas visíveis na figura 4.3.

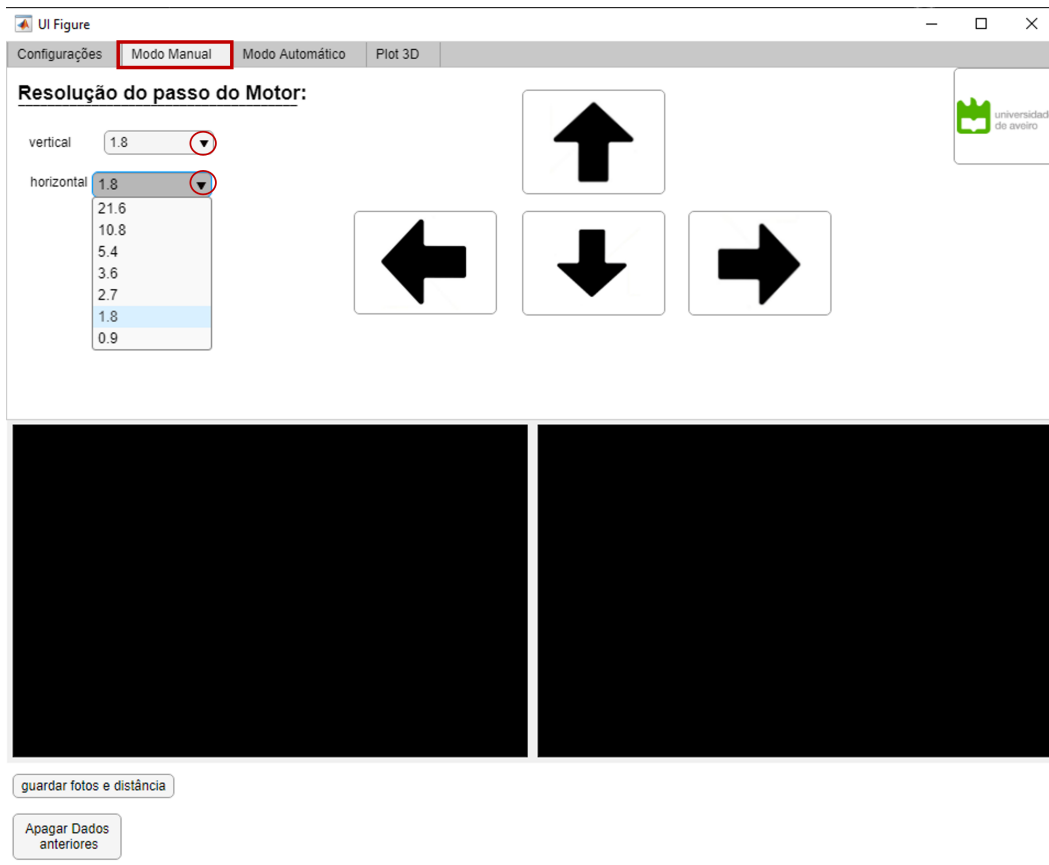


Figura 4.3: Modo Manual.

Neste modo pode-se ainda escolher a resolução do passo do motor tanto vertical, como horizontal, de acordo com o objetivo de cada utilizador. Para conseguir movimentar e controlar os motores consoante a vontade do utilizador foi construída a seguinte função:

```
function step(steps , direction , clock_delay , h , porta )
```

```
%steps      -> numero de passos
%direction  -> 0 -> sentido do ponteiro dos relógio (UP) / (RIGHT)
% 1 -> sentido contrario do ponteiro dos relógio (DOWN) / (LEFT)
%clock_delay -> tempo de espera entre passos
%h          -> objeto criado aquando da ligação USB
%porta      -> '0B' -> Vertical / '0D' -> Horizontal
```

```
h.write_TRC([hex2dec(porta) 4 1]); %Enable =1
pause(0.1)
```

```
if (direction ==1)
    h.write_TRC([hex2dec(porta) 0 1]); %Mode 3 =1
```

```

        pause(0.1)
    else
        h.write_TRC([hex2dec(porta) 0 0]); %Mode 3 =0
        pause(0.1)
    end

    for i=1:steps
        h.write_TRC([hex2dec(porta) 1 0]); %Mode 2=0
        pause(clock_delay)
        h.write_TRC([hex2dec(porta) 1 1]); %Mode 2=1
        pause(clock_delay)
    end

end

end

```

Nesta função *step()*, primeiro é garantido que o sinal “Enable“ está ativo e depois, conforme o valor atribuído no argumento “direction“, o pino do “mode 3“ é atualizado uma vez que é este pino que controla a direção de rotação dos motores. Seguidamente, é aplicada uma onda quadrada que varia com o número de *steps* introduzido, e é este valor que vai fazer variar a resolução de passo, como mostra a tabela 4.1.

Número de passos	Resolução Vertical	Resolução Horizontal
2	0,45°	0,9°
4	0,9°	1,8°
6	1,35°	2,7°
8	1,8°	3,6°
12	2,7°	5,4°
24	5,4°	10,8°
48	10,8°	21,6°

Tabela 4.1: Resoluções de passo disponíveis.

Como já foi mencionado no capítulo anterior, foi usada uma proporção de 1:4 nas rodas dentadas do eixo vertical e uma proporção de 1:2 no par de rodas dentadas do eixo horizontal. Acrescentando esta informação à da configuração usada (“Fixed mode 1/2 step resolution“), ou seja, 1 passo antes que correspondia a 0.45 graus como na figura 3.15, agora, com esta configuração, 1 passo corresponde a 0.225 graus (metade dos 0.45 graus). O mesmo raciocínio aplica-se ao eixo horizontal sendo assim possível obter os dados da tabela 4.1.

4.3 MODO AUTOMÁTICO

Esta funcionalidade permite ao utilizador indicar um número de pontos na horizontal e na vertical através dos campos “N_points_H” e “N_points_V” respetivamente, como é visível na figura 4.4, e o laser vai percorrer a partir do ponto onde está, uma matriz de pontos (“N_points_H’ X ’N_points_V”) automaticamente, sem que seja necessária a presença de um utilizador a tempo inteiro a recolher os dados manualmente, como no modo anterior.

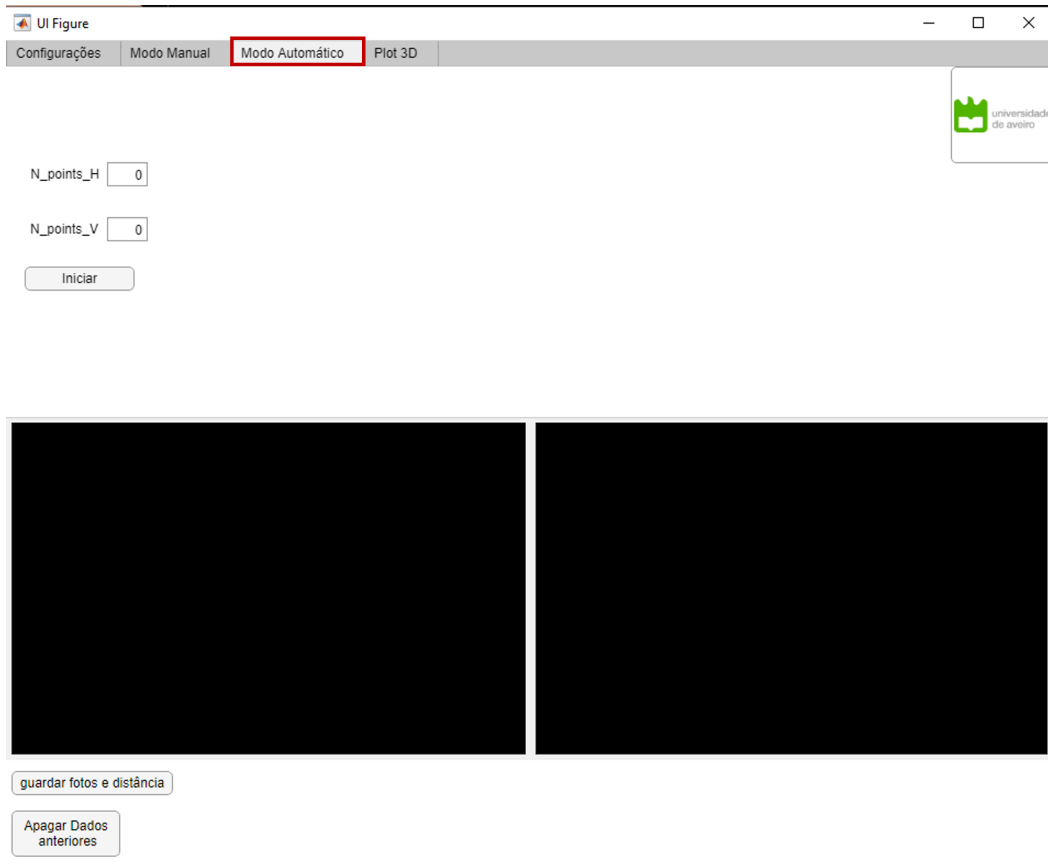


Figura 4.4: Modo automático.

O algoritmo usado para alcançar este objetivo é explicado, na prática, através da análise da figura 4.5, onde foi escolhido um número de pontos na vertical igual a 4, e um número de pontos na horizontal igual a 8.

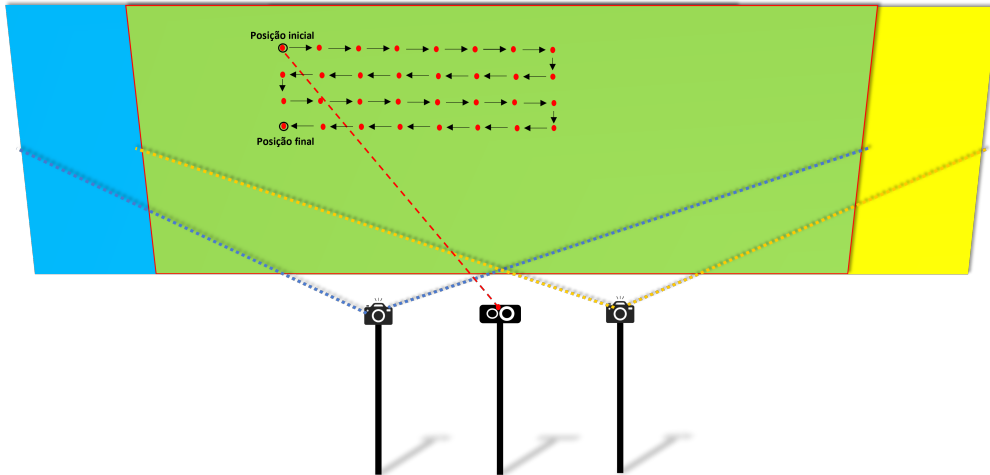


Figura 4.5: Exemplo prático do algoritmo usado no modo automático.

Para cada ponto são guardadas sincronamente as imagens da câmara esquerda e da câmara direita assim como a respetiva distância obtida através do laser, tudo simultaneamente. Cada ponto demora cerca de 3 segundos para garantir que na troca das posições do laser, este consegue estabilizar e depois efetuar o processo de guardar os dados. Neste exemplo, o algoritmo demorava $8 * 4 * 3$ segundos, fazendo um total de 1 minuto e 36 segundos para concluir todas as posições.

O algoritmo utilizado para esta finalidade quando se clica no botão “iniciar“ após terem sido preenchidos os dois campos (“N_pontos_H“ e “N_pontos_V“) é assim, explícito no seguinte trecho de código:

```
% Button pushed function: IniciarButton
function IniciarButtonPushed(app, event)

% Valores introduzidos pelo utilizador:
N_pontos_horizontal=app.N_points_HEditField.Value;
N_pontos_vertical=app.N_points_VEditField.Value;
%
for i=1:N_pontos_vertical
    for j=1: N_pontos_horizontal
        if (rem(i, 2)==0) %Se o indice 'i' for par:
            step(10,1,0,app.h,'0D') %esquerda
        else
            step(10,0,0,app.h,'0D') %direita
        end

        pause(3) % pausa 3 segundos
        % guardar fotos e distancia do laser
        j=j+1;
    end
end
```

```
step(2,1,0,app.h,'0B') % para baixo
pause(3)
i=i+1;
end
```

4.4 PLOT 3D

Quando se inicia a GUI, num levantamento de vários pontos, (pressionar o botão “guardar fotos e distância“), é guardado num diretório a imagem em tempo real da câmara esquerda assim como a da câmara direita, e é atualizado num ficheiro “.mat“ a distância que o laser devolveu para estas fotos, ou seja, é possível fazer load destes ficheiros mais tarde num script do MATLAB e trabalhar sobre estes dados consoante a finalidade do utilizador.

Até ao momento descreveram-se todos os modos diferentes desta plataforma de recolha de dados (imagens das câmaras e valores da distância) e neste modo de plot 3D demonstra-se como a partir de 2 imagens (exemplificadas na figura 4.6) em conjunto com a informação da distância ao ponteiro do laser (estando o ponto de impacto do feixe do laser visível nas imagens), conseguimos obter as coordenadas reais (X,Y,Z) desse mesmo ponto.



Figura 4.6: Imagens capturadas pela câmara esquerda e direita respetivamente.

O algoritmo de determinação das coordenadas de um ponto 3D no terreno em relação ao laser é baseado na seguinte figura 4.7

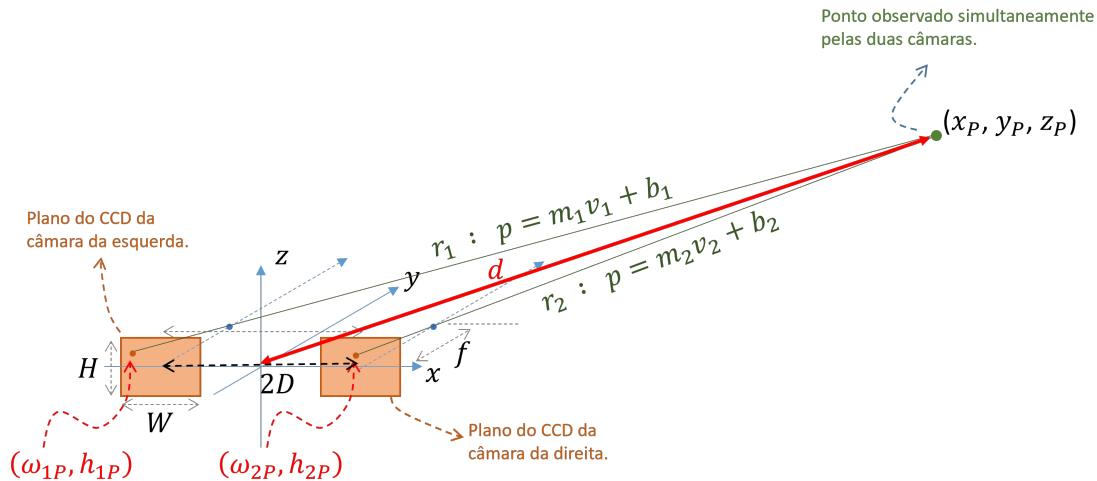


Figura 4.7: Determinação das coordenadas de um ponto 3D Real.

onde:

- **Retângulos laranjas:** Representam os planos CCD das câmaras;
- **2D:** é a distância (em metros) do centro do CCD de uma câmara até ao centro do CCD da outra câmara;
- w_{1P} e h_{1P} : são as coordenadas (em metros) dos píxeis do ponteiro laser na imagem obtida pela câmara esquerda;
- w_{2P} e h_{2P} : são as coordenadas (em metros) dos píxeis do ponteiro laser na imagem obtida pela câmara direita;
- **f:** é a distância focal (em metros);
- **d:** é a medida de distância (em metros) efetuada pelo laser (assume-se que o ponto de origem do laser está no ponto de coordenadas $(0,0,0)$);
- **W:** é largura do CCD em metros;
- **H:** é altura do CCD em metros;
- **r1** e **r2:** são as equações das retas entre píxeis das imagens (esquerda e direita respetivamente) e o ponto a determinar;
- **X_p, Y_p, Z_p:** são as coordenadas reais (em metros) do ponto 3D a determinar.

É de salientar que o plano de cada CCD está associado a um sistema de eixos local, com origem no centro do CCD, e com eixos ω na horizontal (direção positiva para a direita), e h na vertical (direção positiva para cima). Assume-se que os centros dos CCDs estão sobre o eixo x distando “D” da origem deste.

4.4.1 Detecção do ponteiro laser nas imagens capturadas pelas câmaras ((w_{1P}, h_{1P}) e (w_{2P}, h_{2P}))

Para conseguir detetar o ponteiro laser numa imagem é necessário realizar algum processamento de imagem. Assume-se que todos os dados vão ser retirados num ambiente controlado, durante a noite, uma forma eficaz de detecção do ponto do laser, que possui um ponteiro vermelho, mas que ao longe é detetado pelas câmaras como um ponto praticamente branco,

visível na figura 4.6, é aplicar um filtro para tons de cinza, ficando assim com uma imagem onde os píxeis variam numa intensidade de $[0, 255]$, deixando de ser uma imagem a cores (Red Green Blue (RGB)) como mostra a figura 4.8.



Figura 4.8: Imagem da câmara direita após aplicar o filtro BGR2GRAY.

Aplicando um certo *threshhold* à figura anterior com o objetivo de se obter uma figura binária, ou seja, uma imagem onde a intensidade do pixel é 0 ou 255, conseguimos depois selecionar a menor das áreas a branco na imagem, como mostra a figura 4.9 uma vez que a área do ponteiro laser é muito reduzida. O pixel vai ter coordenadas iguais ao centróide desta área.

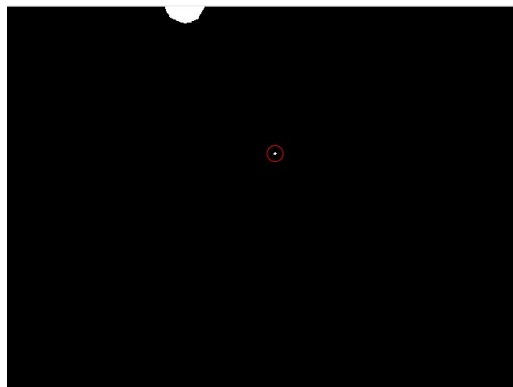


Figura 4.9: Imagem da câmara após aplicar um *Threshold* e obter uma imagem binária.

Atendendo à geometria do sistema, é de notar que as ordenadas y (h_{1P} e h_{2P}) devem ser iguais (as duas câmaras afastam-se apenas segundo a direção “x” pelo que a coordenada vertical dos píxeis é igual nas duas imagens), é assim necessária uma pré-calibração das câmaras antes de se começar a usar a plataforma.

4.4.2 Retas r_1 e r_2

As retas r_1 e r_2 já mencionadas na figura 4.7, podem ser obtidas através das seguintes equações:

$$\begin{cases} p = m_1 v_1 + b_1 \\ p = m_2 v_2 + b_2 \end{cases} \quad (4.1)$$

onde :

$$p = \begin{bmatrix} X_p \\ Y_p \\ Z_p \end{bmatrix}, V_i = \begin{bmatrix} -\omega_{ip} \\ f \\ -h_{ip} \end{bmatrix}, b_i = \begin{bmatrix} (-1)^i * D + \omega_{ip} \\ 0 \\ h_{ip} \end{bmatrix} \quad (4.2)$$

$$m_i \in \mathbb{R}, i = 1, 2$$

Nestas equações (4.1), v_i é um vetor que indica a direção entre o pixel correspondente na imagem “i” e o ponto o ponto “p” (é o vetor que liga o pixel na imagem ao foco do CCD). O vetor b_i corresponde às coordenadas do pixel na imagem “i” no referencial xyz . Conclui-se assim, pelas equações em (4.1) que:

$$p = m v_1 + b_1 \Leftrightarrow \begin{bmatrix} X_p \\ Y_p \\ Z_p \end{bmatrix} = m \begin{bmatrix} -\omega_{1p} \\ f \\ -h_{1p} \end{bmatrix} + \begin{bmatrix} -D + \omega_{1p} \\ 0 \\ h_{1p} \end{bmatrix} \quad (4.3)$$

$$p = m v_2 + b_2 \Leftrightarrow \begin{bmatrix} X_p \\ Y_p \\ Z_p \end{bmatrix} = m \begin{bmatrix} -\omega_{2p} \\ f \\ -h_{2p} \end{bmatrix} + \begin{bmatrix} D + \omega_{2p} \\ 0 \\ h_{2p} \end{bmatrix} \quad (4.4)$$

Este é o resultado proveniente apenas do processamento estereoscópico das duas imagens (a informação do laser não foi ainda incluída). Esta solução tem uma imprecisão muito elevada na determinação da coordenada y_p uma vez que as coordenadas dos píxeis quase nada se alteram à medida que o ponto “p” se afasta das duas câmaras segundo a direção de “d”.

Assim, a inclusão da distância medida pelo laser vai ser necessária para obter uma maior precisão neste eixo “y” através da seguinte equação:

$$d^2 = p^T p = (m v_i + b_i)^T (m v_i + b_i) \rightarrow m^2 (v_i^T v_i) + m (v_i^T b_i + b_i^T v_i) + (b_i^T b_i - d^2) = 0 \quad (4.5)$$

A equação 4.5 é uma equação quadrática relativamente à incógnita “m” cujo valor, por aplicação da fórmula resolvente, é determinado. Note-se que há duas soluções, uma para “m” positivo (sendo esta a que interessa) e outra para “m” negativo (correspondendo ao ponto imagem que estaria por trás das câmaras, colocado simetricamente a “p”). Este valor de “m” pode ser usado, assim, para estimar a coordenada y_p por substituição nas equações das duas retas, ou pode mesmo ser usado para estimar todas as coordenadas de “p”.

Simulação e Resultados

Neste capítulo vão ser apresentados os testes realizados com o protótipo e vão ser apresentados e analisados os respetivos resultados.

O primeiro teste feito consiste em determinar as coordenadas reais sempre de um mesmo ponto que é conhecido em relação ao laser, neste caso, o ponteiro do laser da imagem da figura 5.1 que possui as coordenadas reais de X: 0.72 metros, Y: 2.04 metros e Z: -0.05 metros .

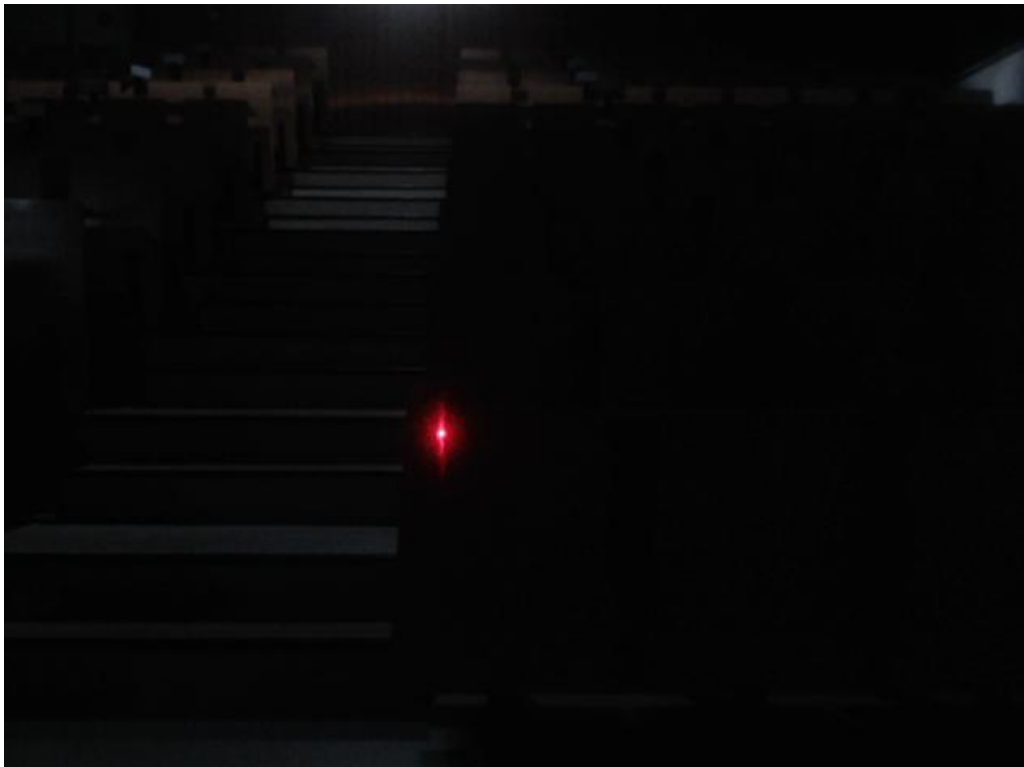


Figura 5.1: Imagem da câmara direita.

Antes de iniciar o teste através da plataforma de software devem ser tidos em conta alguns pré-requisitos que já foram referidos ao longo desta dissertação, tais como:

- Colocar as câmaras ao mesmo nível do laser;
- Saber a distância entre as câmaras em metros;
- Saber os parâmetros das câmaras:
 - Saber qual é a distância focal das câmaras, em metros;
 - Largura do Charge-Coupled Device (CCD), em metros (W);
 - Altura do CCD, em metros (H);
 - Número de linhas e de colunas da imagem do CCD;
- Calibrar as câmaras;
- Estar num ambiente escuro, para simular um ambiente de noite.

Para este teste foi montado o seguinte *setup* da figura 5.2, onde foi imposto que as câmaras se situassem à mesma distância em relação ao laser, e que estão à mesma altura e alinhadas com o distanciômetro laser.

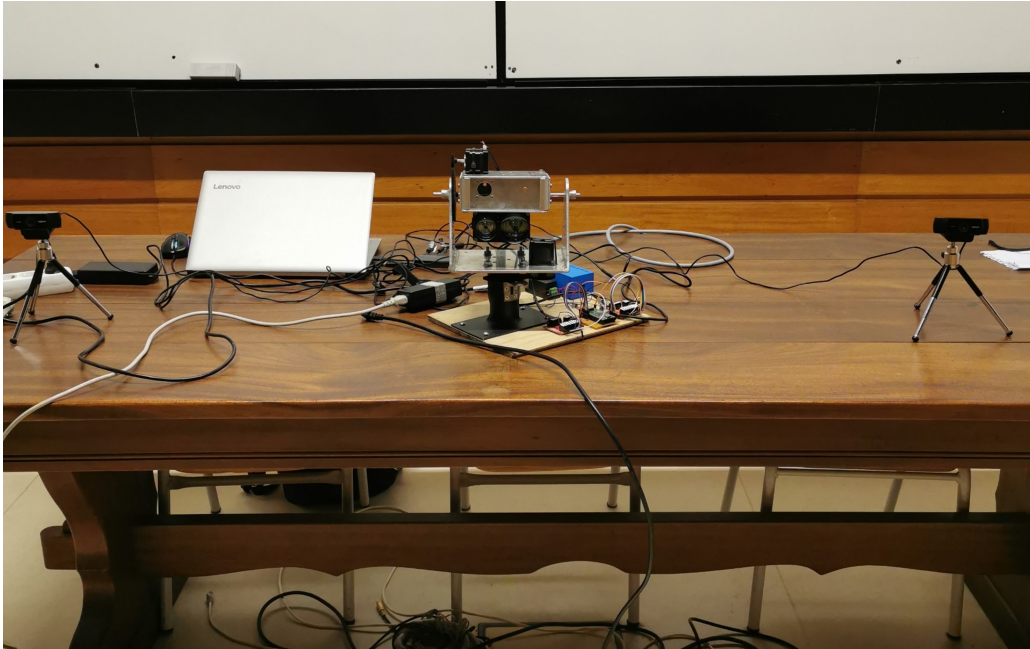


Figura 5.2: *setup* montado para o primeiro teste.

Para calibrar as câmaras foi desenvolvido um *script* em *python* que permite ao utilizador interagir com as imagens produzidas em tempo real das duas câmaras, onde é devolvido o pixel da posição escolhida. Ou seja, na prática, seleciona-se um ponto que esteja presente nas duas imagens (esquerda e direita) e calibra-se as câmaras até que esse pixel tenha o mesmo valor da ordenada y nas duas imagens como é mostrado na figura 5.3.

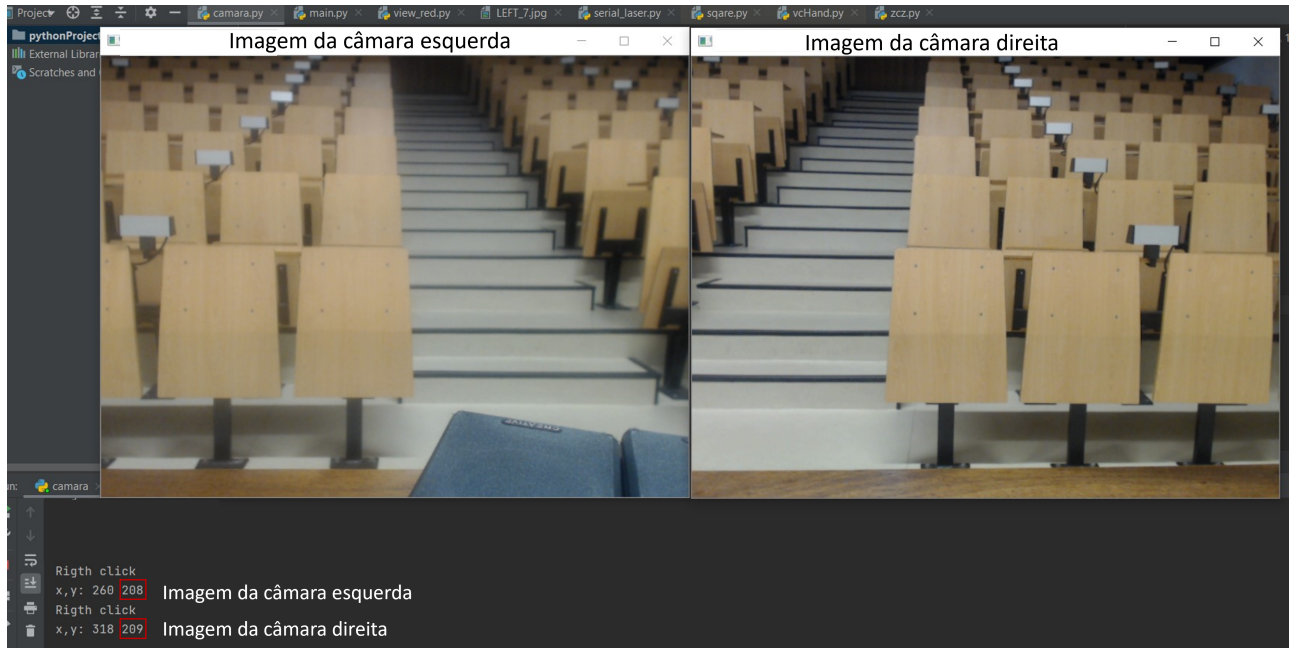


Figura 5.3: Pré-calibração das câmaras.

Completados todos os pré-requisitos, já com um ambiente escuro como mostra a figura 5.1, procede-se à recolha de dados com a plataforma de software desenvolvida.

Neste teste foram tirados cerca de 46 pontos seguidos sem nenhum movimento do laser, num ponto conhecido (X : 0.72 metros, Y : 2.04 metros e Z : -0.05 metros), sendo apresentados os resultados na figura A (no apêndice A), onde se verificaram os valores de desvio padrão em cada coordenada. Com os dados desta figura, foi determinando o erro em cada coordenada em metros visível na figura 5.4

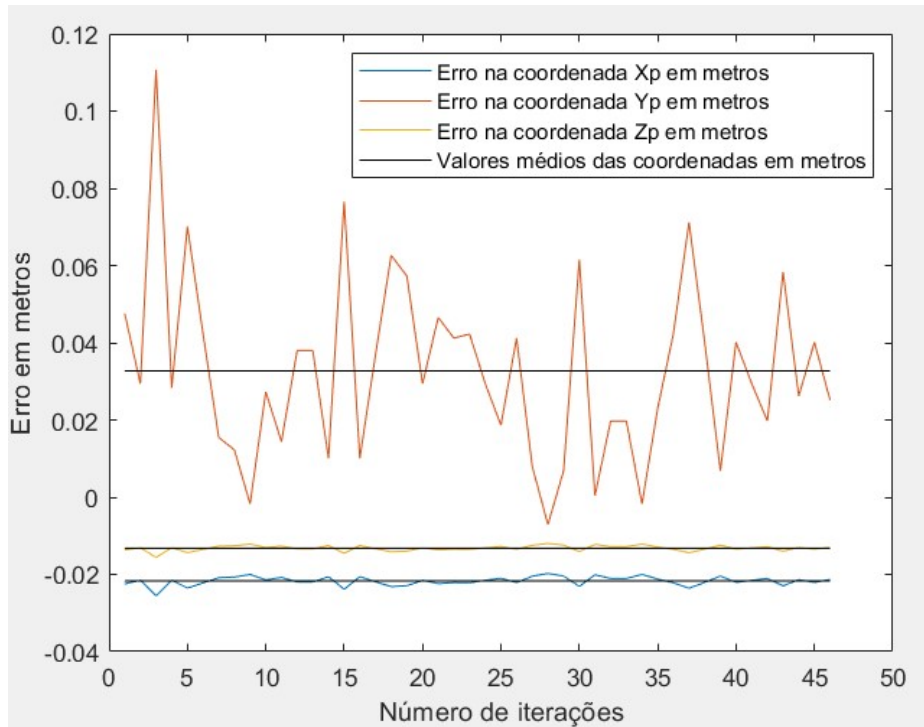


Figura 5.4: Erro das coordenadas em metros

Pela análise da figura 5.4 verifica-se que a maior fonte de erro é a distância dada pelo laser (na coordenada Y_p), no entanto, esta falta de precisão pode ser explicada pelo facto deste distanciômetro laser ser específico para ser usado a grandes distâncias pelo que a esta curta distância (2.18 metros) pode não estar na sua melhor gama de execução. O laser possui também um modo de aquisição de distâncias múltiplas, devolvendo a média destas em vez uma medida única, como foi utilizado neste teste, que iria melhorar os resultados obtidos.

Os valores reais (X : 0.72 metros, Y : 2.04 metros e Z : -0.05 metros), também são portadores de erro uma vez que foram medidos com uma fita métrica.

O segundo teste realizado visa assemelhar-se ao processo do levantamento da morfologia 3D de uma superfície. Uma vez que não se conseguiu testar o protótipo na praia, foi realizado o mesmo teste, mas num anfiteatro, para prova de conceito.

Mantendo o mesmo *setup* da figura 5.2, e cumprindo todos os pré-requisitos mencionados já no teste anterior, foi levantado parcialmente o mapeamento 3D do anfiteatro onde o sistema foi montado para os testes.



Figura 5.5: Vista do sistema montado num anfiteatro, para a realização dos testes..

De salientar que estes dados foram retirados de uma maneira ligeiramente diferente da pensada inicialmente.

Em vez dos pontos serem tirados apenas na interseção das câmaras como mostra a figura 3.7, que seria uma área sensivelmente pequena, como se vê pelas imagens da figura 4.6, os pontos foram assim adquiridos primeiramente com a câmara do lado esquerdo e depois com a câmara do lado direito. Os dados foram posteriormente tratados e agregados, aumentando drasticamente “a área visada” sendo assim possível fazer o levantamento do espaço inteiro da figura 5.5.

Neste teste foram medidos no total 460 pontos (em cerca de 30 minutos, uma vez que foi utilizado o modo manual da plataforma), e em 5 deles não foram detetados o ponteiro laser, ou seja, uma falha de 1.08% na deteção do laser.

O resultado final obtido é mostrado na figura 5.6 que apresenta claramente o espaço do anfiteatro com as coordenadas reais em relação à posição do laser, mas poderia ser melhorado se fossem medidos mais pontos de modo a obter uma figura com mais algum detalhe, no entanto demonstra a potencialidade que o protótipo desenvolvido dispõe.

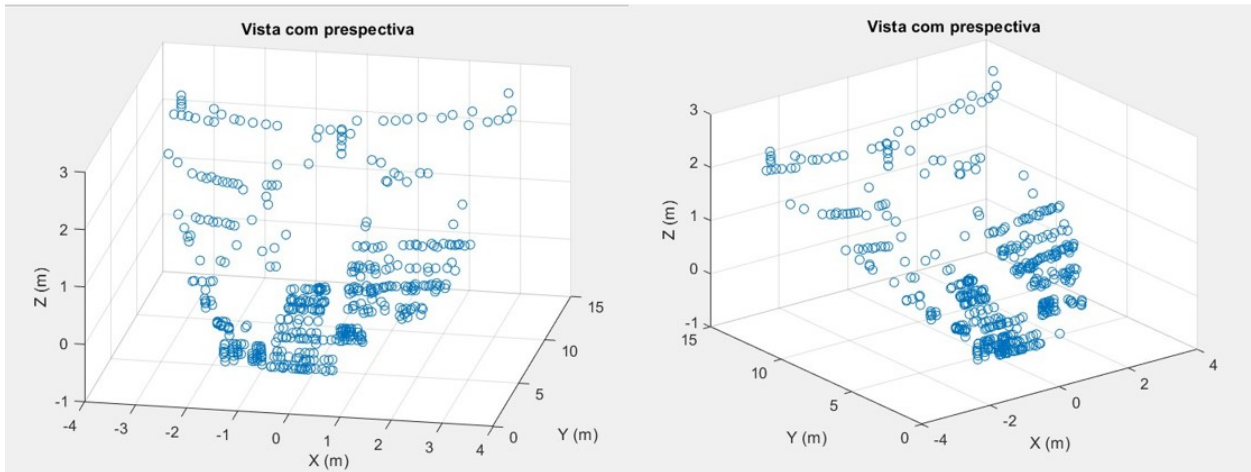


Figura 5.6: Levantamento da morfologia 3D do anfiteatro.

São também disponibilizados na figura 5.7 os planos 2D relativos aos dados utilizados para a construção do mapa 3D da figura 5.6.

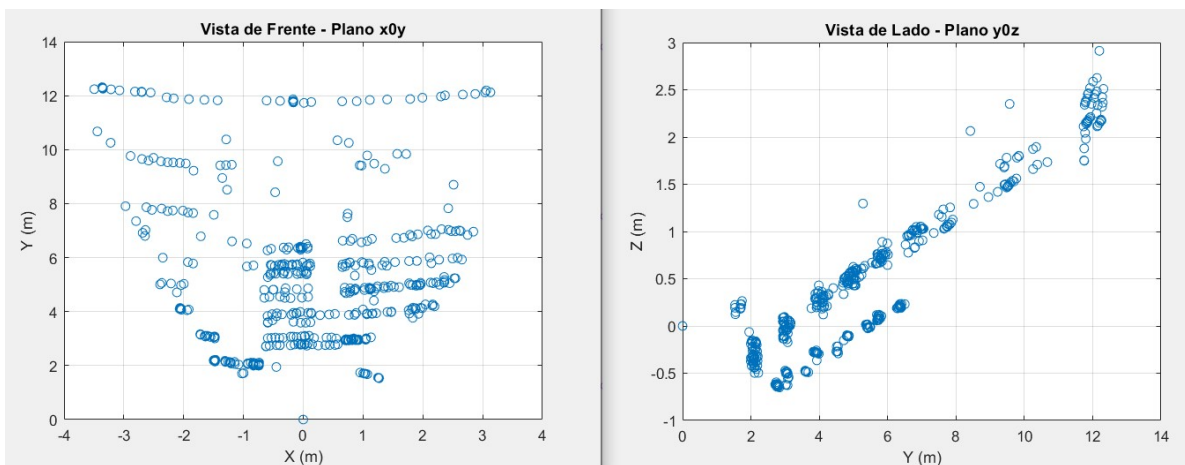


Figura 5.7: Planos 2D do anfiteatro.

Conclusão e Trabalho Futuro

Neste trabalho foi desenvolvido um protótipo, capaz de determinar a morfologia de uma superfície, composto essencialmente por 2 câmaras que capturam imagens (simultaneamente), da mesma superfície, mas de pontos diferentes, e um distanciômetro laser que devolve a distância a que esses pontos estão, através de um suporte que foi construído com 2 eixos de rotação, com recurso a 2 motores de passo.

Para conseguir controlar este protótipo à distância foi desenvolvida uma aplicação em MATLAB com a ferramenta *App Designer* que tem como principal objetivo ser uma aplicação intuitiva onde é possível configurar todos os módulos que constituem o protótipo, estimar as coordenadas de pontos da superfície visada, de um modo manual (onde o utilizador consegue controlar com precisão, a posição do laser) ou de modo automático (onde o utilizador especifica uma matriz de pontos, e o protótipo regista as respetivas imagens e distâncias automaticamente), e no final, obter uma imagem em 3D da superfície em coordenadas reais em relação ao laser, através do uso de um dos modos, ou até mesmo uma junção dos dois modos de utilização.

Como trabalho futuro, deve ser trocado o cabo do laser, que aplica demasiada tensão sobre os motores de passo, condicionando principalmente o funcionamento do modo automático.

Outro aspeto que deve ser melhorado no protótipo é a troca das câmaras utilizadas que devem ser substituídas por umas de melhor resolução, como se pretendia inicialmente. Esta necessidade existe uma vez que nos resultados obtidos, a maior distância registada é cerca de 14 metros, mas o laser consegue atingir até 300 metros de distância se a superfície for refletora, que é o caso de uma praia. Assim, com o aumento da distância, também a configuração dos drivers dos motores deve ser adaptada de modo a obter uma resolução mais pequena, principalmente no eixo vertical. A nível de software, deveria ser implementado um sistema capaz de avisar o utilizador em tempo real que, sempre que determinado ponto não fosse corretamente identificado, esta medida era invalidada e consequentemente, repetida.

Ainda como trabalho futuro, seria desejável testar o protótipo num ambiente de praia, onde este seria montado numa cabine já existente na Praia de Mira, que se encontra no topo

de uma torre. Com o protótipo instalado, conjuntamente com câmaras de elevada resolução (que já foram adquiridas), seriam efetuados testes no terreno, aferindo-se da eficácia desta nova metodologia de levantamento da morfologia dessa zona costeira.

Simulação e Resultados

Iteração	Distância (m)	X (m)	Y (m)	Z (m)
1	2,202	0,697566	2,087619	-0,06367
2	2,185	0,698464	2,069394	-0,06311
3	2,261	0,69446	2,150707	-0,0656
4	2,184	0,698516	2,068321	-0,06308
5	2,223	0,696459	2,110102	-0,06436
6	2,197	0,69783	2,082261	-0,06351
7	2,172	0,69915	2,055442	-0,06269
8	2,169	0,699309	2,052221	-0,06259
9	2,156	0,699997	2,038253	-0,06216
10	2,183	0,698569	2,067248	-0,06305
11	2,171	0,699203	2,054369	-0,06265
12	2,193	0,698041	2,077973	-0,06337
13	2,193	0,698041	2,077973	-0,06337
14	2,167	0,699415	2,050073	-0,06252
15	2,229	0,696143	2,11652	-0,06455
16	2,167	0,699415	2,050073	-0,06252
17	2,192	0,698094	2,076901	-0,06334
18	2,216	0,696828	2,102611	-0,06413
19	2,211	0,697092	2,097258	-0,06396
20	2,185	0,698464	2,069394	-0,06311
21	2,201	0,697619	2,086547	-0,06364
22	2,196	0,697883	2,081189	-0,06347
23	2,197	0,69783	2,082261	-0,06351
24	2,185	0,698464	2,069394	-0,06311
25	2,175	0,698992	2,058663	-0,06278
26	2,196	0,697883	2,081189	-0,06347
27	2,165	0,699521	2,047925	-0,06246
28	2,151	0,700261	2,032878	-0,06199
29	2,164	0,699573	2,04685	-0,06242
30	2,215	0,696881	2,101541	-0,0641
31	2,158	0,699891	2,040403	-0,06223
32	2,176	0,698939	2,059736	-0,06282
33	2,176	0,698939	2,059736	-0,06282
34	2,156	0,699997	2,038253	-0,06216
35	2,179	0,69878	2,062956	-0,06292
36	2,197	0,69783	2,082261	-0,06351
37	2,224	0,696407	2,111172	-0,06439
38	2,195	0,697936	2,080117	-0,06344
39	2,164	0,699573	2,04685	-0,06242
40	2,195	0,697936	2,080117	-0,06344
41	2,185	0,698464	2,069394	-0,06311
42	2,176	0,698939	2,059736	-0,06282
43	2,212	0,697039	2,098329	-0,064
44	2,182	0,698622	2,066175	-0,06301
45	2,195	0,697936	2,080117	-0,06344
46	2,181	0,698675	2,065102	-0,06298
Média	2,188087	0,698301	2,072687	-0,06321
Desvio padrão	0,0217	0,0011	0,0233	0,0007

Figura A.1: Dados do primeiro teste.

Referências

- [1] P. Baptista, T. R. Cunha, A. Matias, C. Gama, C. Bernardes e G. Óscar Ferreira, «New land-based method for surveying sandy shores and extracting DEMs: the INSHORE system,» English, *Springer Science+Business Media*, fev. de 2011. DOI: 10.1007/s10661-011-1873-5.
- [2] F. SANTOS, A. C. TEODORO, P. BATISTA e G. VITOR LOPES, «DESENVOLVIMENTO DE METODOLOGIAS E TÉCNICAS AVANÇADAS DE MONITORIZAÇÃO DA ZONA COSTEIRA (PROJETO MOZCO),» sér. 6.as Jornadas de Hidráulica, Recursos Hídricos e Ambiente. 2011, ISBN: 978-989-95557-5-4.
- [3] R. Holman e G. Stanley J, «The history and technical capabilities of Argus,» 2007, pp. 477–491.
- [4] G. Aarninkhof, «Nearshore Bathymetry Derived from Video Imagery,» 2003.
- [5] U. Andriolo, E. Sánchez-García e R. Taborda, «Operational Use of Surfcam Online Streaming Images for Coastal Morphodynamic Studies,» 2019.
- [6] S. Godfrey, J. Cooper, F. Bezombes e G. Andrew Plater, «Monitoring coastal morphology: the potential of low-cost fixed array action cameras for 3D reconstruction,» English, *EARTH SURFACE PROCESSES AND LANDFORMS*, 2020. DOI: 10.1002/esp.4892.
- [7] «Manual optoNCDT-ILR-1191,» MICRO-EPSILON Eltrotec GmbH. URL: <https://www.micro-epsilon.com/download/manuals/man--optoNCDT-ILR-1191--en.pdf>.
- [8] «Motor de passo FIT0278.» URL: <https://uk.farnell.com/dfrobot/fit0278/stepper-motor-1-8deg-5mm-bipolar/dp/3879694>.
- [9] «SparkFun ProDriver.» URL: https://www.digikey.pt/en/maker/projects/sparkfun-prodriver-hookup-guide/929d9ddcde4d45b59c1d905866484e8c?_ga=2.210120092.888480930.1653384432-304154658.1653384432.
- [10] «WebCams Logitech C922 Pro Stream 1080p.» URL: <https://www.logitech.com/pt-br/products/webcams/c922-pro-stream-webcam.960-001087.html>.