FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Sistema de Reconhecimento de Objetos para Demonstrador de Condução Robótica Autónoma

João Nuno Ferreira Batista

Mestrado Integrado em Engenharia Informática e Computação

Orientador: Armando Jorge Sousa (Professor Auxiliar da FEUP)

19 de Julho de 2011

Sistema de Reconhecimento de Objetos para Demonstrador de Condução Robótica Autónoma

João Nuno Ferreira Batista

Mestrado Integrado em Engenharia Informática e Computação

Aprovado em provas públicas pelo júri:

Presidente: Paulo José Cerqueira Gomes da Costa (Professor Auxiliar da FEUP)

Vogal Externo: Agostinho Gil Teixiera Lopes (Investigador Auxiliar da Universidade do Minho)

Orientador: Armando Jorge Miranda de Sousa (Professor Auxiliar da FEUP)

---

19 de Julho de 2011

Resumo

O aparecimento de dispositivos RGBD, ou seja, sensores que além de captarem imagens RGB, (como qualquer câmara) também captam informação de profundidade, disponíveis para o utilizador comum tornou acessível um sensor que de outra forma seria demasiado caro para um demonstrador de robótica autónoma de baixo custo. Desta forma, pode-se equipar um robô de demonstrações de robótica autónoma com o *Kinect* e usufruir de um sensor de profundidade.

Fazendo o robô reagir a objetos simples, tal como perseguir uma esfera ou fugir de um cilindro e mesmo a combinação de ambos os comportamentos, permite criar demonstrações interativas e apelativas para sensibilizar a assistência ao mundo da robótica e do que através dela se pode criar. Para concretizar estes objetivos, tirando partido da disponibilidade deste tipo de sensor, torna-se necessário desenvolver software de reconhecimento em tempo real de objetos simples (esferas, cones e cilindros) para poderem ser utilizados nas demonstrações de robótica autónoma.

Todo este trabalho implica uma pesquisa científica cuidada dos métodos e técnicas que existem para o reconhecimento dos ditos objetos e também dos diferentes tipos de demonstradores que existem, que embora sirvam diferentes propósitos, demonstram exemplos de robótica autónoma.

É feito também um estudo da precisão do *Kinect* na leitura que faz do posicionamento de objetos no espaço por ele detetado, sendo que se conclui que tem uma precisão bastante satisfatória.

Neste trabalho confirma-se que a possibilidade de identificação de objetos complexos é uma realidade, podendo ser feita com recurso à ajuda de bibliotecas informáticas e algumas metodologias simples, das quais se pode extrair resultados bastantes interessantes, sendo que foi utilizada a mesa como prova de conceito de objetos complexos.

Abstract

The appearance of RGBD devices, which means that these are devices that not only do they capture the RGB information they also capture the depth of a pixel, and their availability as a consumer of the shelf device has made possible their incorporation in any robotic autonomous driving demonstrator cheaply. This way a *Kinect* can be taken advantage of as a depth sensor.

Making the robot interact with simple objects, like follow a sphere, or avoid a cylinder or even combining both behaviours , allows the creation of a appealing and interactive demonstrations to sensitize the public to the world of robotics. To make this a reality, taking advantage of these kind of sensors, it becomes necessary to develop software to recognize simple objects in real time in order for them to be used in autonomous robotics demonstrations.

All this work implies a careful scientific review of the existing methods to recognize the said objects and also review the different kinds of demonstrators and their purpose.

It is also necessary to perform a evaluation of the *Kinect*'s precision on evaluating objects placement in the space it detects, and it was found that its precision is very good.

This work confirms the ability to identify complex objects, and it can be done using software libraries and some simple methodologies, from which we can gather rather interesting results using the table as proof of concept to complex objects.

Conteúdo

Lista de Figuras

Lista de Tabelas

LISTA DE TABELAS

Abreviaturas e Símbolos

RGB	Imagem a cores definida com canais de vermelho (R), verde (G) e azul(B).
Sensor RGBD	Sensor que além de imagem (RGB) devolve informação de profundidade. (D)
LIDAR	Light Detection and Ranging

Capítulo 1

Introdução

1.1 Enquadramento

Esta dissertação de Mestrado surgiu de um acumular de fatores importantes que lhe dão contexto e definem o âmbito da mesma, cuja listagem e descrição se seguem.

Um dos fatores que levaram a que esta dissertação fosse proposta foi o facto de ter sido desenvolvido, no âmbito de uma dissertação de Mestrado do curso de Mestrado Integrado em Engenharia Eletrotécnica e Computadores, um demonstrador de robótica autónoma com vista a fazer demonstrações e também participar em competições de condução autónoma.

Entretanto também foi disponibilizado no mercado, um sensor de RGBD da *Microsoft* chamado *Kinect* que foi desenvolvido para o sistema de jogos *X-Box 360* de modo a se interagir com jogos e software totalmente sem comandos, respondendo, desta forma, ao movimentos e gestos dos jogadores. Pouco depois da sua saída para o mercado foram desenvolvidos controladores *opensource*, sendo que ficou aberta a possibilidade de se utilizar o *Kinect* como um sensor em qualquer robô com portas USB.

Estes dois fatores, aliados ao desejo de se fazer demonstrações portáveis de robótica autónoma tornaram possível esta dissertação, em que se estuda a identificação de objetos 3D em tempo real recorrendo à informação de profundidade que o *Kinect* fornece.

Uma noção basilar é que o tipo de informação a ser processada serão perceções, portanto será sempre a informação obtida pelo *Kinect* mas com contexto, sendo essa perceção considerada sempre em unidades SI.

Em suma, o que se pretende é, utilizando o *Kinect*, criar um sistema que reconheça objetos para um robô que já existe.

1.2 Projeto

Além deste documento que expõe o conhecimento e as tecnologias utilizadas para resolver o problema, e analisa de forma crítica os resultados obtidos, existe também um projeto desenvolvido que permite o reconhecimento de objetos 3D. Este projeto estará também já preparado para se integrar num sistema de um robô que já se encontra desenvolvido, que usa como sensor principal o *Kinect*. Este software permitirá configurar vários parâmetros, de uma forma intuitiva, para ser possível fazer o reconhecimento de objetos 3D e que seja claro para o utilizador final o que está a acontecer.

1.3 Motivação e Objetivos

A motivação principal desta dissertação é fazer o reconhecimento de objetos em tempo real através do sensor de profundidade (utilizando o *Kinect*), e após reconhecidos pelo sistema, serem utilizados para fazer demonstrações de robótica autónoma. As características mais importantes serão: o reconhecimento de objetos 3D em tempo real e a identificação das características dos mesmos.

Esta dissertação pode ser considerada um sucesso se for possível reconhecer vários objetos simples através do sensor de profundidade, tais como planos, cilindros, paralelepípedo e de objetos complexos que sejam compostos pelos objetos simples.

Adicionalmente serve como uma forma de avaliar a capacidade do *Kinect*, como sensor RGBD, de reconhecer objetos do dia a dia e testar as suas limitações.

1.4 Estrutura da Dissertação

Para além da introdução, nesta dissertação é descrito o estado da arte e são apresentados trabalhos relacionados. É também feita a descrição da implementação, apresentam-se os resultados extraídos desta dissertação e faz-se uma análise crítica dos mesmos.

Capítulo 2

Revisão Bibliográfica

2.1 Introdução

Neste capítulo são analisadas todas as áreas científicas relevantes para que esta dissertação possa ser realizada com sucesso, tirando partido do conhecimento mais avançado do que está a ser feito a nível mundial e até equacionar novas abordagens para melhorar as soluções existentes para o problema do reconhecimento de objetos. Mais particularmente será analisado o que está feito em termos de reconhecimento de objetos 3D tanto na vertente científica como das bibliotecas informáticas existentes nesse âmbito.

Faz-se também uma análise ao que existe em termos de demonstradores de robótica autónoma para enquadrar o trabalho a ser desenvolvido respeitante à identificação dos objetos, que sendo reconhecidos, influenciam o comportamento do robô.

2.2 Sensores de Profundidade

2.2.1 Time of Flight Camera

Esta câmara tem recetores para obter o tempo de voo (i.e.: time-of-flight) de uma partícula de luz de modo a saber a que distância se encontram os obstáculos no campo de visão da câmara. Existem várias implementações desta tecnologia, mas a mais usual é ter um emissor de luz, por norma um laser, que envia pulsos de luz que são refletidos nos obstáculos e são captados por um sensor que a cada "pixel" recebe a distância a que o objeto se encontra.

É um sensor que tem muitas aplicações, entre as quais a deteção de objetos nas imediações pois permite obter distâncias dos objetos que se encontram no seu campo de visão sem que isso implique um maior custo de cálculo através de algoritmos de visão por computador.

2.2.2 Light Detection and Ranging

Os sistemas LIDAR (*Light Detection and Ranging*) utilizam um feixe de luz rotativo para fazer mapeamento 3D do ambiente que o circunda. A vantagem destes sistemas é que, com feixes de luz muito estreitos conseguem obter uma resolução muito grande, sendo o ideal para aplicações críticas, tais como o alunar de sondas no caso da NASA [Kei10].

Os LIDAR utilizam técnicas de *backscattering* para conseguir uma medição precisa de distância (além de algumas características do alvo), sendo necessário fazer algumas considerações prévias sobre o meio em que se vai propagar o laser para escolher o melhor comprimento de onda.

2.2.3 Kinect

O *Kinect* é um sensor 3D desenvolvido pela *Microsoft*, cujo objetivo principal era ser utilizado a par com o sistema *X-Box 360*, para a interação com jogos sem recorrer a controladores, sendo que através deste sensor os movimentos do jogador controlam o jogo. Este foi o primeiro sensor 3D disponível para o utilizador comum e permitiu a massificação do acesso da comunidade cientifica a este tipo de sensores, que de outra forma seriam demasiado dispendiosos.

As aplicações deste sensor fora do âmbito original para que foi desenvolvido são inúmeras, e rapidamente foi desenvolvido um controlador *opensource* para se desenvolver aplicações em qualquer sistema operativo.

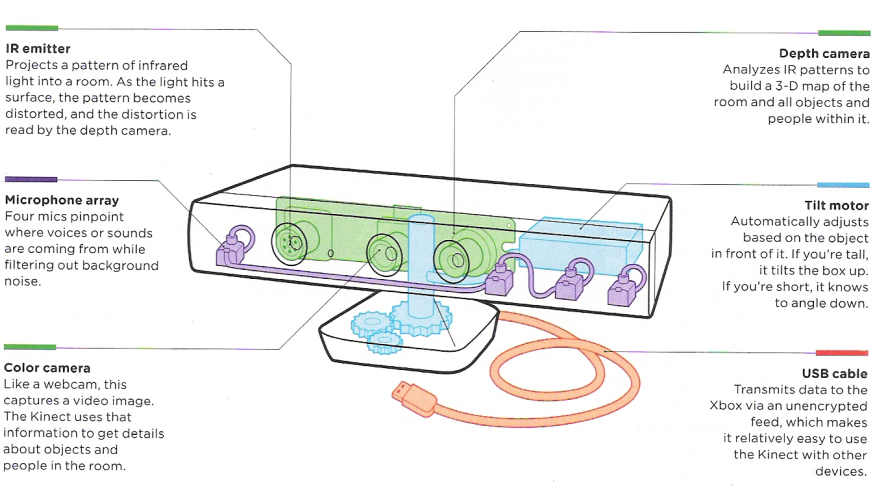


Figura 2.1: Composição do Kinect [Tan11]

O *Kinect* tem, como se pode ver na figura 2.1, uma câmara normal RGB contudo o sensor que extrai a perceção de profundidade é a câmara de infra-vermelhos à direita que capta os infra-vermelhos enviados pelo emissor situado à esquerda.

Entretanto foi lançado no *Windows SDK* um controlador e uma *framework* oficiais para se desenvolver aplicações fazendo uso do *Kinect* na plataforma *Windows*.

2.3 Técnicas de Reconhecimento de Objetos

Nesta secção do documento apresenta-se e explora-se as técnicas mais promissoras de deteção e reconhecimento de objetos em imagens 2D e em estruturas de representação em 3D que assumem a maior importância no âmbito do ramo científico da visão por computador.

2.3.1 Scale Invariant Feature Transform

Scale-Invariant Feature Transform (SIFT) [Low99] é uma técnica para o reconhecimento de objetos bastante popular em computação visual. Para fazer o reconhecimento de objetos, esta técnica necessita de um treino prévio, onde são extraídas características que não variam com a escala, rotações nem com projeções em 3D, sendo também parcialmente resistente a diferenças em iluminação.

A qualidade desta técnica está dependente da qualidade das características que extrai, e do facto de estas serem invariantes apesar das transformações que se possa aplicar à imagem.

A extração de características é feita em vários passos. O primeiro é a aplicação da função gaussiana na direção horizontal a todas as linhas de pixeis e de seguida na vertical em todas as colunas. É utilizada também uma pirâmide de imagens onde se fez uma progressiva interpolação bilinear para suavizar a imagem, sendo que a função gaussiana é aplicada a todas as camadas de modo a que cada uma seja comparada às suas adjacentes para determinar os máximos e os mínimos.

$$g(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-x^3/2\sigma^2} \tag{2.1}$$

O resultado desta análise é um conjunto de vetores que representam as características do objeto.

Figura 2.2: Exemplo de Deteção utilizando SIFT [Low99]

2.3.2 Speeded Up Robust Feature

Speeded Up Robust Feature (SURF) é uma técnica bastante próxima de SIFT2.3.1, contudo tem a vantagem de ser mais robusta às transformações que se pode fazer às imagens conseguindo também, ao mesmo tempo, aumentar a performance e a repetibilidade da deteção nas imagens [BTGL06].

As melhorias enunciadas são conseguidas através da escolha cuidadosa dos pontos característicos de um objeto. Esta escolha é feita utilizando o conceito de imagens integrais [VJ01] cujo conceito básico é que cada pixel x na imagem inicial é a soma dos valores dos pixeis no retângulo formado pela origem da imagem e as coordenadas do pixel atual:

$$I_{\Sigma}(x) = \sum_{i \leq x}^{i=0} \sum_{j \leq y}^{j=0} I(i,j) \tag{2.2}$$

A vantagem destas imagens integrais é que são necessárias apenas adições para calcular a soma das intensidades em qualquer área retangular vertical.

Os pontos característicos são encontrados onde o determinante de uma matriz hessiana é máxima.

2.3.3 Geometric Hashing

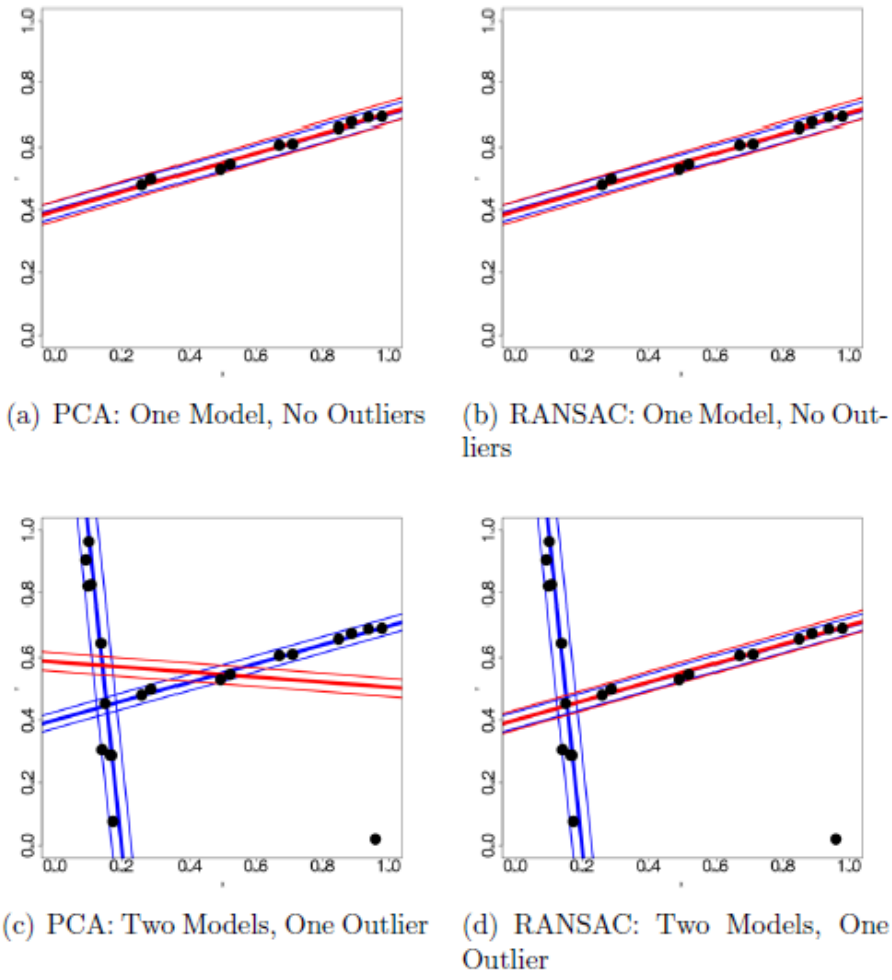
O *geometric hashing*, tal como os métodos acima representados é uma técnica baseada em modelos pré-existentes que visa reconhecer objetos onde são aplicadas rotações, translações e escala [CHPS89].

Esta técnica tem por base também pontos característicos que são obtidos por conjuntos de três pontos não colineares segundo os quais os outros são achados. Desta forma os seus pontos característicos não variam de acordo com as transformações que possam ser aplicadas aos objetos.

2.3.4 RANSAC

RANSAC[FB81], um acrónimo de *RANdom SAmple Consensus*, é um algoritmo que permite extrair, através de um conjunto de dados, os parâmetros do modelo matemático que compõe as características aproximadas do objeto. Este algoritmo funciona de forma iterativa, sendo que a cada iteração melhora a qualidade dos parâmetros extraídos.

Este método representa uma evolução significativa dos métodos mínimos quadrados visto ser permeável a desvio dos dados sem que estes afetem a qualidade da modelação matemática. Como se pode ver na figura 2.3 o método *RANSAC* será um método melhor para uma situação real onde os dados têm muito ruído.



(a) PCA: One Model, No Outliers

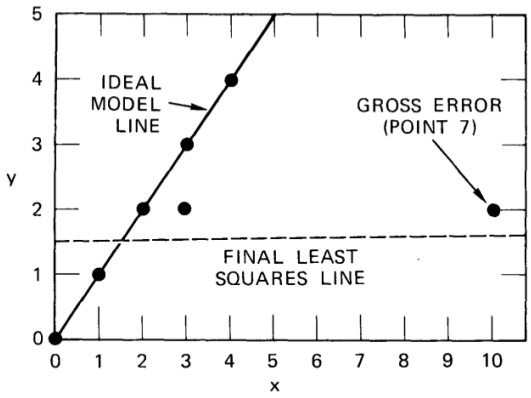
(b) RANSAC: One Model, No Outliers

(c) PCA: Two Models, One Outlier

(d) RANSAC: Two Models, One Outlier

Figura 2.3: Comparação com o método de mínimos quadrados [FB81]

PROBLEM: Given the set of seven (x,y) pairs shown in the plot, find a best fit line, assuming that no valid datum deviates from this line by more than 0.8 units.



POINT	x	y
1	0	0
2	1	1
3	2	2
4	3	2
5	3	3
6	4	4
7	10	2

Figura 2.4: Exemplo de fraqueza do método de mínimos quadrados [FB81]

2.3.5 Point Cloud Library

Point Cloud Library (PCL) é um projeto *opensource* desenvolvido em C++ cujo objetivo é disponibilizar uma ferramenta altamente otimizada para capturar, manipular, visualizar e processar nuvens de pontos. Nuvens de pontos são a informação 3D que sensores como o *Kinect* (2.2.3) devolvem, que não é mais do que a nuvem dos pontos captados no espaço, com a origem no ponto onde o sensor se encontra, também com a informação das cores dos pontos. O projeto PCL [RC] permite que através da nuvem de pontos se extraia a informação desejada, sendo que já tem implementado um conjunto de filtros, técnicas de reconstrução de superfícies, métodos de extração de características 3D (por exemplo normais às superfícies), que a tornam uma ferramenta de muito interessante para usar a par do *Kinect*.

A captura de imagens é bastante facilitada com esta ferramenta, que disponibiliza um formato para que toda a informação seja guardada convenientemente: *.pcd*. Estes ficheiros podem ser visualizados facilmente recorrendo a bibliotecas de visualização da PCL onde são apresentados em 3D, sendo que podem conter só informação de profundidade como na figura 2.5, mas também com toda a informação capturada pelo *Kinect* como pode ser visto na figura 2.6.

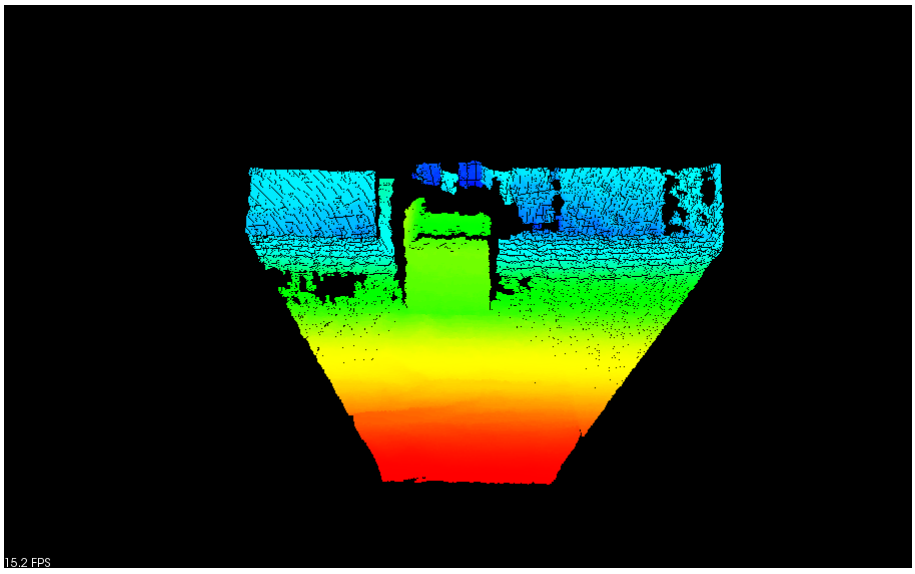


Figura 2.5: Exemplo de imagem capturada só com a informação de profundidade

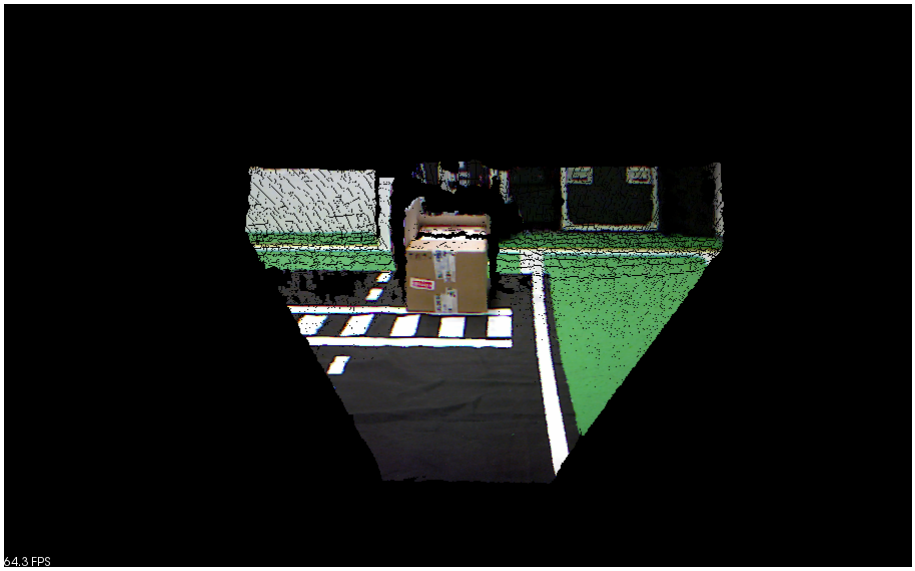


Figura 2.6: Exemplo de imagem capturada com informação RGBD

2.4 Algoritmos de Clustering

Tendo-se já explorado algoritmos de identificação de objetos e os sensores, é conveniente também uma abordagem aos algoritmos de *clustering*, isto porque numa dada imagem em 3D, o que existe são uma série de pontos distribuídos no espaço com a informação de cor, sendo então necessário distinguir quais pertencem a um dado objeto e não a outro.

As técnicas de *clustering* permite que se distribua os pontos por uma série de grupos de acordo com uma série de características dos mesmo e de algumas pré-condições fornecidas.

São expostas nesta secção algumas técnicas de *clustering*, nomeadamente as que apresentam mais vantagens e conveniência para esta dissertação.

2.4.1 Clustering por Vizinhança Euclideana

Este algoritmo simples implica que seja fornecido um número mínimo de pontos e uma distância máxima para que seja definida uma vizinhança e daí a pertença ou não a um *cluster*. A simplicidade deste algoritmo permite que seja facilmente implementado, contudo é muito sensível aos parâmetros iniciais. Esta técnica aliada a uma pré-organização dos pontos, como, por exemplo, numa árvore binária do tipo k-d tree[Ben75] é um algoritmo de *clustering* facilita ainda mais o processo de *clusterização*.

2.5 Demonstradores de Robótica Autónoma

Esta secção refere-se a demonstradores de robótica autónoma, onde são referidos os exemplos que traduzem o que se está a fazer no âmbito de demonstradores de robótica autónoma e que representam o estado da arte nesta área.

2.5.1 DARPA: Grand Challenge

A agência norte americana DARPA (*Defense Advanced Research Agency*), cujos projetos de investigação se destinam principalmente a aplicações militares, realizou três grandes eventos onde foram postos à prova as técnicas de condução autónoma de veículos comerciais devidamente equipados e modificados para se poderem mover de uma forma completamente autónoma.

Existiram duas edições do *Grand Challenge* realizadas em 2004 e 2005 que consistia numa prova de condução autónoma em que os carros percorriam uma estrada de cerca de 242km no deserto do *Mojave*. Em 2004 nenhum dos concorrentes chegou ao final da prova, sendo que o robô que mais distância percorreu ficou pelos 18km.

Em 2007 foi realizado um *Urban Challenge* onde se aproximou as provas às condições encontradas num ambiente urbano, ou seja, estradas com carros a circular em ambas as vias, cruzamentos, sinalização vertical e semáforos.

Na edição de 2007 os veículos autónomos já se encontravam equipados com um conjunto de sensores, entre os quais se destacam LIDAR, Radares, sonares e infravermelhos. O projeto vencedor desenvolvido pela Universidade de *Carnegie Mellon* e apelidado de *Boss* [UAB+08] possuía 18 sensores dispostos como apresentado no diagrama abaixo:

Sigla	Tipo de Equipamento	Modelo
APLX	GPS	Applanix POS-LV 220/420 GPS/IMU
LMS	LIDAR	SICK LMS 291-S05/S14 LIDAR
HDL	LIDAR	Velodyne HDL-64 LIDAR
ISF	LIDAR	Continental ISF 172 LIDAR
XT	LIDAR	IBEO Alasca XT LIDAR
ARS	RADAR	Continental ARS 300 Radar
PGF	Câmara HDR	Point Grey Firefly

Tabela 2.1: Listagem dos sensores do *Boss*

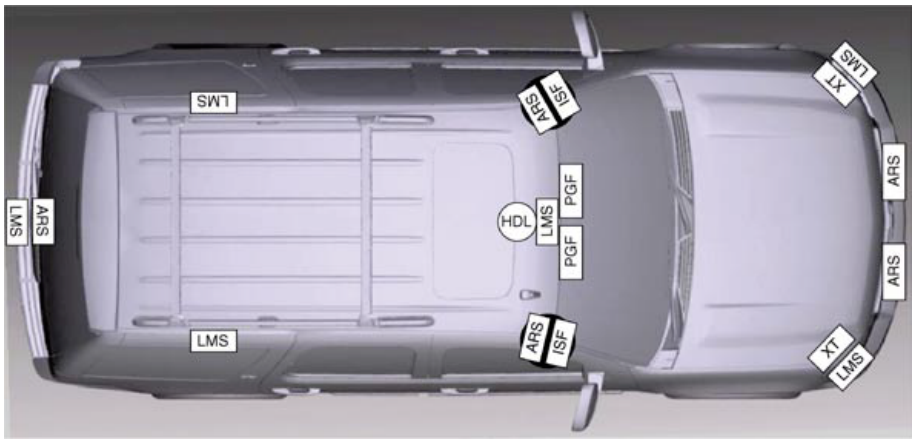


Figura 2.7: Sensores no Boss [UAB+08].

Sendo que os códigos dos sensores correspondem ao indicado na tabela 2.1 pode-se concluir, que os sensores LIDAR são um excelente sensor para ajudar no reconhecimento de objetos no mundo, tal como para mapear o ambiente do robô para se poder orientar de uma forma eficaz.

2.5.2 Micro Rato

O Micro Rato é uma competição criada pela Universidade de Aveiro, onde pequenos robôs de tamanho máximo 300x300x400mm têm de navegar num labirinto. A competição é dividida em duas partes, sendo que o objetivo da primeira é que os robôs encontrem um caminho para uma zona onde existe um farol que emite infravermelhos, e na segunda parte fazerem o caminho de volta para a zona de partida utilizando a informação coletada durante a primeira parte da competição.

2.5.3 Festival Nacional de Robótica

O festival nacional de robótica é um evento organizado anualmente pela sociedade portuguesa de robótica em cidades diferentes onde, além de um encontro de científico

onde investigadores de robótica de todo o mundo discutem e apresentam os trabalhos que estão a desenvolver, são realizadas várias competições e demonstrações de robótica.

As competições que se realizam são as seguintes:

- Busca e Salvamento Júnior RoboCup
- Dança Júnior RoboCup
- Futebol Robótico Júnior RoboCup
- Condução Autónoma
- Liga INFAIMON Futebol Robótico Médio RoboCup
- FreeBots
- Robot@Factory
- Equipas
- Qualificações para o RoboCup

Destas competições, a mais relevante para o trabalho a ser desenvolvido no âmbito desta dissertação é a de Competição Autónoma, onde um robô totalmente autónomo tem de navegar numa pista em oito que tem as características apresentadas:

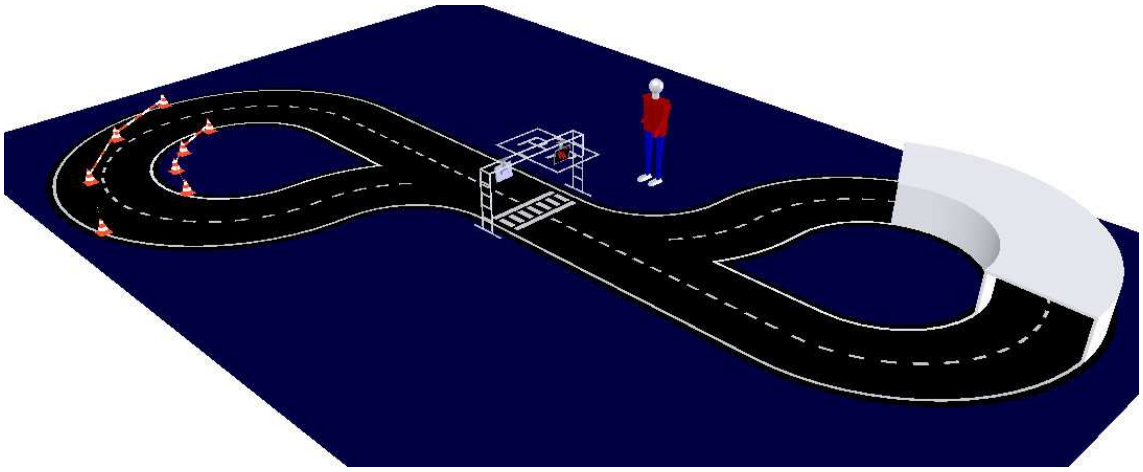


Figura 2.8: Pista de Condução Autónoma no Festival Nacional de Robótica

O objetivo é o robô percorrer a pista circulando pela faixa da direita, seguindo as indicações no sinais verticais que se situam à beira da pista e os semáforos, e evitar os obstáculos que sinalizam obras no percurso.

2.5.4 MINERVA

O minerva é um robô autónomo desenvolvido na universidade Carnegie Mellon, para fazer de robô guia no museu Smithsonian para a exposição de história natural que esteve em exibição no período de 25 de Agosto a 5 de Setembro de 1998.

Figura 2.9: MINERVA - Robô guia do museu Smithsonian

Este robô tornava-se totalmente autónomo a partir do momento em que fazia uma volta de aprendizagem, em que era guiado pelo percurso que lhe estava destinado. Estando a aprendizagem concluída, o robô orientava-se pelo resultado da sua aprendizagem, e por alguns sensores, nomeadamente para se desviar dos visitantes e de obstáculos, uma câmara para detetar marcadores no teto do museu para se conseguir localizar. Além de guiar os visitantes pelo percurso este robô também interagia com os mesmos, respondendo a perguntas e apresentando a exposição.

2.5.5 CleanRob

O CleanRob é um projeto que começou a ser desenvolvido na Faculdade de Engenharia da Universidade do Porto em 2004, no no contexto do Departamento de Engenharia Eletrotécnica e de Computadores. Este robô foi desenvolvido por alunos de modo a envolver alunos no desenvolvimento de projetos académicos com maior aplicação prática tirando partido de técnicas que representam o estado da arte na robótica.

Este robô utiliza um conjunto de câmaras e sonares PSD para fazer a sua localização, de modo a limpar corredores do departamento de Engenharia Eletrotécnica.

Figura 2.10: CleanRob

2.6 Resumo

No que diz respeito a deteção de objetos, existem vários algoritmos bastante robustos e eficazes, contudo são muito voltados para reconhecimento em imagens RGB e não tanto baseados em conjuntos de dados 3D e além disso exigem uma fase de aprendizagem.

Os demonstradores autónomos que existem no momento já são bastante completos, considere-se como exemplo o *Boss* que navega em ambiente bastante próximo do urbano com proeza, estando cada vez mais próximo de um cenário onde condução autónoma nas cidades poderia ser uma realidade e até mesmo uma mais-valia em termos de segurança. É de assinalar também os esforços nas competições de robôs em escalas menores pois com menos recursos e menor escala consegue-se testar técnicas inovadoras com soluções menos dispendiosas obtendo-se resultados igualmente impressionantes.

Capítulo 3

Descrição do Problema

3.1 Reconhecimento de Objetos

O objetivo desta dissertação é o reconhecimento de objetos através das perceções de um robô, tirando o máximo partido dos sensores 3D que recentemente apareceram no mercado. Utilizando tecnologia já desenvolvida, como os sensores 3D e bibliotecas informáticas, será desenvolvida uma aplicação onde se faça o reconhecimento de objetos simples, nomeadamente planos, paralelepípedo e cilindros, para se utilizar no reconhecimento de objetos mais complexos dos quais estes fazem parte.

Visto que o objetivo último é a integração no software de um demonstrador de robótica autónoma, um objetivo secundário é que o reconhecimento seja feito num curto espaço de tempo para maximizar a performance do demonstrador.

O trabalho desenvolvido também poderá ser utilizado no demonstrador autónomo para resolver o problema de localização, isto é: como é que um robô autónomo sabe a sua localização no espaço. Havendo reconhecimento de objetos permitirá que o robô utilize a posição dos objetos reconhecidos para mais facilmente se localizar no espaço. É de salientar que este não é o objetivo principal deste trabalho, contudo com algumas melhorias poder-se-ia tornar esta possibilidade num facto.

Para resolver o problema do reconhecimento de objetos em 3D decidiu-se seguir uma estratégia *bottom-up* tratando primeiro do reconhecimento de objetos simples, nomeadamente planos no espaço, cilindros e paralelepípedo, e de seguida reconhecer mesas em cuja constituição se encontram esses objetos. A metodologia será de seguir por uma fusão sensorial de modo a combinar toda a informação disponível para culminar numa avaliação precisa dos objetos nas imagens RGBD.

O sensor utilizado para a captura de informação 3D foi o *Kinect* da *Microsoft* que é o mais comum no mercado e também foi o que a comunidade *opensource* mais rapidamente

adotou, desenvolvendo controladores e bibliotecas que permitem que o desenvolvimento seja multi-plataforma.

3.1.1 Reconhecimento de Objetos Simples

No que diz respeito a objetos simples optou-se pelos seguintes que são os constituintes mais comuns do principal caso de estudo dos objetos complexos:

- Cilindros;
- Paralelepípedos;
- Planos;

O reconhecimento destes objetos mais simples terá associado um valor de confiança para se poder avaliar a qualidade do método, contudo é preciso ter em atenção o custo em termos de tempo de processamento e comparar as vantagens que estes reconhecimentos intermédios trazem.

3.1.2 Reconhecimento de Objetos Complexos

Para focar os esforços de desenvolvimento definiu-se a *mesa* como prova de conceito para os objetos complexos. Esta escolha apoia-se no facto destes objetos relativamente simples terem propriedades interessantes para o campo da robótica autónoma.

Um desses fatores é o facto de ser um objeto bastante comum que pode ser encontrado em qualquer ambiente.

Outro facto é que, sendo a mesa um objeto relativamente simples, a sua morfologia pode ser muito diferente de caso para caso. Podemos ter mesas com um só apoio central, outras poderão ter três ou quatro apoios, e estes apoios podem ter a forma de cilindros, paralelepípedos ou mesmo até formas menos regulares. Os tampos da mesa podem também ter formas bastante diferentes: desde tampos retangulares, circulares, ovais ou mesmo de formas menos regulares à semelhança dos seus apoios.

Existirá também um dicionário de mesas conhecidas, sobre as quais será produzido um fator de confiança de as ter detetado que é calculado a partir do fator de confiança de se ter encontrado os seus constituintes.

3.2 Resumo e Conclusões

Esta dissertação fará então o reconhecimento de mesas, através do reconhecimentos dos seus constituintes mais simples, i.e.: planos horizontais com várias formas e cilindros e paralelepípedos e indicará a certeza que tem de que é realmente uma mesa.

Capítulo 4

Implementação

A ideia principal do projeto seria fazer a deteção de objetos simples e depois capitalizar nessa capacidade desenvolvida para fazer a deteção de objetos mais complexos. Esta deteção de objetos, que culminou na escolha da mesa como caso de estudo, pelas considerações que já foram explanadas da sua morfologia, tem como objetivo último a sua utilização num demonstrador de robótica autónoma, fazendo recurso a um dicionário pré-existente em que as mesas conhecidas são listadas num ficheiro *XML*.

O programa desenvolvido foi escrito em *C++* recorrendo a algumas bibliotecas informáticas, nomeadamente a *Point Cloud Library* (PCL) para a captura e manipulação de nuvens de pontos e a *Boost* porque além de ser um requisito da PCL tem uma miríade de bibliotecas auxiliares implementadas que potenciam o desenvolvimento e trazem uma garantia de terem a melhor performance possível. De seguida são analisados com mais detalhe os pormenores de implementação.

4.1 Captura de Imagens 3D para análise

A captura de imagens em 3D foi desenvolvida utilizando a biblioteca *Point Cloud Library*, e é uma das capacidades do projeto associado a esta dissertação. As nuvens de pontos são guardadas em ficheiros PCD e guardam o que é capturado pelo *Kinect*, sendo que contém toda a informação RGBD e pode ser visualizada com a mesma aplicação que as gerou.

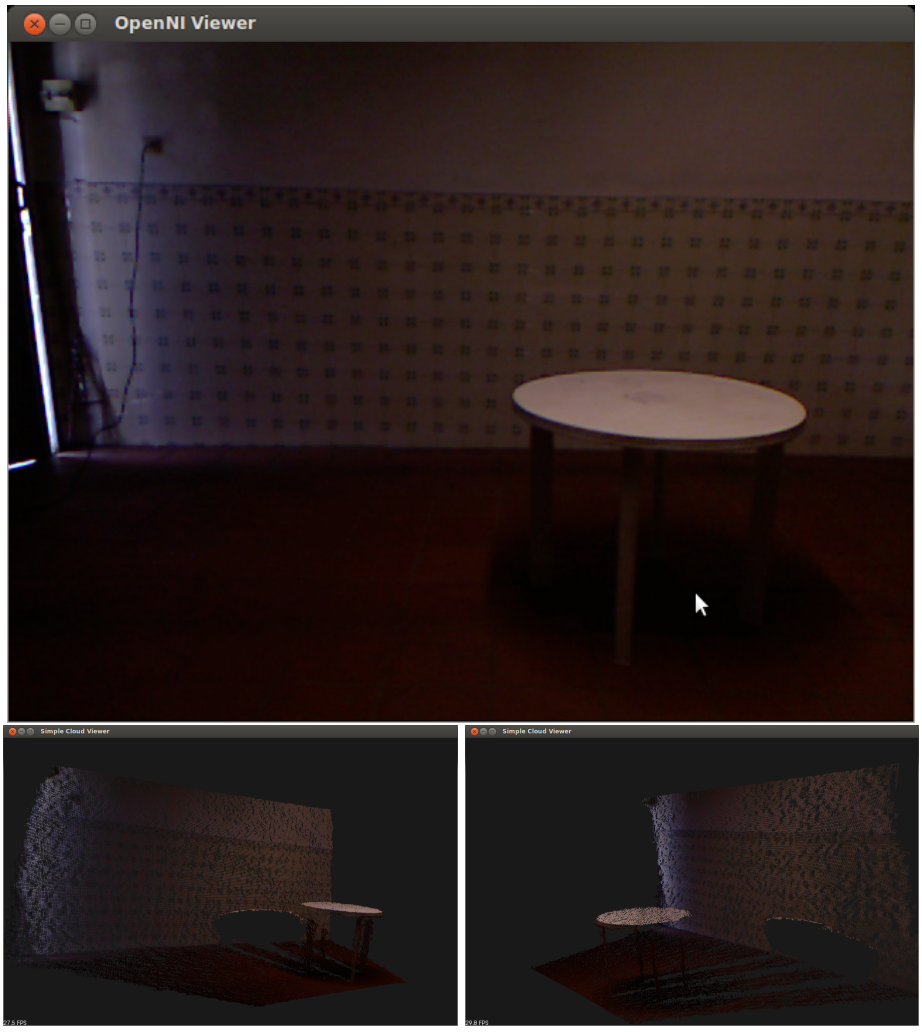


Figura 4.1: Exemplo de imagem capturada com informação RGBD (duas perspetivas) e a imagem RGB correspondente.

4.2 Estudo de Precisão do Kinect

Antes de se passar à implementação da deteção, considerou-se necessário fazer um estudo da precisão das medições do Kinect de modo a garantir que as análises das imagens, e qualquer consideração prévia, estariam corretas e não se baseavam em suposições.

Sendo assim escolheu-se um objeto e mudou-se a sua posição relativa ao *Kinect*. Em cada uma das posições registadas, foram medidas e guardadas as distâncias e os ângulos, sendo que de seguida foram registadas doze imagens RGBD por posição e submeteu-se cada uma a uma análise automatizada para medir a distância e o ângulo ao objeto.

A sequencia dos registos encontra-se na tabela 4.1 que se encontra abaixo.

A posição da mesa relativamente ao *Kinect* para cada posição pode ser verificado na figura 4.2.

Nº da Posição	Distância (m)	Ângulo (°)
1	1.35	0
2	1.34	21
3	2.00	0
4	1.93	14
5	2.30	-27
6	2.75	0
7	2.75	18
8	3.00	-18

Tabela 4.1: Experiências à precisão do Kinect

Figura 4.2: Foto das posições da mesa relativas ao Kinect.

De seguida para cada imagem de cada posição avaliou-se qual a distância do *Kinect* à mesa que a imagem RGBD regista. Os resultados encontram-se registados na tabela 4.2 sendo que para cada uma das posições criou-se um diagrama temporal para avaliar quanto as posições registada pelo *Kinect* se desviam da posição medida fisicamente, que se encontram apresentados nas figuras 4.3 a 4.10.

Para aprofundar a análise, extraiu-se alguns dados estatísticos das medições que se encontram na tabela 4.3

Olhando para o desvio padrão, percebe-se que a variação dos dados do *Kinect* para a posição do objeto não é muito grande, sendo no máximo de 2 centímetros, contudo houve uma variação algo significativa para o valor medido mas não era demasiado grande, sendo que se considerou bastante aceitável.

Quanto aos ângulos, fez-se uma análise semelhante à das distâncias, cujos resultados são os apresentados nas tabelas 4.4 e 4.5.

Comparando o valor da mediana dos valores obtidos pelo *Kinect* aos que foram fisicamente medidos (ver tabela 4.6), consegue-se concluir que a diferença absoluta entre ambos é no máximo 15 graus, o que aliado aos baixo valores do desvio padrão (< 1 grau) é um indicador que o dispositivo é bastante preciso.

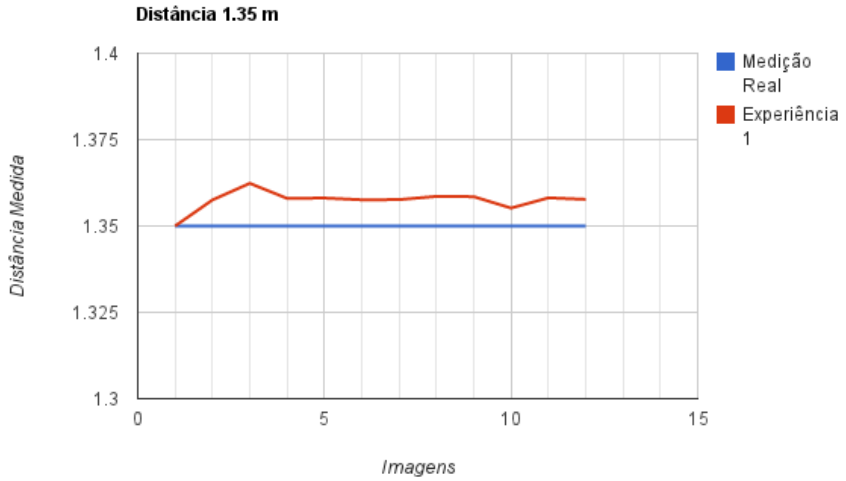


Figura 4.3: Diagrama temporal para a primeira posição (1.35 m)

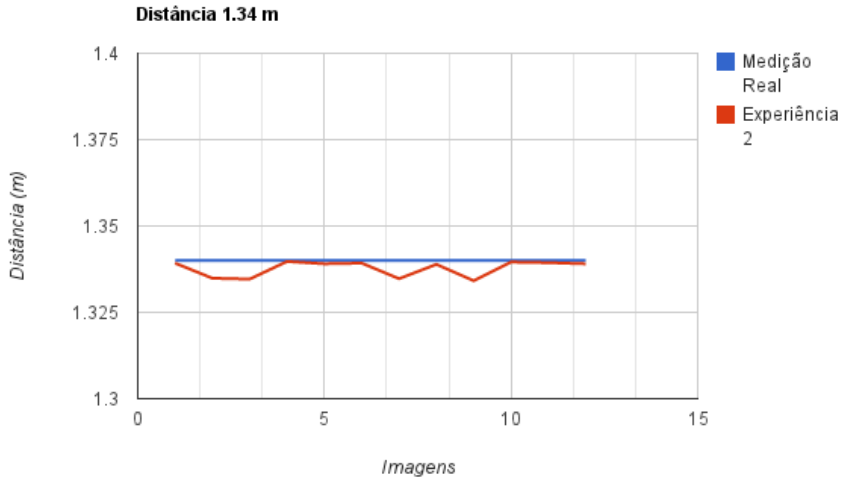


Figura 4.4: Diagrama temporal para a segunda posição (1.34 m)

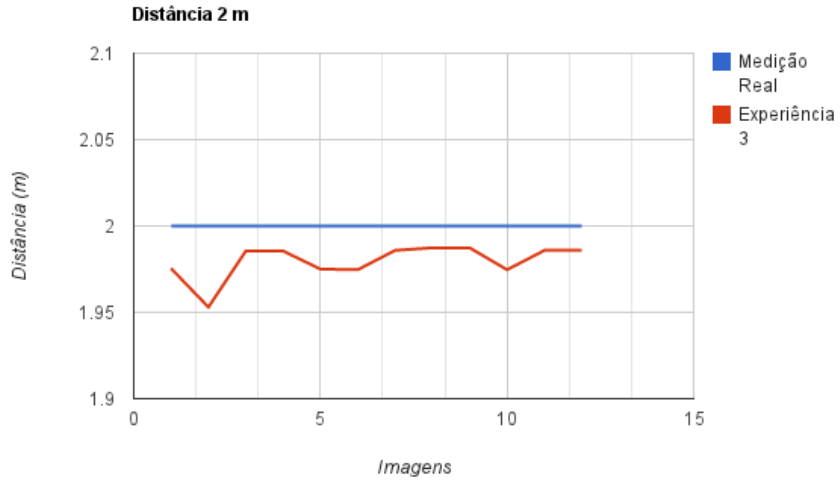


Figura 4.5: Diagrama temporal para a terceira posição (2.00 m)

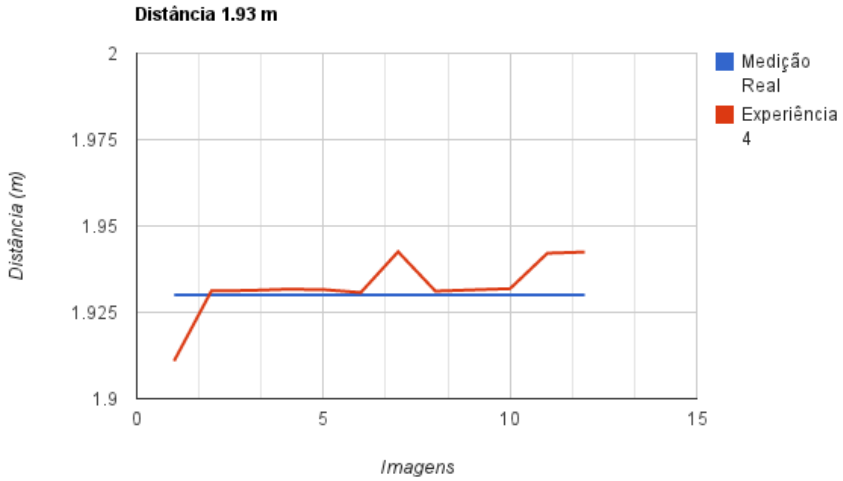


Figura 4.6: Diagrama temporal para a quarta posição (1.93 m)

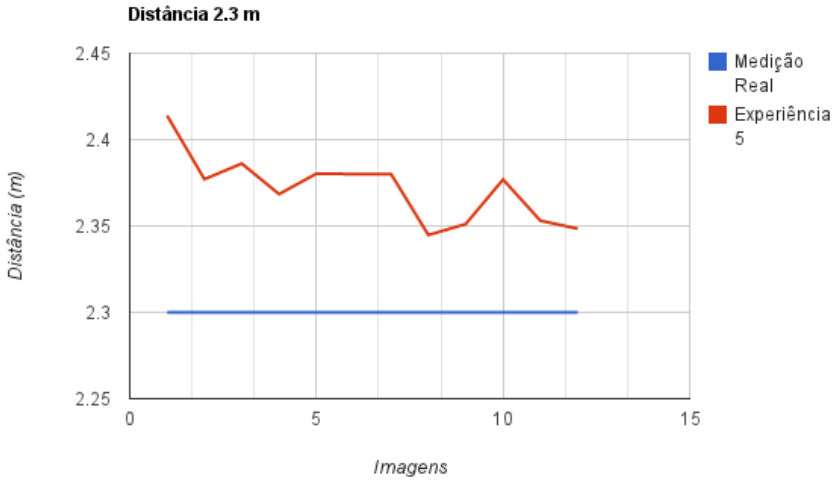


Figura 4.7: Diagrama temporal para a quinta posição (2.30 m)

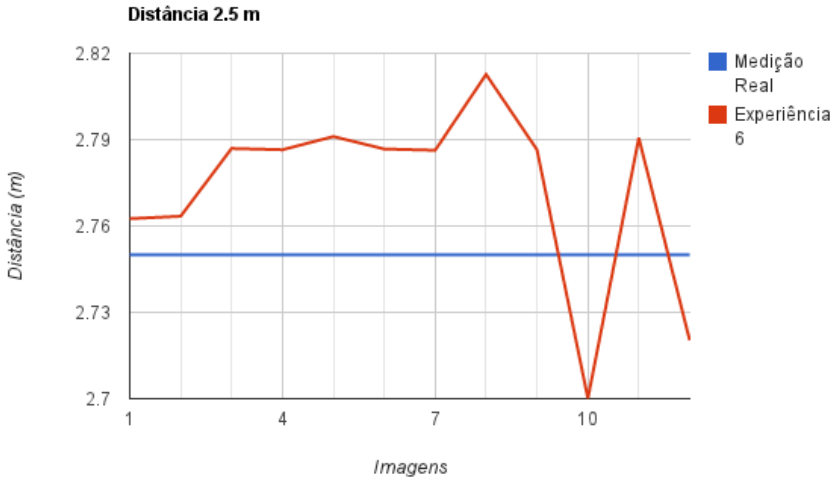


Figura 4.8: Diagrama temporal para a sexta posição (2.75 m)

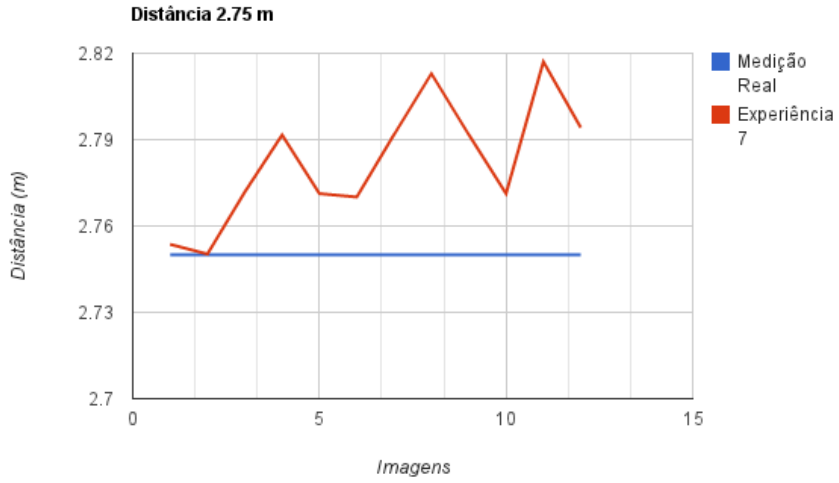


Figura 4.9: Diagrama temporal para a sétima posição (2.75 m)

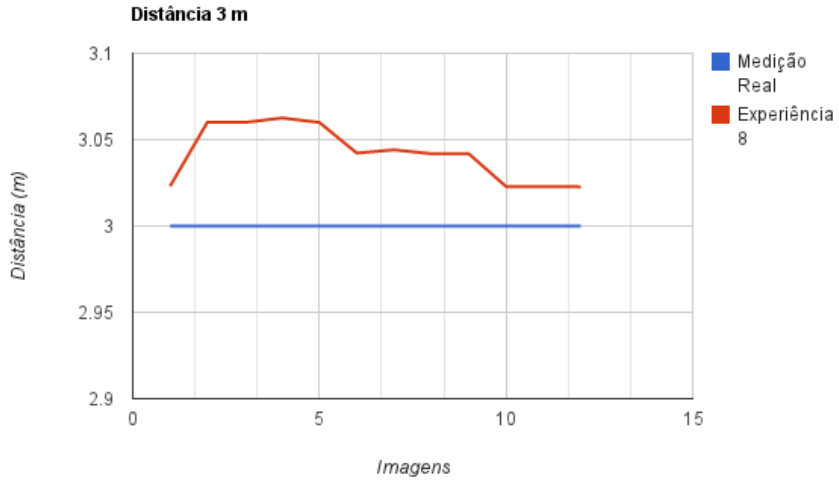


Figura 4.10: Diagrama temporal para a oitava posição (3.00 m)

Posições (m)							
1	2	3	4	5	6	7	8
1.35	1.339247	1.975414	1.910938	2.413945	2.762543	2.753629	3.023028
1.35754	1.334866	1.952989	1.931228	2.377091	2.76339	2.750247	3.060131
1.362351	1.334632	1.985518	1.931326	2.38616	2.786931	2.771791	3.059977
1.35799	1.339735	1.985523	1.931682	2.36845	2.786493	2.791635	3.062525
1.358089	1.339071	1.975026	1.931579	2.380317	2.791077	2.771245	3.059977
1.35759	1.339247	1.974677	1.930731	2.380086	2.78675	2.770061	3.042213
1.357634	1.334736	1.985874	1.942546	2.380075	2.786312	2.791803	3.044104
1.358572	1.338895	1.987369	1.931143	2.344773	2.812669	2.813004	3.04185
1.358485	1.334144	1.987345	1.931502	2.351119	2.786493	2.791803	3.04185
1.355191	1.339556	1.974662	1.931858	2.376915	2.700064	2.771245	3.02285
1.358148	1.339383	1.98587	1.942116	2.353092	2.790619	2.817074	3.02285
1.357711	1.339069	1.98587	1.942461	2.348412	2.720287	2.794201	3.022684

Tabela 4.2: Resultados obtidos das distâncias para cada posição

Posição	Média (m)	Desvio Padrão (m)	Mediana (m)
1	1.3574	0.0017	1.3579
2	1.3377	0.0024	1.3391
3	1.9797	0.0104	1.9855
4	1.9324	0.0051	1.9315
5	2.3717	0.0154	2.3770
6	2.7728	0.0336	2.7865
7	2.7823	0.0200	2.7817
8	3.0420	0.0157	3.0420

Tabela 4.3: Dados estatísticos da recolha de distâncias.

Olhando para os resultados, é possível também afirmar que a medição física dos ângulos provavelmente não estaria completamente alinhada com o foco do *Kinect*.

Além disso utilizou-se a informação extraída dos ângulos para comparar, recorrendo a gráficos, a posição real dos objetos com a posição onde o sensor regista o objeto.

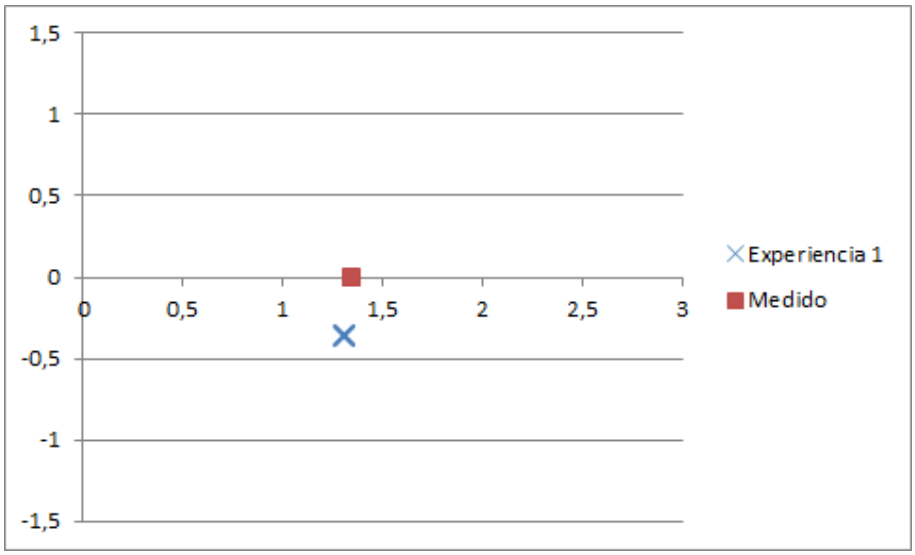


Figura 4.11: Posição real e posições registadas pelo Kinect para a posição 1 (1.35 m)

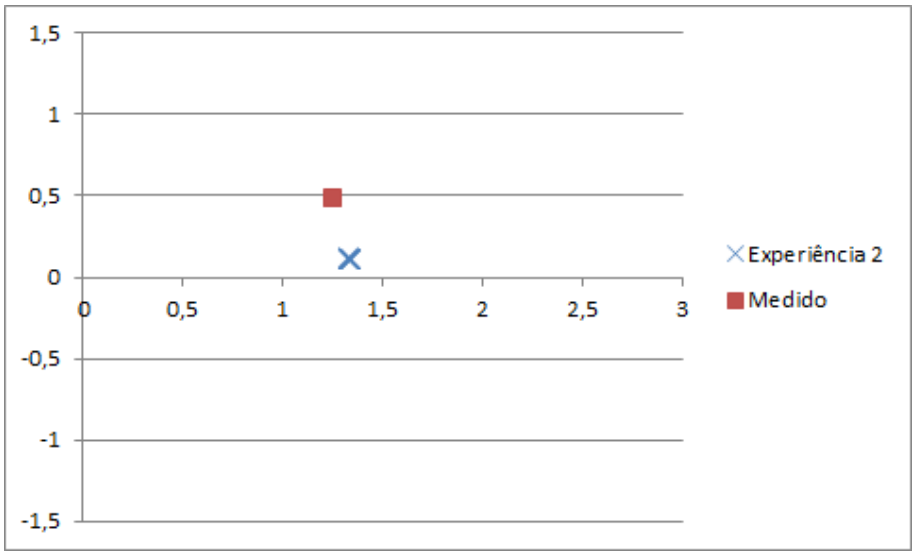


Figura 4.12: Posição real e posições registadas pelo Kinect para a posição 2 (1.34 m)

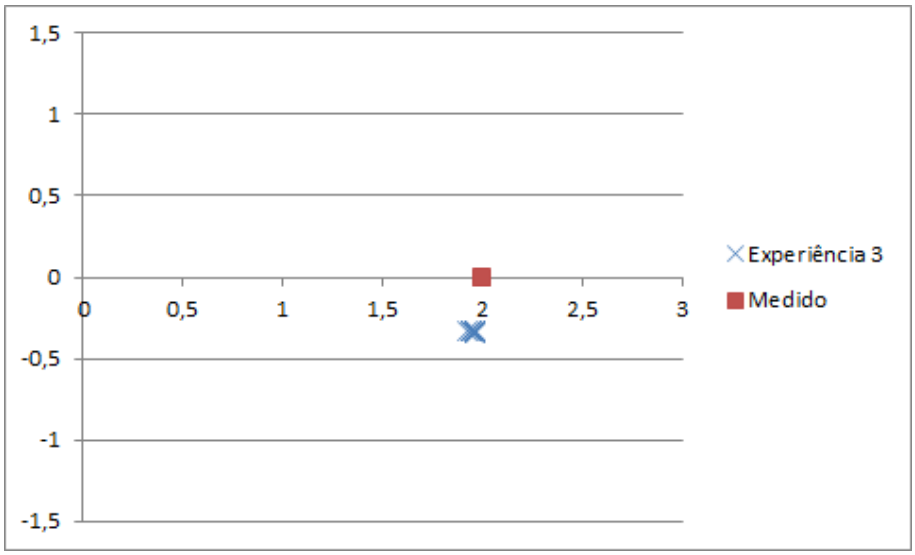


Figura 4.13: Posição real e posições registadas pelo Kinect para a posição 3 (2.00 m)

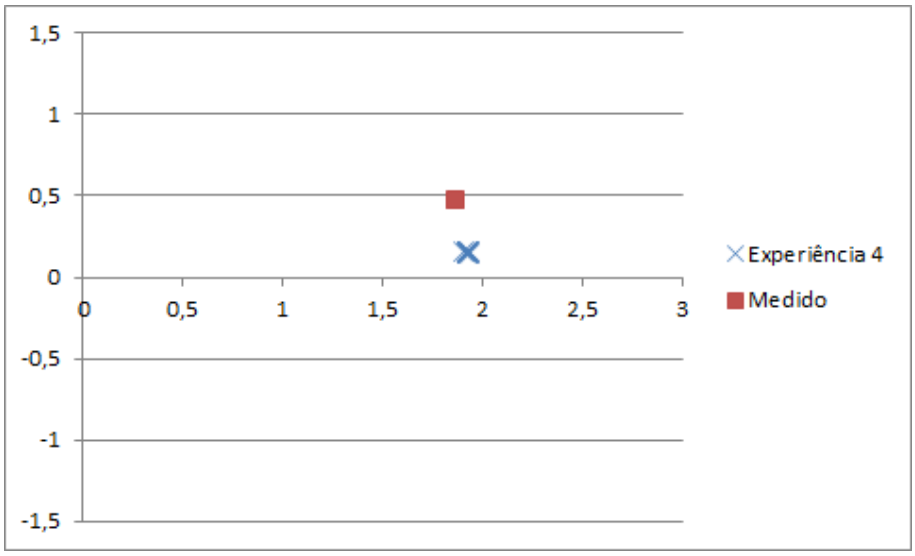


Figura 4.14: Posição real e posições registadas pelo Kinect para a posição 4 (1.93 m)

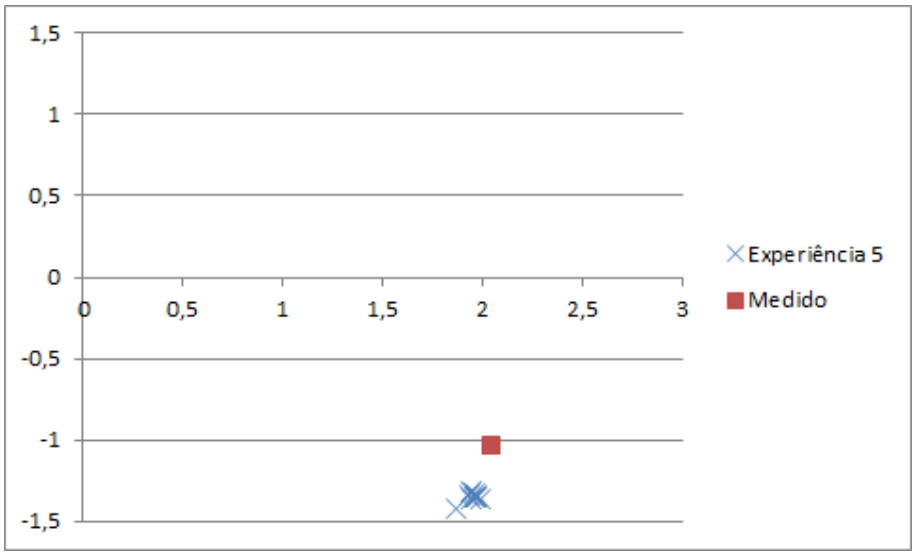


Figura 4.15: Posição real e posições registadas pelo Kinect para a posição 5 (2.30 m)

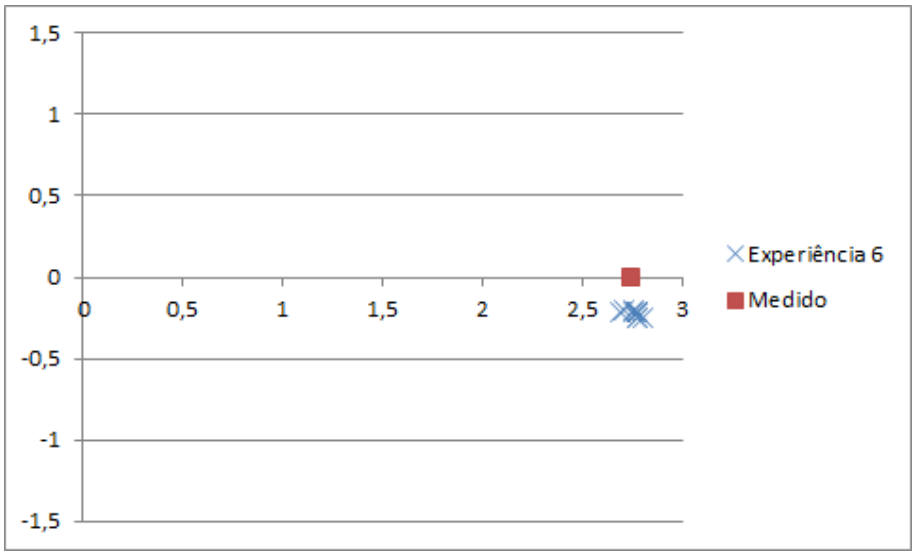


Figura 4.16: Posição real e posições registadas pelo Kinect para a posição 6 (2.75 m)

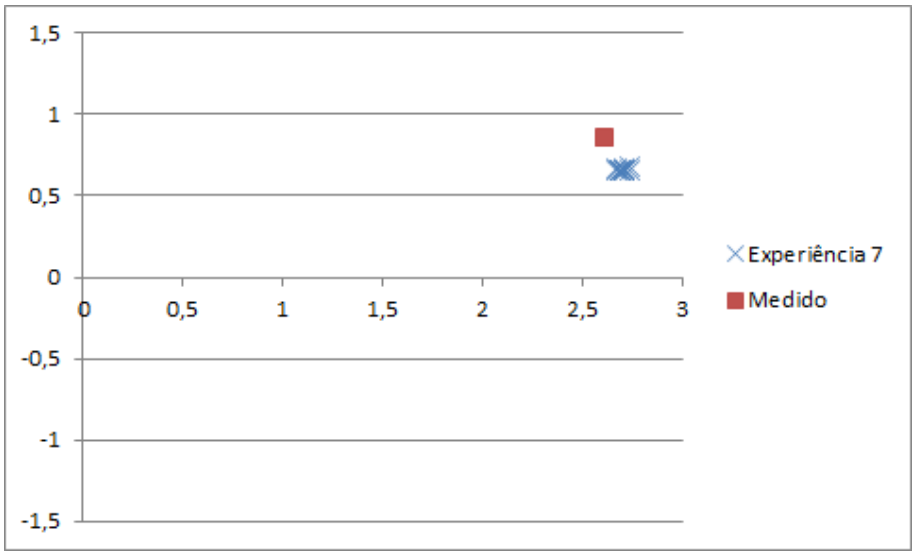


Figura 4.17: Posição real e posições registadas pelo Kinect para a posição 7 (2.75 m)

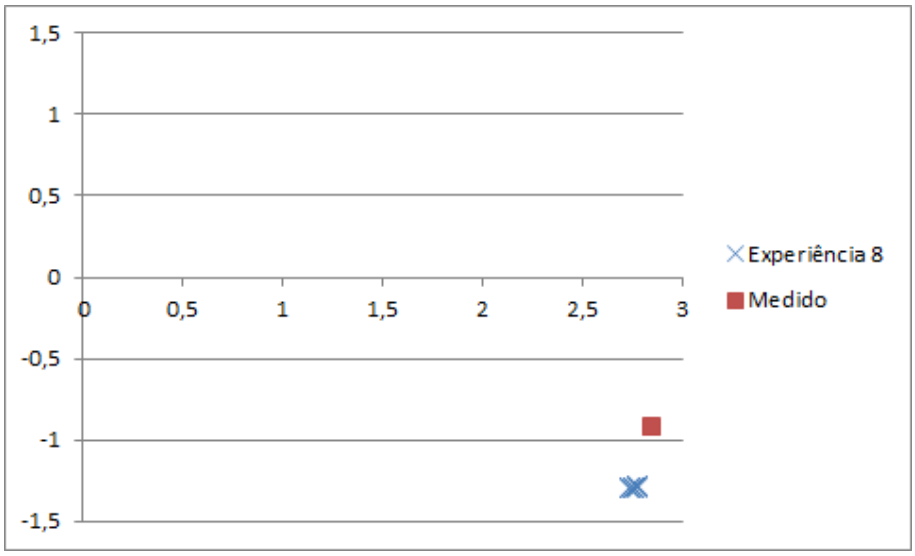


Figura 4.18: Posição real e posições registadas pelo Kinect para a posição 8 (3.00 m)

Ângulos (°)							
1	2	3	4	5	6	7	8
-15.357386	4.792201	-9.630516	4.790509	-34.3685	-4.143964	14.146553	-25.240545
-15.223657	4.976701	-9.608464	4.506991	-34.520763	-4.379865	13.799928	-24.792519
-15.168276	4.835194	-9.448602	4.506991	-34.371037	-4.460659	13.811712	-24.792519
-15.301353	4.932463	-9.449469	4.63898	-34.216621	-4.343729	13.708774	-24.889385
-15.301353	4.700834	-9.563814	4.63898	-34.165874	-5.161427	13.811712	-24.792519
-15.223657	4.792201	-9.501884	4.317291	-34.626232	-4.460659	13.69715	-25.071798
-15.238359	4.994671	-9.510148	4.612736	-34.626232	-4.343729	13.708774	-25.169027
-15.383727	4.584675	-9.765748	4.476339	-37.209705	-5.005285	13.602537	-25.071798
-15.370365	4.740714	-9.761795	4.607438	-34.071758	-4.343729	13.708774	-25.071798
-15.642712	4.926197	-9.501012	4.738488	-34.981014	-4.604316	13.811712	-25.240545
15.318219	4.861241	-9.511069	4.451008	-34.617245	-5.044734	13.94077	-25.240545
-15.250415	4.563166	-9.511069	4.581371	-33.973186	-4.329277	13.935993	-25.240545

Tabela 4.4: Resultados obtidos dos ângulos para cada posição

Posição	Média (°)	Desvio Padrão (°)	Mediana (°)
1	-15.3150	0.1279	-15.3014
2	4.8084	0.1491	4.8137
3	-9.5636	0.1114	-9.5111
4	4.5723	0.1152	4.5944
5	-34.6457	0.8926	-34.4459
6	-4.5518	0.3215	-4.4203
7	13.8070	0.1034	13.8058
8	-25.0511	0.1861	-25.0718

Tabela 4.5: Dados estatísticos da recolha de ângulos.

Posição	Valor Real (°)	Mediana (°)	|Valor Real - Mediana| (°)
1	0	-15.3014	15.3014
2	21	4.8137	16.1863
3	0	-9.5111	9.5111
4	14	4.5944	9.4056
5	-27	-34.4459	7.4459
6	0	-4.4203	4.4203
7	18	13.8058	4.1942
8	-18	-25.0718	7.0718

Tabela 4.6: Comparação do valor da mediana com o valor real dos ângulos.

4.3 Implementação da Deteção

Agora serão listados todos os passos por que passa uma imagem RGBD para se poder detetar a existência de mesas na nuvem de pontos.

Antes de tudo a informação RGB é descartada pois não tem uma grande influência sobre deteção, isto porque após considerações cuidadosas chegou-se à conclusão que a cor não é um aspeto que tenha grande peso na caracterização da mesa.

Adicionalmente o dicionário de mesas conhecidas é lido a partir do ficheiro *dictionary.xml* e guardado em memória para se poder fazer a comparação com o que é encontrado na imagem e dar um valor que avalia a confiança com que pode garantir que é a mesa indicada.

O ficheiro de dicionário, onde se podem descrever as mesas do dicionário, tem a seguinte gramática:

table - Este é o nó principal que contém os nós com as descrições da mesa;

name - Onde é guardado o nome representativo da mesa;

top - Onde estão contidos os atributos do tampo que tem a seguinte estrutura:

shape - Palavra descritiva da forma do tampo da mesa;

large_dimension - A maior distância medida no tampo em metros;

small_dimension - A menor distância medida no tampo em metros;

legs - Onde estão contidos os atributos das pernas da mesa que tem os seguintes parâmetros:

number - O número de pernas que a mesa tem;

height - A altura das pernas em metros;

parallel - Se as pernas são paralelas ou não;

center_distance - distância das pernas ao centro da mesa.

Em suma, as características das mesas são distribuídas entre o tampo da mesa e as pernas, descrevendo a sua morfologia em tanto parâmetros quantitativos como a altura e dimensões dos tampos como em parâmetros qualitativos como a geometria do tampo e se as pernas são paralelas.

4.3.1 Pré Processamento

Agora que se trabalha sobre a informação de profundidade, que se encontra em SI (metros), é feito um pré processamento onde se descarta tudo que aparece na imagem que dista a mais de 4,5 metros do sensor. Isto é no sentido de eliminar erros persistentes na imagem, pois o *Kinect* vai perdendo precisão com a distância [KE12]. No passo seguinte vai-se remover o maior plano que corresponde ao chão, de modo a isolar os objetos que se encontrem na imagem.

Depois é feita uma clusterização usando o algoritmo de vizinhança Euclidiana para separar o que resta da imagem em objetos coerentes e separáveis. Isto, dando um exemplo prático, é para não acontecer o caso de estarmos a reconhecer o que parece uma perna de uma mesa a dois metros de distância do que é reconhecido como um tampo.

No final deste pré processamento o resultado será um conjunto de nuvens de pontos, em que cada uma representa um objeto que se encontra na imagem capturada inicial.

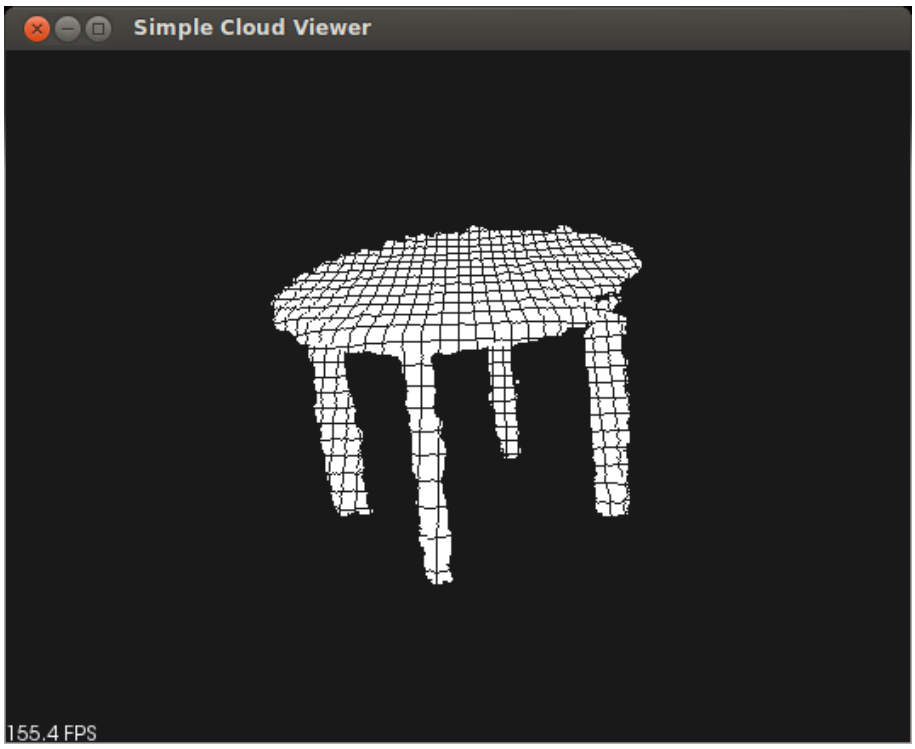


Figura 4.19: Exemplo de imagem após pré processamento.

4.3.2 Deteção de Mesas

De seguida cada um dos objetos separados passa por um processo onde se tenta detetar constituintes das mesas, como pernas e tampos. Caso sejam encontrados verifica-se se correspondem a alguma das mesas conhecidas que estão guardadas no dicionário das mesas conhecidas, retornando um valor de de confiança C que se encontra sempre entre 0 e 1.

Esta deteção é feita nos seguintes passos: primeiro faz-se a deteção do tampo, que é obrigatoriamente uma superfície planar, e caso não tenha um plano é logo atribuída uma confiança de 0, considerando-se que não é uma mesa, pois considera-se que caso exista um plano então existirá um tampo. A deteção do plano é feito recorrendo a um conjunto de métodos da PCL que devolvem o conjunto de pontos que pertencem ao plano e as suas características matemáticas.

Para avaliar a confiança (C_t) de o tampo da mesa ser o do modelo que está no dicionário, é seguida a seguinte formulação para cada uma das características (C_i):

$$C_i = 1 - \frac{|valor\_modelo - valor\_observado|}{valor\_modelo} \tag{4.1}$$

Chegou-se a esta formulação por se considerar, através da experimentação, que seria a que melhor consideraria todas as características para uma melhor qualidade na identificação.

Sendo que se avalia tanto a dimensão maior C_1 como a mais pequena C_2 e são ambas igualmente relevantes para a avaliação do ajuste da mesa observada vão ter ambas um peso de 0.5 na avaliação do tampo, portanto o valor da confiança C_t é calculado através da seguinte fórmula:

$$C_t = 0.5C_1 + 0.5C_2 \tag{4.2}$$

As dimensões do plano encontrado são avaliadas e comparadas com cada um dos modelos, sendo que a metodologia utilizada para a sua obtenção foi utilizar métodos da PCL para extrair os pontos máximos e mínimos do *cluster* por ele formado e depois utilizá-los para calcular as ditas dimensões.

De seguida é feita uma análise aos apoios. Estes apoios são obtidos através da remoção dos pontos que fazem parte do tampo, restando apenas os pontos que lhes pertencem, sendo de seguida separados utilizando um algoritmo de *clustering*. A confiança dos apoios é representada por C_a sendo que tem em consideração o número de apoios observados, e a altura dos mesmos. Cada um destes fatores, respetivamente C_n e C_h, são calculados seguindo a formulação 4.1. Os valores obtidos são pesados onde a C_n é aplicado um fator de 0.4 pois facilmente um dos apoios é oculto nas imagens em 3D portanto tem de ter um peso menor, enquanto que a altura C_h dos apoios tem um peso de 0.6 na avaliação de os apoios corresponderem aos do modelo no dicionário. Em termos de fórmula:

$$C_a = 0.4C_n + 0.6C_h \tag{4.3}$$

Com base no que foi detetado o valor de confiança global é calculado, distribuindo a confiança igualmente pelo que foi calculado quanto ao tampo C_t e e quanto aos apoios C_a sendo que a formulação final fica:

$$C = 0.5C_t + 0.5C_a \tag{4.4}$$

De seguida analisa-se os resultados que são produzidos pela metodologia descrita, contudo todo este processo encontra-se esquematizado na figura 4.20.

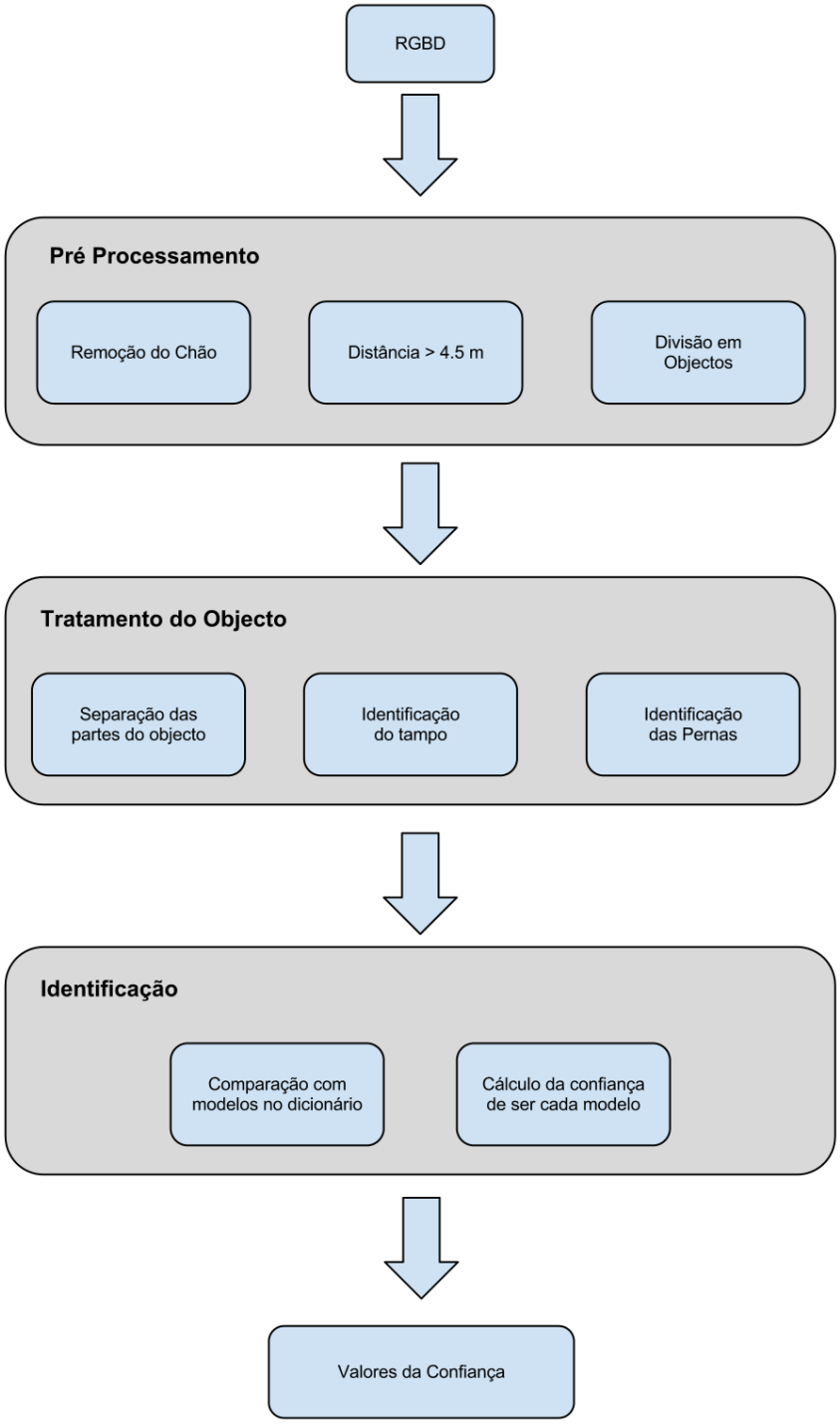


Figura 4.20: Funcionamento do processo global.

4.4 Resultados Obtidos

Para se obter resultados foram analisadas três mesas diferentes que são descritas no seguinte ficheiro de dicionário em XML.

dictionary.xml

```
<?xml version="1.0" ?>
<table>
<name>Mesa 1</name>
<top shape="round" large_dimension="0.89" small_dimension="0.89"/>
<legs  number="4" height="0.68" parallel="true"/>
</table>
<table>
<name>Mesa 2</name>
<top shape="squared" large_dimension="1.0" small_dimension="1.0"/>
<legs  number="4" height="0.7" parallel="true"/>
</table>
<table>
<name>Mesa 3</name>
<top shape="squared" large_dimension="1.0" small_dimension="1.0"/>
<legs  number="4" height="0.59" parallel="true"/>
</table>
<table>
<name>Mesa 4</name>
<top shape="squared" large_dimension="2.0" small_dimension="1.5"/>
<legs  number="3" height="0.9" parallel="true"/>
</table>
<table>
<name>Mesa 5</name>
<top shape="round" large_dimension="1.5" small_dimension="1.5"/>
<legs  number="1" height="0.8" parallel="true" />
</table>
```

A primeira mesa, apresentada na figura 4.21, como aparece representado no dicionário, é uma mesa redonda, com aproximadamente 90 centímetros de diâmetro e tem quatro apoios com cerca de 70 centímetros de altura.

A segunda mesa a ser identificada é idêntica à primeira mas tem um tampo quadrado de 1 metro de lado, como pode ser observado na figura 4.22.

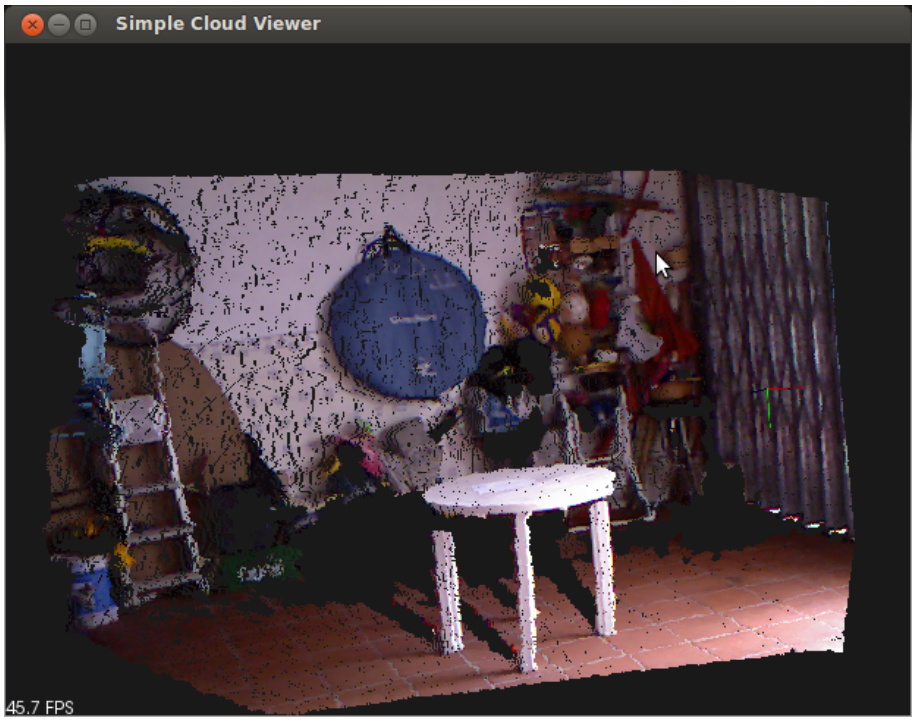


Figura 4.21: Primeira mesa a ser identificada.

A terceira mesa (figura 4.23) é mais baixa com cerca de 60 centímetros de altura e tem quatro apoios. O tampo é quadrado e tem 1 metro de lado.

As mesas 4 e 5 não correspondem a mesas testadas, contudo servem para analisar como é que esta metodologia de reconhecimentos avalia morfologias diferentes.

Comparação das mesas

Característica	Mesa 1	Mesa 2	Mesa 3
Forma	Circular	Quadrada	Quadrada
Dimensão Maior do Tampo	0.89	1.0	1.0
Dimensão Menor do Tampo	0.89	1.0	1.0
Numero de Pernas	4	4	4
Altura	0.68	0.7	0.59
Pernas Paralelas	sim	sim	sim

Tabela 4.7: Comparação das dimensões das mesas analisadas.

De seguida captou-se uma série de imagens RGBD em várias perspetivas dessas mesas para tentar fazer o reconhecimento.

Os resultados da identificação para cada uma das mesas descritas são apresentados abaixo, sendo que antes se apresenta a significância de cada uma das variáveis:

- C_1 - Confiança da dimensão maior;

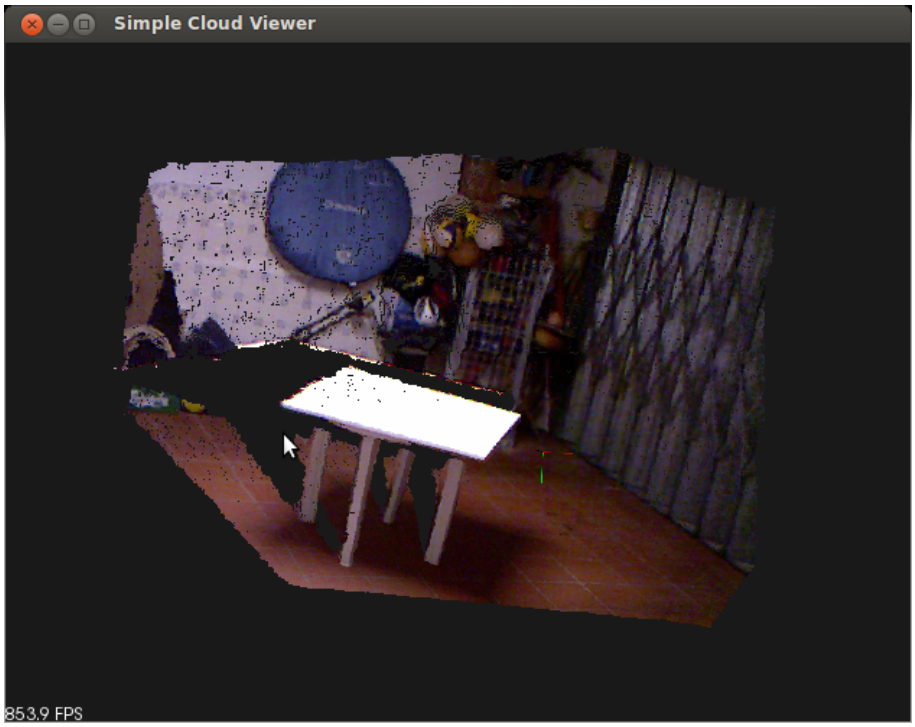


Figura 4.22: Segunda mesa a ser identificada.

- C_2 - Confiança da dimensão menor;
- C_n - Confiança do número de pernas;
- C_h - Confiança da altura das pernas;
- C - Confiança de ter identificado a mesa analisada;

De seguida seguem-se os resultados das experiências feitas, sendo que de uma série de imagens RGBD das três mesas diferentes foram escolhidas três e verificou-se se era possível identificar a mesa correta através da metodologia proposta. É também de notar que as experiências foram realizadas num computador com um processador *Core i7-2630QM* e 8GB de memória.

4.4.1 Identificação

A análise de uma imagem RGBD da Mesa 1 apresenta resultados bastante satisfatórios, onde o modelo do dicionário com um valor de confiança mais alto é o que na realidade a representa nas três experiências. Verificou-se também que existe pouca diferença entre o valor de confiança do modelo da Mesa 1 e o da Mesa 2, contudo poderá ser explicável pelo facto de as duas mesas com morfologias muito aproximadas.

Adicionalmente os *clusters* representativas das mesas encontram-se na figura 4.24.

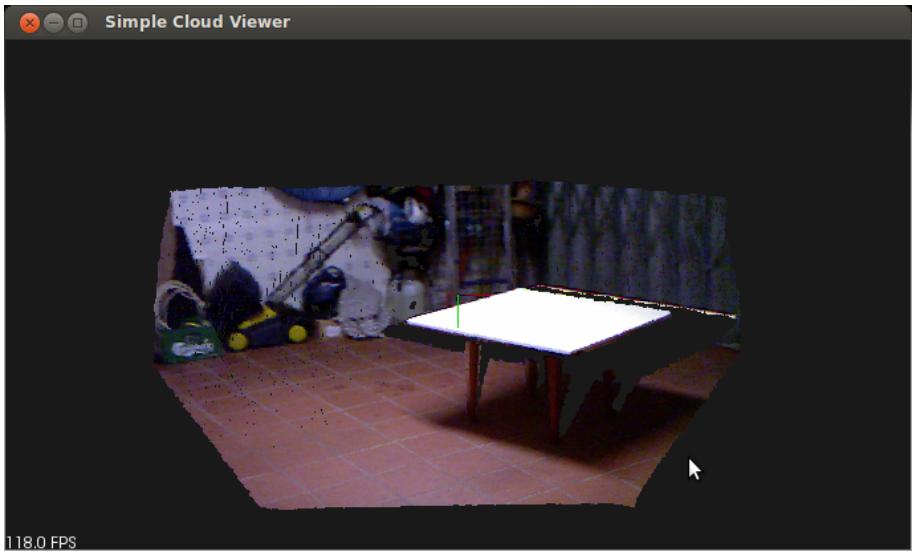


Figura 4.23: Terceira mesa a ser identificada.

Figura 4.24: Clusters representativos da mesa 1.

No reconhecimento desta mesa o resultado já não é tão satisfatório visto que dá uma confiança maior para o modelo da mesa 3, seguido pelo modelo que representa de facto a mesa: mesa 2.

O resultado desta análise 4.10 é também bastante satisfatório pois permite a identificação correta da mesa em análise na maioria dos casos: mesa 3.

Os resultados são satisfatórios, pois pelos dados extraídos pode-se concluir que com uma técnica de identificação bastante simples, apesar de extensível, é possível reconhecer com sucesso uma mesa, cuja morfologia tenha sido previamente guardada. Um dos grandes defeitos é o tempo que o pré-processamento demora, que a ser melhorado permitiria o rápido reconhecimento das mesas criando uma vantagem significativa no reconhecimento de objetos complexos.

Modelo do dicionário	C_1	C_2	C_n	C_h	C
Primeiro Ensaio					
Mesa 1	0.997753	0.987125	1.000000	0.702837	0.907070
Mesa 2	0.892000	0.878541	1.000000	0.682756	0.847462
Mesa 3	0.892000	0.878541	1.000000	0.810049	0.885650
Mesa 4	0.446000	0.585694	0.666667	0.531032	0.550567
Mesa 5	0.594667	0.585694	-2.000000	0.597411	0.074314
Segundo Ensaio					
Mesa 1	0.938588	0.816854	0.750000	0.785464	0.824500
Mesa 2	0.835343	0.727000	0.750000	0.763022	0.769492
Mesa 3	0.835343	0.727000	0.750000	0.905281	0.812170
Mesa 4	0.417671	0.484667	1.000000	0.593462	0.603623
Mesa 5	0.556895	0.484667	-1.000000	0.667645	0.260684
Terceiro Ensaio					
Mesa 1	0.986337	0.946067	1.000000	0.607531	0.865360
Mesa 2	0.902160	0.842000	1.000000	0.590173	0.813092
Mesa 3	0.902160	0.842000	1.000000	0.700205	0.846102
Mesa 4	0.451080	0.561333	0.666667	0.459023	0.524144
Mesa 5	0.601440	0.561333	-2.000000	0.516401	0.045614

Tabela 4.8: Resultados das análises da Mesa 1

Figura 4.25: Clusters representativos da mesa 2.

4.4.2 Tempos de Processamento

Os tempos de processamento são de extrema importância pois avaliam a adequação da solução ao âmbito da robótica autónoma, pois este requer baixos tempos de execução para fazer rapidamente a identificação de objetos.

Como se vê na tabela 4.11 a maior parte do tempo de execução é passado no pré processamento, sendo a comparação com os modelos do dicionário feita de uma forma praticamente instantânea.

Verifica-se a tendência do pré-processamento ser bastante custoso, oscilando aqui entre aproximadamente 40 segundos e 13 segundos, enquanto que a identificação demora

Modelo do dicionário	C_1	C_2	C_n	C_h	C
Primeiro Ensaio					
Mesa 1	0.622630	0.789887	1.000000	0.647027	0.747238
Mesa 2	0.774141	0.923000	1.000000	0.628541	0.812847
Mesa 3	0.774141	0.923000	1.000000	0.745726	0.848003
Mesa 4	0.612930	0.718000	0.666667	0.488865	0.612725
Mesa 5	0.817239	0.718000	-2.000000	0.549973	0.148802
Segundo Ensaio					
Mesa 1	0.434151	0.717978	1.000000	0.789819	0.724978
Mesa 2	0.606394	0.859000	1.000000	0.767252	0.796524
Mesa 3	0.606394	0.859000	1.000000	0.910299	0.839438
Mesa 4	0.696803	0.760667	0.666667	0.596752	0.676726
Mesa 5	0.929070	0.760667	-2.000000	0.671346	0.223838
Terceiro Ensaio					
Mesa 1	0.706658	0.900000	1.000000	0.707300	0.813855
Mesa 2	0.848926	0.979000	1.000000	0.687092	0.863109
Mesa 3	0.848926	0.979000	1.000000	0.815194	0.901540
Mesa 4	0.575537	0.652667	0.666667	0.534405	0.600706
Mesa 5	0.767383	0.652667	-2.000000	0.601205	0.135374

Tabela 4.9: Resultados da análise da Mesa 2

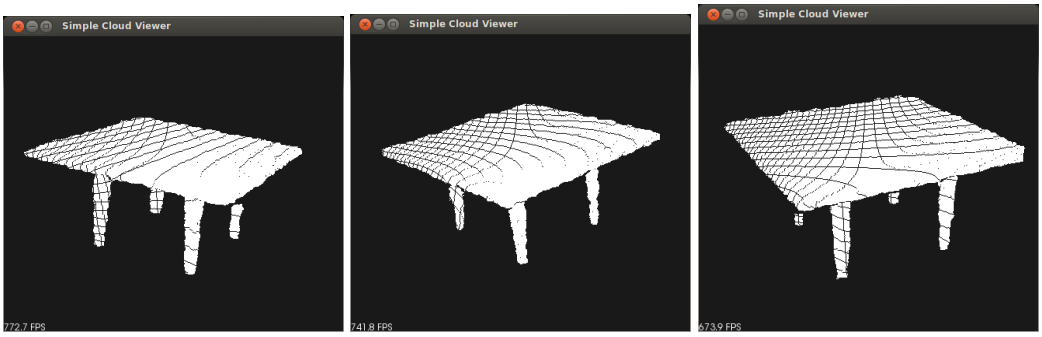


Figura 4.26: Clusters representativos da mesa 3.

sempre aproximadamente 1 segundo.

Nos tempos de processamento para uma imagem RGBD com a Mesa 3 confirma-se que o tempo de pré processamento é bastante significativo, e apesar de absolutamente necessário acaba por atrasar o mais importante da questão que é a identificação do objeto.

Modelo do dicionário	C_1	C_2	C_n	C_h	C
Primeiro Ensaio					
Mesa 1	0.423086	0.664045	1.000000	0.375991	0.584580
Mesa 2	0.596547	0.811000	1.000000	0.365248	0.661461
Mesa 3	0.596547	0.811000	1.000000	0.433345	0.681890
Mesa 4	0.701727	0.792667	0.666667	0.284082	0.592156
Mesa 5	0.935636	0.792667	-2.000000	0.319592	0.127953
Segundo Ensaio					
Mesa 1	0.490549	0.612359	0.750000	0.440277	0.557810
Mesa 2	0.656589	0.765000	0.750000	0.427698	0.633707
Mesa 3	0.656589	0.765000	0.750000	0.507438	0.657629
Mesa 4	0.671706	0.823333	1.000000	0.332654	0.673556
Mesa 5	0.895608	0.823333	-1.000000	0.374236	0.342006
Terceiro Ensaio					
Mesa 1	0.610414	0.776405	1.000000	0.288464	0.633244
Mesa 2	0.763269	0.911000	1.000000	0.280222	0.702634
Mesa 3	0.763269	0.911000	1.000000	0.332467	0.718307
Mesa 4	0.618366	0.726000	0.666667	0.217950	0.534810
Mesa 5	0.824488	0.726000	-2.000000	0.245194	0.061180

Tabela 4.10: Resultados da análise da Mesa 3

Tempos de processamento

Ensaio	Pré processamento (s)	Identificação (s)
1	11.840000	0.41
2	8.530000	0.40
3	18.770000	0.46

Tabela 4.11: Tempos de processamento da análise da Mesa 1

Tempos de processamento

Ensaio	Pré processamento (s)	Identificação (s)
1	39.330000	1.02
2	13.460000	0.83
3	24.620000	0.74

Tabela 4.12: Tempos de processamento da análise da Mesa 2

Tempos de processamento

Ensaio	Pré processamento (s)	Identificação (s)
1	35.510000	0.24
2	39.630000	0.24
3	62.590000	0.36

Tabela 4.13: Tempos de processamento da análise da Mesa 3

Capítulo 5

Conclusões e Trabalho Futuro

A avaliação da precisão do *Kinect* foi um sucesso porque se conseguiu verificar que é bastante preciso ao posicionar um objeto no espaço, ou seja, a identificar a que distância está e que ângulo faz com o sensor.

O objetivo de reconhecimento de mesas foi cumprido sendo que as mesas são identificadas corretamente. O reconhecimento, apesar de ter margem para melhorar, é rápido e consegue detetar uma mesa com poucas características e produz alguns resultados satisfatórios, contudo existem diferenças demasiado pequenas entre a confiança de modelos diferentes.

Quanto aos apoios a ideia inicial seria detetar a sua forma, contudo essa ideia foi simplificada aquando da implementação, pois a PCL permite já fazer a deteção de algumas dessas primitivas mas principalmente porque o processo de identificação é custoso em termos de tempo de execução, atrasando o reconhecimento e não trazendo vantagens significativas.

Uma das limitações encontradas para esta metodologia foi que quando o tampo se encontra alinhado com o sensor em termos de altura, a mesa não é reconhecida, pois postulou-se que a existência de um plano horizontal é obrigatória, sendo que desta forma nenhum plano é encontrado.

Finalmente também é de salientar que é possível fazer o reconhecimento de várias mesas que estejam presentes numa só imagem RGBD, contudo há que ter atenção aos casos onde uma mesa esteja parcialmente oculta por outra podendo ter um impacto muito negativo na confiança de ser o modelo que na realidade a descreve no dicionário.

5.1 Trabalho Futuro

No que diz respeito a trabalho futuro, existem várias melhorias que poderiam ser introduzidas, contudo as mais concretas seriam em dois campos muito particulares: rapidez do pré-processamento e a forma como a avaliação é realizada.

No que diz respeito à rapidez da identificação, tendo em conta que a sua rapidez de execução é um requisito muito importante na robótica autónoma, poder-se-á potenciar a velocidade de clusterização usando bibliotecas de paralelização (como a OpenMP) para avaliar os *clusters*, encontrados na imagem, em paralelo. Outra das hipóteses seria usar, por exemplo, algoritmos genéticos para otimizar os fatores envolvidos na deteção de *clusters* pois é a operação mais custosa em termos de tempo no processo global, principalmente o que é feito no pré-processamento.

Fica também para trabalho futuro a deteção da morfologia das pernas das mesas, principalmente as que são paralelepípedas e cilíndricas, mas certamente contribuiria para melhorar a qualidade da identificação.

Referências

[Ben75] Jon Louis Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, September 1975.

[BTGL06] Herbert Bay, Tinne Tuytelaars, Van Gool e L. SURF: Speeded Up Robust Features. In *9th European Conference on Computer Vision*, Graz Austria, May 2006.

[CHPS89] M. S. Costa, R. M. Haralick, T. I. Phillips e L. G. Shapiro. Optimal affine-invariant point matching. In J. E. Laird, editor, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 1095 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 515–+, March 1989.

[FB81] Martin A. Fischler e Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, June 1981.

[KE12] Kourosh Khoshelham e Sander Oude Elberink. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12(2):1437–1454, 2012.

[Kei10] Mobasser S. Kuang D. Cheng Y. Ivanov T. Johnson A.E. Goldberg H.R. Khanoyan G. Natzic D.B. Keim, J.A. Field test implementation to evaluate a flash lidar as a primary sensor for safe lunar landing. Big Sky, MT, 2010. cited By (since 1996) 0; Conference of 2010 IEEE Aerospace Conference; Conference Date: 6 March 2010 through 13 March 2010; Conference Code: 80381.

[Low99] David G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ICCV '99, pages 1150–, Washington, DC, USA, 1999. IEEE Computer Society.

[RC] Radu Bogdan Rusu e Steve Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*.

[Tan11] Jason Tanz. A thousand points of infrared light. wired magazine. 19.07:117–118, July 2011.

[UAB+08] Chris Urmson, Joshua Anhalt, Drew Bagnell, Christopher Baker, Robert Bittner, M. N. Clark, John Dolan, Dave Duggins, Tugrul Galatali, Chris Geyer,

Michele Gittleman, Sam Harbaugh, Martial Hebert, Thomas M. Howard, Sascha Kolski, Alonzo Kelly, Maxim Likhachev, Matt McNaughton, Nick Miller, Kevin Peterson, Brian Pilnick, Raj Rajkumar, Paul Rybski, Bryan Salesky, Young-Woo Seo, Sanjiv Singh, Jarrod Snider, Anthony Stentz, William "Red" Whittaker, Ziv Wolkowicki, Jason Ziglar, Hong Bae, Thomas Brown, Daniel Demitrish, Bakhtiar Litkouhi, Jim Nickolaou, Varsha Sadekar, Wende Zhang, Joshua Struble, Michael Taylor, Michael Darms e Dave Ferguson. Autonomous driving in urban environments: Boss and the urban challenge. *J. Field Robot.*, 25:425–466, August 2008.

[VJ01] Paul Viola e Michael Jones. Rapid object detection using a boosted cascade of simple features. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 1:511, 2001.