

Sistema de aumento de segurança para cadeira de rodas baseada num sensor Kinect

Joaquim Guilherme Vasconcelos Gonçalves da Silva

Trabalho realizado sob a orientação de

Professor Doutor José Luís Lima

Professor Doutor José Gonçalves

Engenharia Industrial, ramo de Engenharia Eletrotécnica

Sistema de aumento de segurança para cadeira de rodas baseada num sensor Kinect

Relatório de Dissertação
Mestrado em Engenharia Industrial, ramo de Engenharia Eletrotécnica
Escola Superior de Tecnologia e de Gestão

Joaquim Silva
Outubro de 2013

Agradecimentos

Por detrás do trabalho pessoal realizado, existe sempre um grande número de apoios, sugestões, comentários e críticas vindas de muitas pessoas. Sem essas contribuições de extrema importância, seria impossível levar o trabalho realizado ao final com êxito e com os objetivos cumpridos.

Aos Professores Doutores José Luís Lima e José Gonçalves, pela orientação, ajuda e auxílio na realização de todas as fases do desenvolvimento deste projeto, pela disponibilidade em todos os momentos e pelo rigor dos conhecimentos transmitidos.

À minha família, em especial ao meu pai e meus irmãos, pelo apoio e por me terem sido dadas as condições e oportunidade de depois da licenciatura, conseguir obter e concluir o mestrado. Um agradecimento também à minha tia Olimpia e sua família por toda a ajuda ao longo da minha formação académica.

A todos os meus amigos e colegas que conheci durante todos estes anos, que de alguma forma me apoiaram e ajudaram a chegar até aqui, são muitos para serem nomeados, mas eles sabem quem são.

À minha namorada Carolina, que durante este ano sempre me apoiou e acompanhou em todas estas fases, e com as suas palavras e a sua atenção sempre esteve, incondicionalmente, do meu lado.

A todos o meu muito e sincero obrigado!

Joaquim Guilherme Vasconcelos Gonçalves da Silva

Resumo

Com a evolução tecnológica dos meios e dispositivos ao nosso dispor ao longo dos anos, devemos tentar aproveitar e usar o máximo possível dessa tecnologia para o bem-estar e segurança das pessoas, principalmente com as que necessitam de mais cuidados.

Esta dissertação enquadra-se precisamente nesse objetivo, o de aumentar a segurança através de um sistema aplicado numa cadeira de rodas. Partindo deste princípio, este projeto apresenta uma solução para que seja possível resolver este problema. É necessário por isso, desenvolver uma aplicação capaz de tomar decisões através de um sistema de visão artificial que seja aplicado ao problema em questão. Para isso, o dispositivo escolhido para este caso é o Microsoft Kinect, um equipamento de visão tridimensional, estudado profundamente de modo a compreender todo o seu funcionamento e constituição para melhor aplicar as suas características ao problema em causa. Para o uso deste dispositivo, desenvolveu-se uma aplicação capaz de realizar estudos e cálculos através da análise da imagem recebida, podendo tomar decisões e alertar para as mudanças no meio, recorrendo à plataforma Lazarus. Esta aplicação é assim capaz de rápidas análises à informação extraída da imagem recebida mas principalmente da profundidade que o Microsoft Kinect adquire do meio observado.

Numa fase inicial, recolheu-se imagens fixas do meio e recorreu-se ao *software* Matlab para a sua análise estática, de momentos e situações pontuais, ou seja, um processamento de imagem básico em que se detetam obstáculos ou impedimentos na imagem; numa segunda fase, a aplicação desenvolvida em Lazarus deteta, em tempo real e situações simples, objetos numa vista em frente e buracos ou desníveis no chão. Através da interface desenvolvida é possível visualizar a distância a certos pontos da imagem, chamados de sensores virtuais, assim como uma sinalização se o “caminho” está livre, ou encontra algum impedimento para o seu normal funcionamento.

Palavras-chave: *3D, análise de imagem, Kinect, sensores virtuais.*

Abstract

With the technological evolution of media and devices available to us over the years, we should try to take advantage and use as much as possible this technology for the welfare and safety of people, especially with those who need more care.

This work fits precisely in that goal, to increase the security through a system applied in a wheelchair. From this principle, this project presents a solution to be able to solve this problem. It is therefore necessary to develop an application capable of making decisions through an artificial vision system that is applied to the problem in question. For this, the device chosen for this case is the Microsoft Kinect, a device of three-dimensional vision, deeply studied in order to understand its functioning and constitution to better apply their characteristics to the problem in question. To use this device, has been developed an application that can conduct studies and calculations by analyzing the received image and can make decisions and alert for changes in the environment, using the Lazarus platform. This application is capable of quick tests to the information extracted from the input image but mostly the depth that Microsoft Kinect acquires from the local observed.

Initially, was collected fixed images from what was observed and used the software Matlab for its static analysis, from moments and specific situations, in other words, a basic image processing that detects obstacles or impediments in the picture; In a second stage, the application developed in Lazarus detect, in real time and simple situations, objects in a front view and holes or gaps in the ground. Through the developed interface is possible to visualize the distance to certain points in the image, called virtual sensors, as well as a signalization that indicate the "path" is free, or find any impediment to the normal functioning of the system.

Keywords: *depth, image analysis Kinect, virtual sensors.*

Conteúdo

1	Introdução	1
1.1	Motivação	1
1.2	Estrutura do Documento	2
2	Estado da Arte	4
2.1	Sensorização Passiva	5
2.1.1	Visão Estereoscópica	5
2.2	Sensorização Ativa	7
2.2.1	Triangulação ativa por luz estruturada	8
2.2.2	Triangulação ativa por <i>Time-of-Flight</i>	9
2.2.2.1	Sensor Ultrassónico	9
2.2.2.2	<i>Laser Rangefinder</i>	10
2.2.2.3	Câmara Time-of-flight	11
2.2.3	PrimeSense	11
3	Arquitetura e Ferramentas	15
3.1	Microsoft Kinect	15
3.1.1	História	16
3.1.2	Aplicações desenvolvidas com o Kinect	17
3.1.3	Componentes do Kinect	21
3.1.3.1	Sensor de profundidade	22
3.1.3.2	Câmara RGB	23
3.1.3.3	Motor, Acelerómetro e Microfones	23
3.1.4	Aquisição de Imagem	23
3.1.5	Limitações do Kinect	25
3.1.6	'Pin-Hole Model' do Kinect	27
3.2	Sdpo Freenect e Lazarus	29
3.3	Arquitetura do Sistema	30
4	Implementação e Resultados	33
4.1	Aplicação Desenvolvida	33
4.1.1	Interface	33
4.1.2	Funcionalidades da aplicação	34
4.2	Conversão da profundidade	35
4.3	Sensores Virtuais	39
4.4	Resultados	40
5	Conclusão e Trabalho Futuro	48
5.1	Trabalho Realizado	48

5.2 Trabalho Futuro	49
Referências	50

Lista de Tabelas

Tabela 1 - Tabela comparativa dos dispositivos da PrimeSense.....	14
Tabela 2 - Valores reais comparativamente aos valores calculados pelo Kinect	37
Tabela 3 - Tabela dos valores de distâncias calculadas pelos sensores.	42

Lista de Figuras

Figura 1– Exemplos de sistemas de visão estereoscópica.....	6
Figura 2– Diagrama de visão estereoscópica	7
Figura 3– Visão estereoscópica.....	8
Figura 4– Projeção de uma serie de padrões de listas com diferentes comprimentos de onda ..	9
Figura 5– Sensor ultrassónico Parallax's PING.....	11
Figura 6– Exemplo e modo de funcionamento de um sensor laser rangefinder	11
Figura 7– Exemplo e modo de funcionamento de uma câmara TOF.....	12
Figura 8– Equipamentos 3D desenvolvidos pela PrimeSense	13
Figura 9– Microsoft Kinect e seus componentes	15
Figura 10– Modo de interação dos três primeiros jogos com o Kinect: Ricochet (esquerda), Paint Party (centro) e Milo & Kate (direita)	17
Figura 11– Número de movimentos corretos do participante durante todo o programa.....	20
Figura 12– Mapeamento do esqueleto do utilizador	20
Figura 13– YScope, sistema de manipulação de imagens médicas	21
Figura 14– À esquerda, o reconhecimento da expressão facial, à direita, a segmentação da área da mão.....	22
Figura 15– Componentes e arquitetura do Kinect.....	23
Figura 16– Padrão de infravermelhos projetado pelo Kinect.....	25
Figura 17– Informação obtida pelo Kinect: RGB (esquerda), profundidade (centro) e IR (direita)	26
Figura 18– Ocultação de objeto	27
Figura 19– Três regiões de funcionamento, original da patente da PrimeSense.....	28
Figura 20– Pin-Hole do sensor de profundidade.....	29
Figura 21– Diagrama do sistema.....	31
Figura 22– Menú principal da aplicação	34
Figura 23– Valores do Kinect e os valores em metros gerados pela equação 4.1	36
Figura 24– Representação dos valores da tabela 2.....	37
Figura 25– Imagem de profundidade a cores (esquerda) e em escala de cinza (direita).....	39
Figura 26– Imagem de profundidade retirada pelo Kinect	41

Figura 27– Sensores virtuais na imagem de profundidade	42
Figura 28– Posição não paralela do Kinect a uma parede	43
Figura 29– Sinalização dos sensores	44
Figura 30– Detecção e sinalização de um obstáculo do lado esquerdo	45
Figura 31– Detecção e sinalização de um buraco	46

Lista de Abreviações

2D	Duas dimensões
3D	Três dimensões
CMOS	Complementary metal-oxide semiconductor
Fps	Frame per second
GB	Green blue
IR	Infrared
LIDAR	Light detection and ranging
OpenCV	Open source computer vision
RGB	Red green blue
RGB-D	Red green blue – Depth
RG	Red green
SDK	Software development kit
TOF	Time-of-flight

Capítulo 1

1 Introdução

1.1 Motivação

Pessoas com deficiências motoras ou com idade avançada que possuem limitações a nível de controlo motor, força ou movimentos ficam drasticamente limitados na realização de tarefas diárias, tais como vestir, tomar banho, pentear o cabelo ou mesmo deslocar-se. Adicionalmente, estes problemas podem reduzir a sua participação em atividades comunitárias e de lazer, e até mesmo ter um impacto negativo nas perspetivas de trabalho [1].

No contexto desta dissertação abordamos o caso específico das pessoas com dificuldade em se deslocar e que necessitam para isso de uma cadeira de rodas, quer no caso de curto prazo, quer em definitivo. Existem, basicamente, dois tipos de cadeiras de rodas: as manuais, mais convencionais, em que o utilizador ou uma segunda pessoa é necessária para empurrar a cadeira e guiá-la; e as elétricas, em que o próprio utilizador através de um *joystick* ou botões de comando, acelera, trava e muda de direção, ou seja, pode controlar todos os movimentos da cadeira sem ser necessário uma segunda pessoa e sem requerer demasiado esforço do próprio utilizador, que pode eventualmente e dependendo do seu estado, nem conseguir por isso utilizar uma cadeira de rodas manual.

De modo a tornar a cadeira de rodas elétrica ainda mais autónoma, e mesmo mais segura, pode-se melhorar alguns aspetos da própria cadeira, e desenvolver métodos que intervenham diretamente no sistema de controlo da cadeira de rodas. Assim, ao dotar a cadeira de rodas com um sistema de visão tridimensional associado a uma aplicação de controlo e tomada de decisões, podemos aumentar a sua segurança e fornecer ao utilizador indicações do que o sistema de visão deteta em tempo real. Para isso, necessitamos de estudar

ao pormenor um dispositivo que satisfaça estas necessidades. Foram analisados e investigados vários sensores e a escolha recaiu no Microsoft Kinect [2], conforme vai ser explicado no próximo capítulo.

Com este equipamento, integrado com a aplicação desenvolvida, aplicado na cadeira de rodas, é possível detetar obstáculos, desníveis e cavidades no meio, através da janela de observação do Microsoft Kinect e assim tomar a decisão de desviar, parar, mudar de direção ou mesmo aumentar a velocidade da cadeira de rodas.

Assim, conseguimos o objetivo de aumentar a segurança da cadeira de rodas, podendo até, numa fase mais avançada, contrariar o comando dado pelo utilizador da cadeira, se o mesmo não corresponder de facto, ao que o sistema de visão deteta e observa.

Além de tudo isto, conseguimos estes resultados cada vez mais acessíveis economicamente, dado que com a evolução tecnológica e dos dispositivos ao dispor de cada vez mais público, é possível serem criadas soluções bastante interessantes e fiáveis. Recorrendo á visão tridimensional, 3D, cada vez mais usada em detrimento da 2D, conseguimos acesso às mais diversas aplicações nos mais variados ramos da indústria, tecnologia, entretenimento, bem-estar e segurança das pessoas.

1.2 Estrutura do Documento

Para além da introdução, este documento contém mais 4 capítulos.

No capítulo 2 é apresentado o estado de arte atual, apresentando as tecnologias de visão computacional existentes, subdivididas pelas suas características.

A arquitetura e ferramentas são apresentadas no capítulo 3. Aqui é onde está o estudo teórico, onde são apresentados todas as características do Microsoft Kinect que foram estudadas de modo a perceber o seu funcionamento assim como a arquitetura do sistema realizado, com as suas várias fases.

No capítulo 4 é apresentada a aplicação desenvolvida em Lazarus. A análise inicial recorrendo ao Matlab e os passos desenvolvidos para o cálculo em tempo real das distâncias é aqui apresentado.

Finalmente, a conclusão e possíveis trabalhos futuros do trabalho desenvolvido, encontra-se no capítulo 5.

Capítulo 2

2 Estado da Arte

Neste capítulo serão apresentadas diversas soluções tecnológicas de visão computacional. Para a realização deste projeto, a tecnologia usada foi a visão tridimensional, 3D, por isso as referências bibliográficas usadas vão de encontro a esse mesmo tipo de visão escolhida.

Um sistema autónomo tem como principal objetivo realizar diversas operações de acordo com a informação que ele próprio capta e recolhe do meio envolvente. Este objetivo só é possível concretizar se o sistema possuir algum equipamento capaz de recolher essa informação. Os dispositivos capazes de tal função são chamados de sensores. Alguns sensores são usados para simples medidas de valores tais como a temperatura ou a velocidade de rotação de um motor. Para o caso em questão, estes sensores tem de ter a capacidade de fazer medições de distâncias a vários pontos do meio envolvente. De seguida, é necessário extrair essas informações e processar os dados recolhidos de modo ao equipamento ter a perceção do meio que o rodeia.

Existem diversos tipos de sensores, que podem ser classificados de várias formas [3]. De acordo com a sua função, estes podem ser divididos em duas classes, são essas:

- Propriocetivos – Sensores que medem valores internos do sistema, por exemplo, velocidade, bateria.
- Exterocetivos – Sensores que adquirem informação do meio envolvente ao sistema, como por exemplo, distância, intensidade luminosa.

Assim, de acordo com o explicado anteriormente, serão abordados apenas os sensores exterocetivos. Estes sensores podem ser também classificados pelo modo como interagem com o meio envolvente. Desta forma, pode-se classificar também como:

- Passivos – Sensores que medem diretamente a energia do meio adquirida, por exemplo, termômetros, microfones, câmara CMOS.
- Ativos – Sensores que emitem energia para o meio e medem a sua reação, como por exemplo, sensores ultrassônicos, laser rangefinders.

Assim, de acordo com esta última classificação, serão apresentados sensores de ambos os tipos, para se perceber as suas características, os que os distinguem, as suas vantagens e desvantagens na sua utilização.

2.1 Sensorização Passiva

Olhando para o mesmo objeto mas de diferentes pontos de vista, separados por um vetor de base b , como representado na figura 2, resulta em diferentes ângulos de visão. De uma forma ou de outra, esta diferença nos ângulos de visão resulta numa mudança no plano de imagem, chamado de disparidade, a partir da qual a profundidade ao objeto pode ser conhecida.

2.1.1 Visão Estereoscópica

Através da observação de uma cena a partir de dois pontos de vista diferentes é possível que a distância desses mesmos pontos a determinados objetos seja determinada. Uma configuração com dois sensores de imagem é chamada de sistema estereoscópico, figura 1.



Figura 1 – Exemplos de sistemas de visão estereoscópica

Muitos sistemas visuais biológicos, tais como a visão humana, realizam a percepção da profundidade deste modo. A figura 2 representa como a profundidade pode ser determinada a

partir de uma arquitetura de visão estereoscópica. Duas câmaras são colocadas próximas uma da outra, com eixos óticos paralelos. O vetor distância b entre os dois eixos óticos é chamado de base estereoscópica.

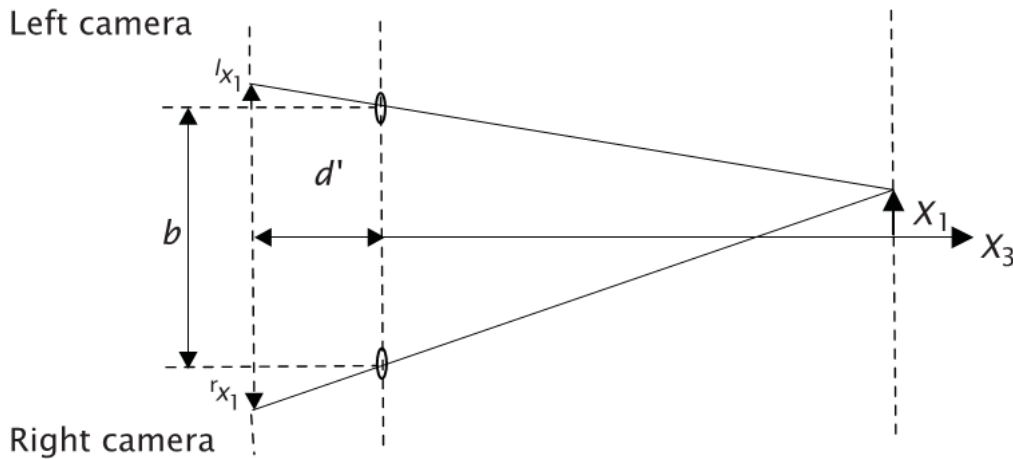


Figura 2 – Diagrama de visão estereoscópica

Um objeto estará projetado em diferentes posições do plano da imagem porque é visto de ângulos ligeiramente diferentes. A diferença na posição está denotada como disparidade ou paralaxe, p , facilmente calculada a partir da equação 2.1.

$$p = r_{x_1} - l_{x_1} = d' \frac{X_1 + \frac{b}{2}}{X_3} - d' \frac{X_1 - \frac{b}{2}}{X_3} = b \frac{d'}{X_3} \quad (2.1)$$

A disparidade é inversamente proporcional à distância X_3 do objeto (zero para um objeto no infinito) e é diretamente proporcional à base estereoscópica d' . Assim, a distância estimativa torna-se mais difícil com o aumento da distância.

A disparidade é uma grandeza vetorial e paralela à base estereoscópica b . Isto tem a vantagem de que se as duas câmaras estão exatamente orientadas no mesmo eixo, ou linhas epipolares, então é possível sabermos a direção da disparidade de antemão. Por outro lado, não se pode calcular a disparidade em todos os casos. Se um sector da imagem não mostra alterações nos valores de cinza da base estereoscópica, então não podemos determinar a disparidade.

A informação de profundidade contida nas imagens estereoscópicas pode ser percebida através de diferentes métodos. Em primeiro lugar, a imagem vista da esquerda e

vista da direita pode ser representada numa só imagem, se uma for mostrada em vermelho e outra em verde. O espectador usa óculos com filtro vermelho para o olho direito e com filtro verde para o olho esquerdo. Desta forma, o olho direito observa apenas o verde e o olho esquerdo apenas a imagem vermelha [4,6].

Concluindo, a disparidade pode ser calculada pelo método mencionado, figura 3(d), e assim descobrir a profundidade através da correspondência de duas imagens, achando uma característica comum às duas, ao longo das chamadas linhas epipolares, como por exemplo, o cubo da figura 3 [5].

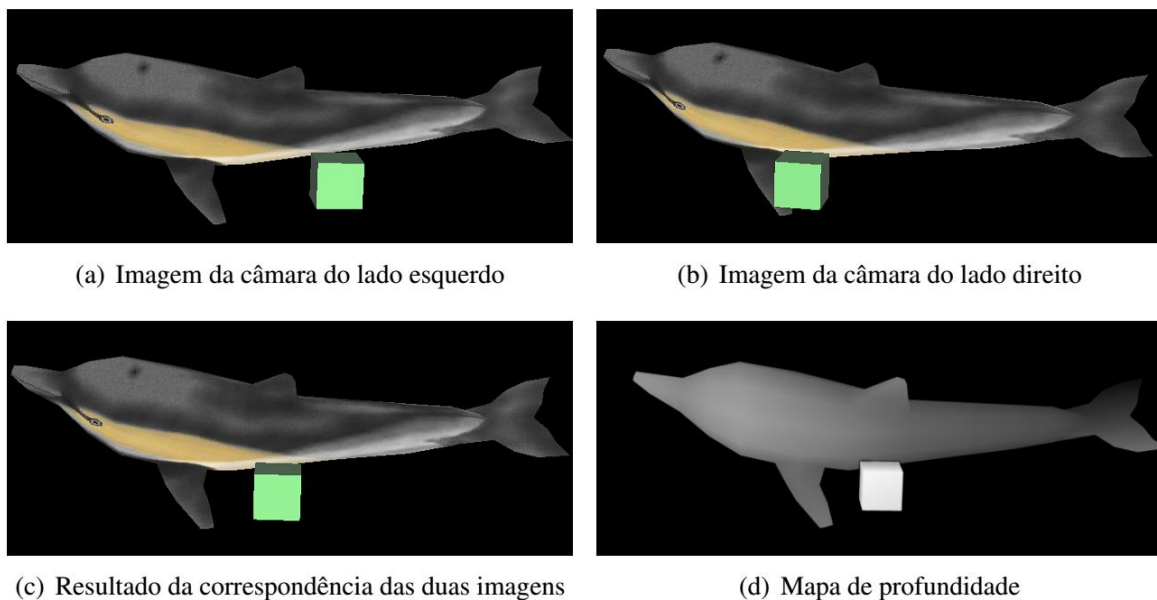


Figura 3 – Visão estereoscópica

2.2 Sensorização Ativa

Ao invés de uma arquitetura de visão estereoscópica, uma câmara pode ser substituída por uma fonte que emite um feixe de luz. De modo simples, estes sensores emitem um feixe de luz no meio envolvente que se está a observar. A reflexão dessa emissão de luz é capturada por um recetor, que nos indica características e nos permite concluir acerca da distância do objeto. Para sabermos a profundidade é então necessário identificar em cada *pixel* de que direção a reflexão do feixe de luz é capturada. Estes dados adquiridos é o equivalente aos

adquiridos da disparidade. Assim, uma técnica de triangulação ativa partilha todos os recursos básicos da visão estereoscópica, discutido no subcapítulo anterior.

2.2.1 Triangulação ativa por luz estruturada

Têm sido desenvolvidas técnicas sofisticadas nos últimos anos para codificar os feixes de luz de uma forma única.

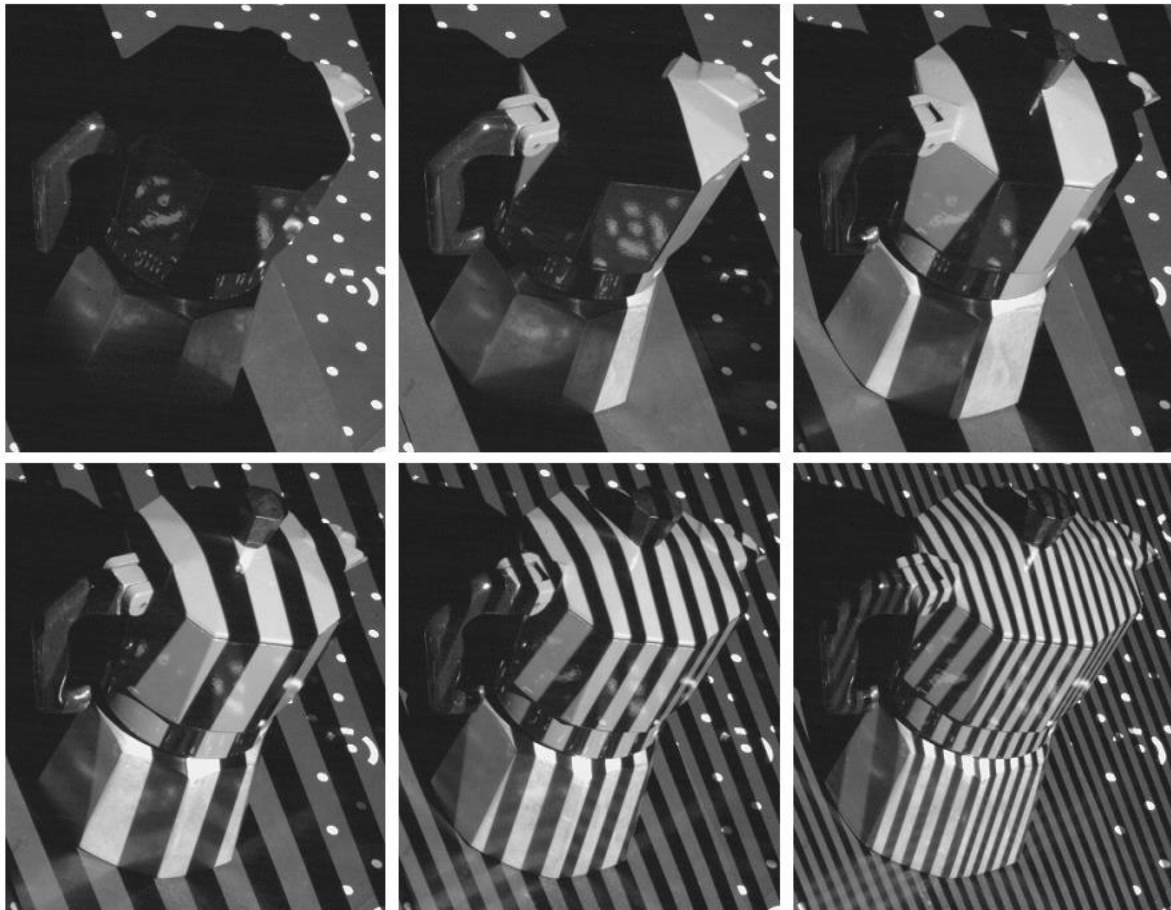


Figura 4 – Projeção de uma série de padrões de listas com diferentes comprimentos de onda

Normalmente, os emissores de luz usados projetam padrões com listas perpendiculares à linha da base da triangulação sobre a cena observada. Um único padrão não é suficiente para identificar a posição do padrão no plano da imagem, mas com uma sequência de padrões com listas, com diferentes comprimentos de onda, cada posição horizontal no plano de imagem do

emissor de luz pode ser identificado por uma única sequência de listas claras e escuras. Uma sequência de seis padrões parciais é mostrada na figura 4.

2.2.2 Triangulação ativa por *Time-of-Flight*

Sensores do tipo *Time-of-flight* medem o atraso causado pelo tempo que um sinal demora a percorrer determinada distância. Se o sinal é enviado a partir da posição da câmara, este tem de percorrer duas vezes a distância entre a camara e o objeto que reflete o sinal. Assim, o atraso t é dado por:

$$t = \frac{2z}{c} \quad (2.2)$$

Onde c é a velocidade do sinal e z a distância ao objeto. Pela equação (2.2) é evidente que o erro estatístico da medição da profundidade é independente da distância ao objeto, só dependendo da precisão do cálculo do atraso.

Com o *time-of-flight* imediatamente se pensa e se corresponde com a modulação de impulsos, isto é, medindo o tempo do atraso entre o envio e a receção de um impulso. A distância máxima medida depende da frequência com que os impulsos são enviados para o objeto [4].

2.2.2.1 Sensor Ultrassónico

Sensores ultrassónicos, figura 5, também conhecidos como transdutores, funcionam num princípio semelhante a um radar ou um sonar, o qual avalia atributos de um objeto observado, interpretando a reflexão das ondas geradas, rádio ou sonoras. Estes sensores emitem ondas de som e calculam o intervalo de tempo entre a emissão e a reflexão do sinal recebido de volta ao sensor para determinar a distância ao objeto.



Figura 5 – Sensor ultrassónico Parallax's PING

2.2.2.2 Laser Rangefinder

Um sensor *laser rangefinder*, figura 6, é um dispositivo que utiliza um feixe laser, através de ondas eletromagnéticas, para determinar a distância a um objeto. A forma de funcionamento destes sensores é a típica de um *time-of-flight*, emitindo um feixe laser em direção ao objeto de que queremos saber a distância e medindo o tempo que demora o impulso a ser refletido pelo objeto e recebido pelo sensor. Estes dispositivos são também conhecidos por sensores LIDAR (*LIght Detection And Ranging*) e a diferença para outros sensores *time-of-flight* reside no facto de poderem emitir vários feixes laser para o meio envolvente através de um mecanismo com um espelho rotativo. Estes sensores são bastante aplicados em áreas como a militar, geologia, mapeamento, condução autónoma de veículos, etc.

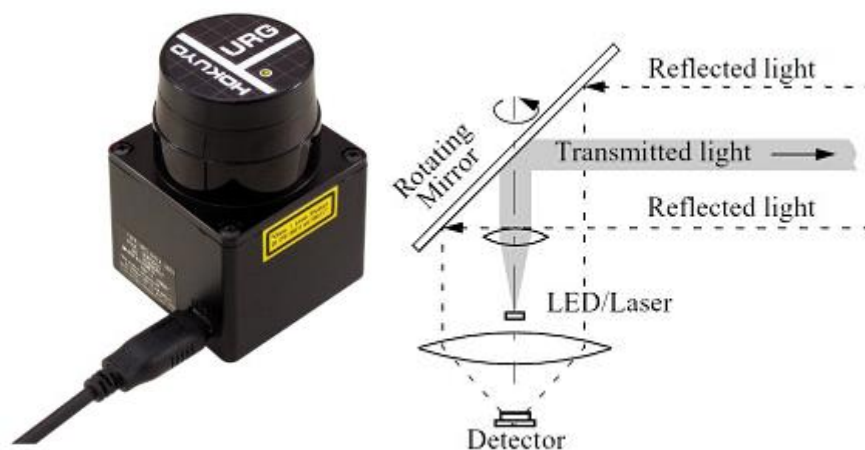


Figura 6 – Exemplo e modo de funcionamento de um sensor *laser rangefinder*

2.2.2.3 Câmara Time-of-flight

Uma câmara *Time-of-flight* (TOF), figura 7, é um sistema que mede a distância com base na velocidade da luz conhecida, medindo o tempo entre a emissão de um sinal de luz, reflexão e receção do mesmo sinal. O feixe de luz é emitido no meio envolvente e reflete nos objetos, a lente da câmara reúne a luz refletida e transpõe-na no plano do sensor. Dependendo da distância, a luz refletida pode sofrer atrasos. Este sistema tem na simplicidade, na eficiência no cálculo da distância e na velocidade boas vantagens, mas como desvantagens tem a interferência através de vibrações, luz de fundo e múltiplas reflexões.

Estes sensores são bastante utilizados em aplicações automóveis [7], interface homem-máquina e robótica.

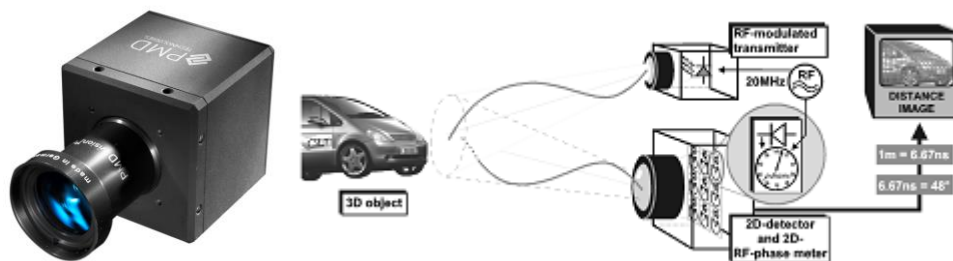


Figura 7 – Exemplo e modo de funcionamento de uma câmara TOF

2.2.3 PrimeSense

A tecnologia por trás do Microsoft Kinect [2] foi desenvolvida pela PrimeSense [9], uma empresa israelita responsável pelo desenvolvimento deste equipamento e de outros, com modos de funcionamento semelhantes mas com características próprias que os distinguem, o que fez com que fosse possível o avançar da evolução, figura 8.

Esta tecnologia desenvolvida pela PrimeSense, revelou-se rapidamente bastante útil e interessante devido às suas características, tendo deixado já o seu marco na comunidade

científica na área da visão tridimensional com várias aplicações nas mais diversas áreas de investigação.



1) PSDK Reference



2) Microsoft Kinect



3) Asus Xtion Live



4) Asus Xtion



5) PrimeSense Carmine

Figura 8 – Equipamentos 3D desenvolvidos pela PrimeSense

Existem atualmente vários produtos desenvolvidos pela PrimeSense, figura 8, como por exemplo:

1. PSDK Reference – Primeiro dispositivo criado pela empresa;
2. Microsoft Kinect – Desenvolvido como acessório para a consola Microsoft Xbox 360;
3. Asus Xtion Live – Idêntico ao Kinect, mas para uso em computadores;
4. Asus Xtion – Semelhante ao Asus Xtion Live, mas sem câmara RGB;
5. PrimeSense Carmine – Igual ao Asus Xtion Live, mas com nome da própria empresa.

Como dito anteriormente, todos estes dispositivos tem por base o mesmo modo de funcionamento, no entanto, possuem pequenos detalhes e características que os distinguem e que podem pesar aquando a escolha do equipamento [10]. O PSDK Reference já foi retirado do mercado, tendo sido substituído pelos outros equipamentos apresentados. O Asus Xtion, como não possui câmara RGB não foi tido em conta para esta análise, já que essa característica limita as suas capacidades. Assim, na tabela 1 são analisados os dispositivos mais idênticos e mais disponíveis, o Microsoft Kinect, o Asus Xtion Live [11] e o PrimeSense Carmine [9]. De salientar que o Asus Xtion Live e o PrimeSense Carmine são iguais, apenas diferem no nome da marca.

Dado o reconhecimento e a enorme visibilidade que o Microsoft Kinect obteve pela sua disponibilidade no mercado, graças à interação e lançamento para os videojogos, a comunidade cresceu e desenvolveu-se vários estudos e aplicações para este dispositivo, sendo por isso o usado neste projeto. De realçar que o preço deste três dispositivos é idêntico, estando por volta dos 150 euros, com possibilidade de ser ainda mais acessível num futuro próximo, aquando a saída do Microsoft Kinect 2.

Dispositivo	Vantagens	Desvantagens
MS Kinect	<ul style="list-style-type: none"> • Boa qualidade dos drivers • Estável com vários modelos de <i>hardware</i> • Motor de inclinação controlado remotamente • Mais acessível (mais popular) 	<ul style="list-style-type: none"> • Menos compacto e maior • Mais pesado • Necessita de alimentação externa • Menor qualidade de imagem RGB
Asus Xtion Live / PrimeSense Carmine	<ul style="list-style-type: none"> • Mais compacto e pequeno • Mais leve • Alimentação por USB • Melhor qualidade de imagem RGB 	<ul style="list-style-type: none"> • Menos acessível no mercado • Menor qualidade dos drivers • Não funciona com alguns controladores USB • Sem motor de inclinação

Tabela 1 – Tabela comparativa dos dispositivos da PrimeSense

O funcionamento deste dispositivo será abordado profundamente no capítulo 3, estudando todas as características, modo de funcionamento, a sua história e diversas aplicações nas mais variadas áreas em que é usado.

De modo geral, o Microsoft Kinect é constituído por uma câmara e um emissor de infravermelhos (IR), uma câmara RGB, dois microfones e um motor que permite variar a posição sobre o eixo horizontal. Assim, combinando estes constituintes conseguimos ter uma imagem em infravermelhos, uma em RGB e uma em profundidade. O projetor e a câmara infravermelhos juntos, conseguem transmitir uma nuvem de pontos em 3D através da triangulação de pontos no meio envolvente, e a câmara RGB permite obter informação sobre a cor e a textura dos objetos observados [12]. A figura 9 representa o Microsoft Kinect e seus constituintes.

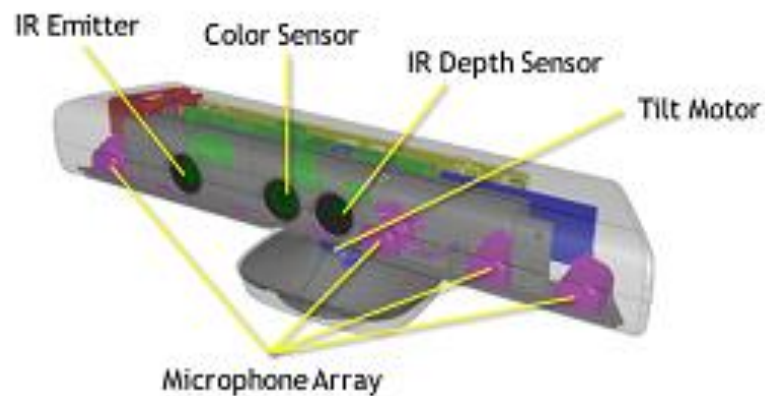


Figura 9 – Microsoft Kinect e seus componentes [13].

Capítulo 3

3 Arquitetura e Ferramentas

3.1 Microsoft Kinect

O Microsoft Kinect foi lançado no dia 4 de Novembro de 2010 como um acessório da consola Xbox 360. Este dispositivo foi desenvolvido pela empresa PrimeSense em colaboração com a Microsoft. Em Janeiro de 2012, mais de 18 milhões de unidades já tinham sido comercializadas.

O anúncio de lançamento já tinha sido no ano de 2009, tendo causado grande expectativa nas comunidades académicas de gráfica e visão computacional. O produto prometeu uma nova forma de interação nos videojogos, completamente baseada em gestos e comandos por voz (sem ser necessário qualquer outro tipo de comando). Desde a sua apresentação, o Kinect tornou-se um dispositivo largamente usado na indústria dos jogos, robótica, interface homem-máquina, etc. Assim, através do Kinect, novas possibilidades foram introduzidas no desenvolvimento de aplicações em várias indústrias e pesquisas.

O Kinect possui uma câmara RGB e uma câmara e emissor de infravermelhos. Estes são capazes de capturar imagem colorida e a profundidade de cada *pixel* no meio envolvente observado. Estes dados contêm informação geométrica e visual do ambiente que se observa. Com a combinação dos dados obtidos pela câmara RGB e de IR existe a possibilidade de execução de algumas tarefas que eram muito difíceis, se não mesmo impossíveis, se fossem usadas apenas imagens 2D. A maior parte dos sistemas de processamento de imagem são baseados apenas na cor das imagens. Todavia, outros atributos presentes nas imagens podem ser usadas para o processamento e análise, como por exemplo, profundidade, intensidade luminosa, etc. Estes atributos contêm informação que permite implementar alguns

procedimentos que são difíceis, se não impossíveis, usando apenas a cor. Consequentemente, a informação adquirida pelo Kinect, RGB e profundidade, possui uma estrutura que possibilita um novo modo de processamento de imagens.

3.1.1 História

O sensor de profundidade usado no Kinect foi desenvolvido por Zeev Zalevsky, Alexander Shpunt, Aviad Maizels e Javier Garcia, em 2005 [14]. O Kinect foi anunciado oficialmente no dia 1 de Junho de 2009, sob o nome “ Projeto Natal”, na E3, Electronic Entertainment Expo. O nome “Natal” foi uma referência à cidade brasileira Natal, porque um dos diretores da Microsoft responsável pelo projeto, Alex Kipman, é brasileiro e escolheu o nome da cidade, que significa “nascimento” em latim, para o nome do projeto.

Inicialmente, a principal função do Kinect era ser usado como ferramenta do utilizador para interagir com a consola Xbox 360 usando gestos e comandos por voz. Por esta razão, o sensor é capaz de capturar imagem numa resolução de 640x480 *pixels*, em 30Hz. Com a informação de profundidade, é possível obter uma esqueletização de quem está em frente ao sensor, e assim, é possível definir gestos do utilizador. O modo de como seria possível utilizar os gestos do utilizador usando o Kinect só foi anunciado em Janeiro de 2010, aquando o lançamento de três jogos: Ricochet, Paint Party e Milo & Kate. A figura 10 mostra esses três jogos.

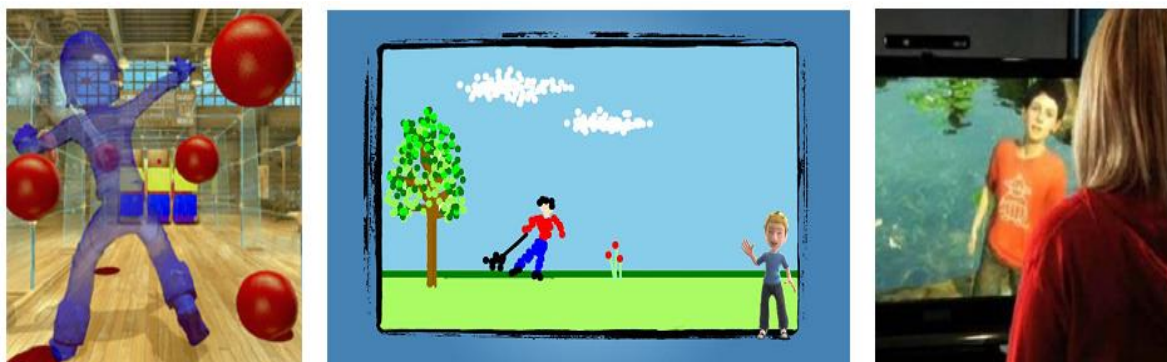


Figura 10 – Modo de interação dos três primeiros jogos com o Kinect: Ricochet (esquerda), Paint Party (centro) e Milo & Kate (direita).

Ricochet é um jogo que tem um personagem, ou avatar, que copia os movimentos do utilizador e o objeto é acertar em bolas virtuais. Paint Party é um programa de pintura e desenho, o utilizador usa formas e cores para desenhar virtualmente. Milo & Kate é o mais complexo. Neste jogo, o utilizador interage com um rapaz virtual (Milo) ou com um cão virtual (Kate). Este jogo usa inteligência artificial para definir os gestos e respostas do personagem virtual.

Nos primeiros 60 dias depois do lançamento, mais de 8 milhões de unidades foram vendidas. Assim, o Kinect tornou-se o consumível eletrónico vendido mais rapidamente do livro de recordes *Guinness Book*. Em Janeiro de 2012, o Kinect já tinha atingido mais de 18 milhões de unidades vendidas [15].

Com todo o interesse que o Kinect gerou sobre o seu modo de funcionamento e as suas características de interação na visão computacional, rapidamente cresceram comunidades dedicadas à liberalização do *software* e ao desenvolvimento de *drivers open source*. Assim, em 2010, várias entidades projetaram e tornaram públicos esses drivers para todos que quisessem desenvolver e investigar novos usos e aplicações para o Kinect pudessem ter acesso a este equipamento. Como exemplo desses drivers temos o libfreenect [16], desenvolvidos pela OpenKinect, e o OpenNI [17], criados pela PrimeSense.

Apercebendo-se do rápido crescimento e impacto que o Kinect gerou com estes drivers, através de todas as aplicações publicadas e desenvolvidas nas mais diversas áreas da visão computacional, a Microsoft acaba por lançar também, em 2011, um SDK para a programação do Kinect no sistema operativo Windows.

3.1.2 Aplicações desenvolvidas com o Kinect

Apesar do principal objetivo do Kinect ser a sua utilização em jogos da Xbox 360, não é apenas limitado a este tipo de aplicações. Existem várias outras possibilidades para o uso do Kinect tais como navegação e controlo de robots, entretenimento e aplicações de interação.

Alguns desses exemplos específicos são, por exemplo, na área de segurança o uso de gestos e comandos por voz para o controlo remoto de um robot, exploração de edifícios que contenham explosivos, etc. Na indústria dos cuidados de saúde, a visualização de uma série de raios X ou o controlo de operações médicas à distância devido à esterilização. Em fábricas

ou outro tipo de indústria, dependendo do tipo de ambiente de trabalho onde mudanças de funcionamento ou de gestão são complicadas, podemos usar o Kinect para simulação de tarefas específicas. Um arquiteto pode usar o Kinect para girar, aumentar, diminuir modelos virtuais e manipular esse mesmo modelo com as próprias mãos.

Posto isto, o Kinect pode ser usado em vários contextos e ambientes. Aproveitando estas possibilidades, várias aplicações foram desenvolvidas, das quais serão apresentadas algumas de seguida.

A. Sistema baseado no Kinect para reabilitação física de jovens adultos com dificuldades motoras [1].

Este estudo assenta na possibilidade de reabilitação de dois jovens adultos com dificuldade motoras utilizando um sistema baseado no Kinect num ambiente de escola pública.

Parte do princípio de uma sequência ABAB onde A representa uma linha de base, sem recurso a nenhuma tecnologia, apenas com a intervenção de uma terapia habitual, em que um fisioterapeuta demonstra e ajuda os participantes nos exercícios convencionais de terapia física. B representa as fases de intervenção, aqui recorrendo ao sistema desenvolvido, com os participantes a interagirem com o Kinect, imitando os movimentos dados pelo sistema.

Este estudo mostra que os participantes aumentam significativamente a sua motivação para a reabilitação física, e até melhoram a sua performance física durante as fases de intervenção. O gráfico da figura 11 mostra os resultados de um participante deste estudo, nos dois modos de reabilitação.

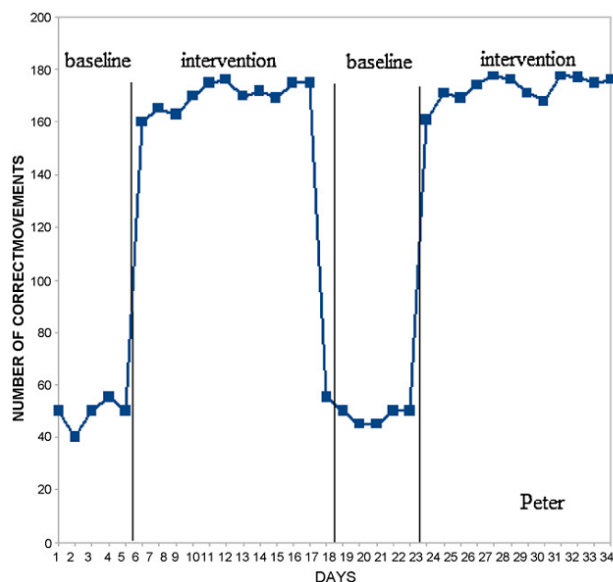


Figura 11 – Número de movimentos corretos do participante durante todo o programa.

B. Desenvolvimento de um jogo baseado em gestos para pessoas surdas/mudas usando o Kinect [18].

A maioria dos videogames interativos existentes funciona através de comandos de voz, o que dificulta o acesso a pessoas com problemas de audição e de fala. Assim, com o Kinect é possível o reconhecimento de gestos que o utilizador faz para realizar várias tarefas no jogo, substituindo os comandos de voz. Esta aplicação permite o controlo e interação do jogo pelo utilizador, através do mapeamento do esqueleto do utilizador para conseguir decifrar os gestos.



Figura 12 – Mapeamento do esqueleto do utilizador

C. YScope [19]

Os cirurgiões deparam-se com estritas restrições no bloco operatório quando necessitam de manipular imagens médicas devido à necessidade de manterem as mãos esterilizadas. Para terem acesso e poderem utilizar os exames, era usado um rato ou teclado do computador, o que requeria que se teria de repetir o processo de esterilização novamente, ou ter uma outra pessoa só para manipular as imagens médicas por eles.

Assim, a YDreams, em conjunto com o Hospital de Santa Maria em Lisboa, desenvolveu um sistema que permite a possibilidade, através da visualização por parte do Kinect, do próprio médico possa controlar as imagens médicas à distância através de gestos, mantendo assim as mãos estéreis.

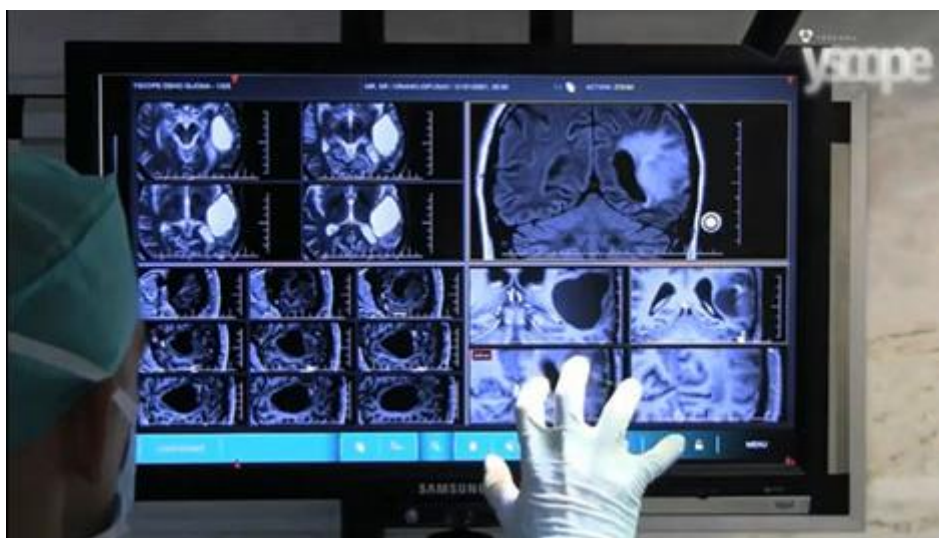


Figura 13 – YScope, sistema de manipulação de imagens médicas.

D. Reconhecimento de elementos da língua gestual português com Kinect [20]

Como o próprio nome indica, esta aplicação pretende reconhecer elementos da língua gestual portuguesa, já que os sistemas atuais dependem da interação entre várias soluções, elevando assim a sua complexidade e o seu custo.

Assim, esta aplicação tem com objeto principal desenvolver um procedimento simples, em que através do Kinect e das suas ferramentas de desenvolvimentos, seja possível em tempo real, detetar e seguir movimento humano para reconhecer os gestos da linguagem. Para isso apresenta duas fases: a expressão facial, onde são detetadas e reconhecidas expressões faciais utilizadas pela língua gestual portuguesa; e o reconhecimento de gestos estáticos, onde utiliza a informação de profundidade do Kinect para o processamento e rastreio do esqueleto do utilizador, para detetar e segmentar a área de interesse da imagem onde se encontra a mão do utilizador.

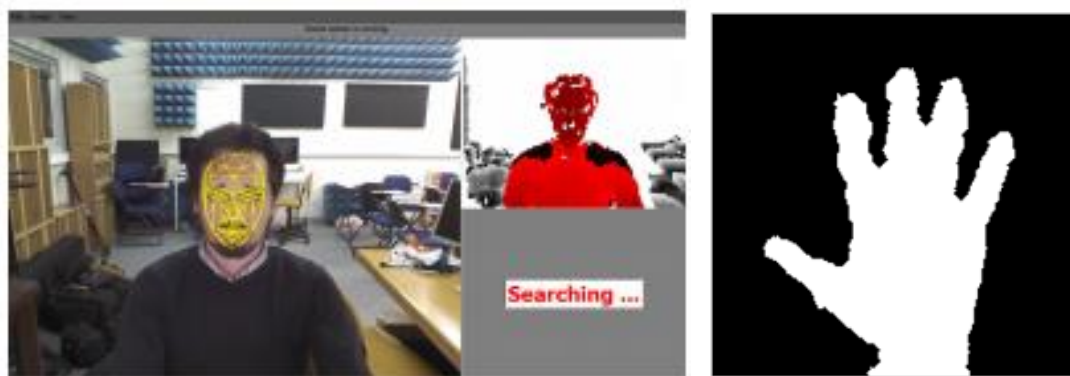


Figura 14 – À esquerda, o reconhecimento da expressão facial, à direita a segmentação da área da mão.

3.1.3 Componentes do Kinect

No desenvolvimento do Kinect o *software* interno foi desenvolvido pela Rare, uma subsidiária da Microsoft Game Studios, e a tecnologia da câmara de profundidade pela empresa Israelita PrimeSense. O Kinect é composto por um sensor de profundidade (câmara e emissor IR) uma câmara RGB, um acelerómetro, um motor e um *array* múltiplo de microfones. A figura 15 mostra os componentes do Kinect e a sua arquitetura de funcionamento.

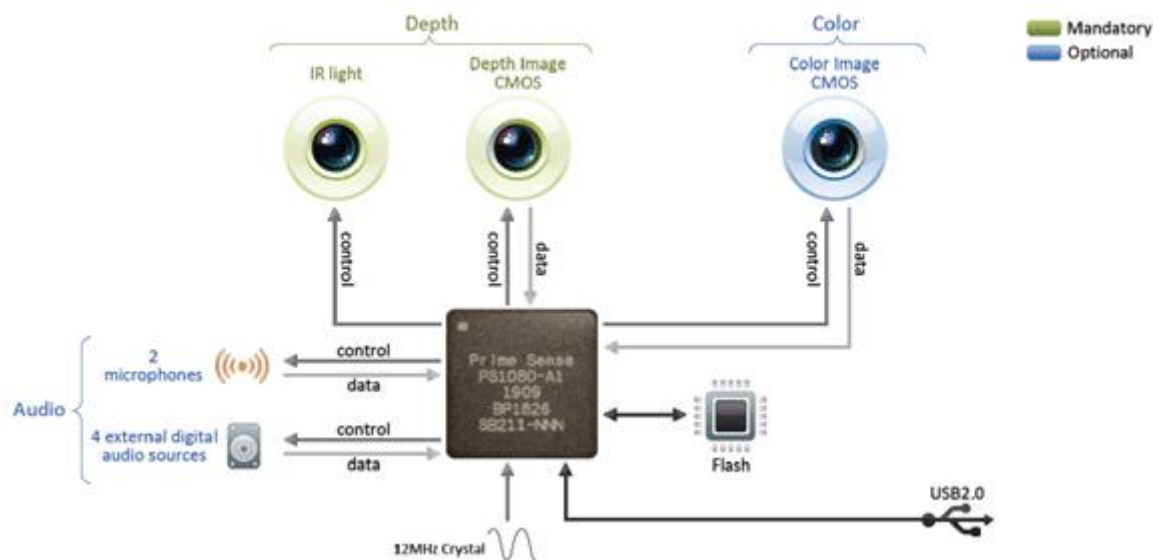


Figura 15 – Componentes e arquitetura do Kinect.

3.1.3.1 Sensor de profundidade

Este sistema consiste num emissor e numa câmara de infravermelhos (IR). O emissor cria um padrão conhecido de luz infravermelha estruturada.

A câmara IR funciona a 30 Hz e transmite imagens de 640x480 *pixels* de 11 *bits*, o que resulta numa sensibilidade de 2048 níveis. O campo de visão é de 58° graus horizontais, de 45° graus verticais e de 70° graus na diagonal e para bons resultados o alcance é entre 0.8 metros e 3.5 metros.

O sensor de profundidade usa o método de luz estruturada para as suas medições. Um padrão conhecido de pontos é projetado do emissor de infravermelhos. Esses pontos são guardados pela câmara de infravermelhos e comparados com o padrão conhecido pelo dispositivo. Quaisquer perturbações são conhecidas por serem variações no meio e podem ser detetadas mais próximas com mais precisão ou mais afastadas.

3.1.3.2 Câmara RGB

A câmara RGB funciona a 30 Hz, transmite imagens com uma resolução de 640x480 pixels com uma sensibilidade de 8 bits. O Kinect possui também a opção de aumentar a resolução da câmara, operando a 10 fps e 1280x1024 pixels. A própria câmara possui um conjunto de recursos incluindo balanço automático de brilho, saturação de cor, correção de defeitos e interferências, etc. A imagem transmitida pela câmara RGB é construída por um padrão de RG e GB.

3.1.3.3 Motor, Acelerómetro e Microfones

O Kinect possui dois sistemas que estão inter-relacionados e são bastante importantes: um método de inclinação do Kinect na horizontal e um acelerómetro. A inclinação é realizada através de um motor bidirecional que nos permite deslocar o Kinect no seu eixo horizontal. O acelerómetro é um dispositivo que mede a aceleração. O acelerómetro diz ao sistema qual a parte inferior, ou seja, qual a sua base medindo a aceleração devido à gravidade. Isto permite ao sistema estabelecer o eixo do Kinect no nível exato e de o calibrar para um valor que mova o Kinect para um ângulo específico.

O *array* de microfones proporciona quatro microfones que funcionam, cada canal de processamento, a 16 *bits* de som com uma taxa de amostragem de 16 kHz.

3.1.4 Aquisição de Imagem

A imagem colorida é obtida pela câmara RGB e o processo de cálculo da profundidade é realizado usando o emissor e a câmara de infravermelhos através de luz estruturada. Assim, como o Kinect utiliza luz infravermelha pode-se classificar como um dispositivo interior, já que no exterior a qualidade da medição da profundidade é afetada.

Como dito anteriormente, a profundidade é conhecida utilizando a técnica de luz estruturada. Esta abordagem consiste em projetar um padrão de *pixels* no meio envolvente e capturar a deformação da projeção, deformação essa que permite calcular a distância ao

pixels, ou seja, a profundidade. É necessário calibrar o emissor e a câmara de infravermelhos para realizar este cálculo porque a distância entre o emissor IR e a câmara é de 7.5 cm. Este cálculo consiste numa triangulação no emissor, câmara e posições dos *pixels*.

O padrão que o Kinect utiliza é patente da PrimeSense, é um padrão baseado numa mancha de pontos infravermelhos. Esse padrão é gerado a partir de um conjunto de elementos difrativos, com especial cuidado para reduzir o efeito de propagação de erro de ordem zero de pontos brilhantes, como mostra a figura 16.

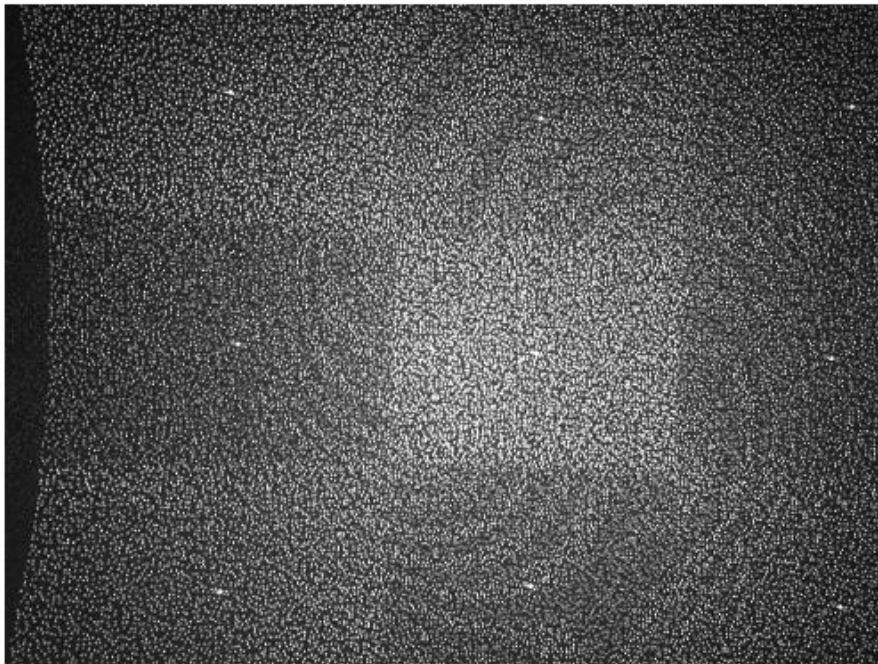


Figura 16 – Padrão de infravermelhos projetado pelo Kinect

Para cada *pixel* na imagem de infravermelhos, uma pequena janela de correlação de 9x9 ou 9x7, é usada para comparar o padrão local naquele pixel com o padrão memorizado nesse mesmo pixel e nos 64 pixels vizinhos numa janela horizontal. A melhor correspondência origina um *offset* da profundidade conhecida, em *pixels*, ou seja, a chamada disparidade. O Kinect realiza depois uma interpolação da melhor correspondência para obter uma precisão de 1/8 *pixel*. Dada a profundidade conhecida do plano memorizado e a disparidade, uma profundidade estimada para cada *pixel* pode ser calculada por triangulação.

Um *pixel* na imagem RGB refere-se a um ponto diferente do mesmo *pixel* na imagem de profundidade. Isto é por causa da diferença de posição das duas câmaras, RGB e IR, de cerca de 2.5 cm e para corrigir esta diferença é necessária a calibração as câmaras.

A representação da informação obtida com o Kinect é uma imagem RGB-D, ou seja, a combinação de três canais de cores (vermelho, verde e azul) e outro para a informação de profundidade.

A diferente natureza da informação neste tipo de imagens, as cores com uma natureza visual e a profundidade com uma natureza geométrica, permite o uso da informação RGB-D para realizar algumas tarefas que anteriormente eram muito difíceis, se não impossíveis apenas com informação RGB, como por exemplo, segmentação de objetos em tempo real ou reconhecimento de pose.

Além destes atributos, é possível também adicionar informações visuais, tais como luminância, ou informação geométrica, como a normalização ou gradientes. A informação visual pode ser calculada do mesmo modo que o processamento de imagem. A informação geométrica pode, na verdade deve, ser obtida recorrendo à profundidade [15].

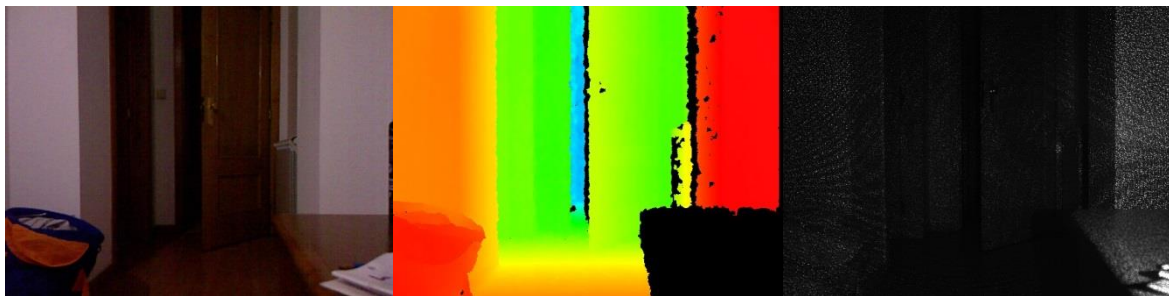


Figura 17 – Informação obtida pelo Kinect: RGB (esquerda), profundidade (centro) e IR (direita)

3.1.5 Limitações do Kinect

Como todos os equipamentos, também o Kinect possui algumas limitações devido às suas características.

Como o Kinect usa luz infravermelha, não é aconselhável a sua utilização no exterior, ou onde esteja presente a luz solar, pois do sol também provem luz infravermelha, o que pode causar danos no sensor IR do Kinect. Neste caso, o recetor iria receber não apenas a luz IR do

emissor mas também de uma fonte desconhecida, dificultando o processo de aquisição de imagem anteriormente explicado. Interferências com a luz IR também podem acontecer quando no meio observado existem objetos refletivos, transparentes ou que absorvem IR. Assim, para bons resultados, é aconselhável o uso do Kinect em ambientes interiores.

Outra limitação existente no Kinect é o facto de, quando existem dois objetos na cena observada, um deles causa ocultação do outro. Analisando a figura 18, se um dos objetos estiver na zona sombreada indicada na figura, não será detetado. Apesar da câmara de profundidade consiga encontrar o objeto, este não reflete o sinal IR. Este acontecimento provoca na imagem a chamada zona morta, denominados buracos negros, sem qualquer informação, o que pode dificultar a análise e aquisição de dados [21].

Outra limitação é a necessidade de calibração das imagens de profundidade e RGB se for necessária a sua correspondência, já que tem um ligeiro desvio entre as suas posições devido à posição das câmaras no Kinect.

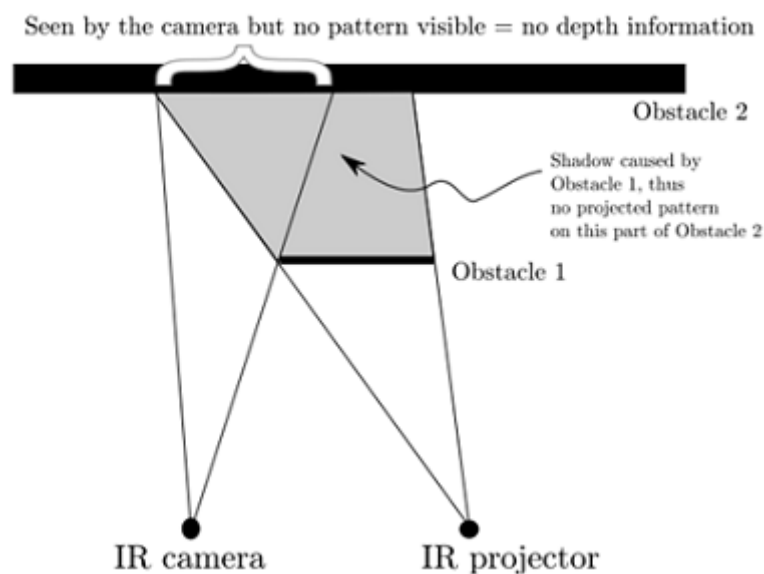


Figura 18 – Ocultação de objeto

A própria PrimeSense divide o cálculo de profundidade em três regiões distintas, cada uma com distâncias diferentes, como apresentado na figura 19. A primeira região possibilita a observação com valores bem precisos de objetos entre 0.8m e 1,2m, a segunda região tem média precisão para objetos entre 1.2m e 2m, e a terceira região já baixa na precisão dos resultados para objetos que estejam entre 2m e 3.5m. Acima dos 3.5m é possível detetar

objetos, mas os erros já são bastante grandes para uma aplicação que requer boas medições. Abaixo dos 0.5m o Kinect não deteta o meio envolvente.

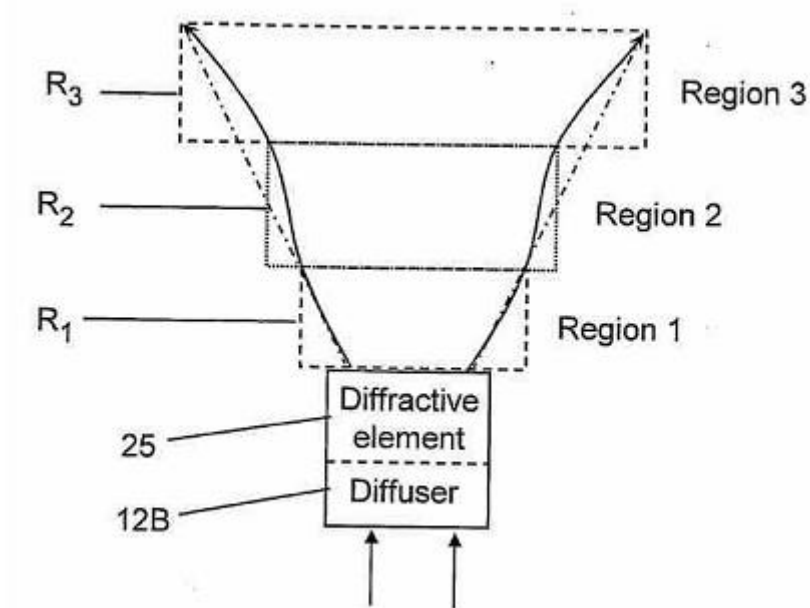


Figura 19 – Três regiões de funcionamento, original da patente da PrimeSense

3.1.6 'Pin-Hole Model' do Kinect

O 'Pin-Hole Model' de uma câmara não é mais do que descrever a relação matemática entre um ponto 3D no mundo e a sua projeção no plano de imagem. Este modelo consiste num orifício bastante pequeno por onde passam todos os raios de luz antes de serem invertidos pela lente da câmara e projetados no plano da imagem. A relação entre a real posição e a projetada no plano de imagem é dada pela distância focal. Para ser possível mapear um ponto 3D do meio envolvente para um ponto 2D do plano de imagem, o procedimento da projeção é de acordo com o explicado na figura 20.

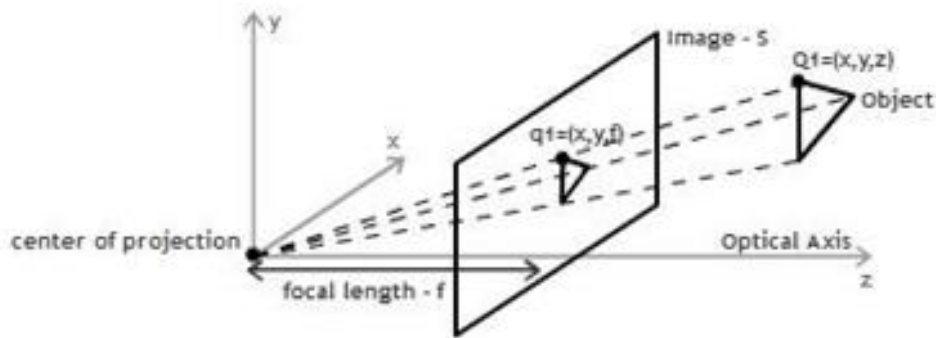


Figura 20 – Pin-Hole do sensor de profundidade

Na figura 20, um ponto $Q1$ nas coordenadas do mundo é projetado no plano de imagem passando pelo centro de projeção, resultando no ponto $q1$ no plano de imagem S . Como se pode verificar, a relação entre os dois triângulos similares pode ser usada para derivar a relação entre os pontos $Q1$ e $q1$ como se pode verificar pelas equações 3.1 e 3.2.

$$\frac{x}{f} = \frac{X}{Z} \quad (3.1)$$

$$\frac{y}{f} = \frac{Y}{Z} \quad (3.2)$$

De facto o centro de imagem do sensor não é normalmente no eixo ótico da câmara, o que leva à introdução de dois novos parâmetros, c_x e c_y para modelizar essa diferença. Existem também dois valores diferentes para a distância focal, uma por cada eixo de coordenadas. Este procedimento deve-se ao facto dos *pixels* numa câmara serem normalmente de forma retangular em vez de quadrangular. Estes novos parâmetros introduzem alterações nas equações 3.1 e 3.2, como se pode verificar nas equações 3.3 e 3.4.

$$x = f_x \frac{X}{Z} + C_x \quad (3.3)$$

$$y = f_y \frac{Y}{Z} + C_y \quad (3.4)$$

Assim cada valor da imagem, $f(x; y)$, tem correspondência direta com a coordenada Z do objeto (distância da câmara a um ponto do objeto). Conhecendo x e y , coordenadas do plano de imagem do ponto $f(x,y)$, determina-se facilmente as coordenadas X e Y no mundo de um ponto do objeto utilizando as equações 3.3 e 3.4, respetivamente [22].

3.2 Sdpo Freenect e Lazarus

A aplicação de controlo do Kinect foi desenvolvida tendo por base um componente de aquisição de imagem tridimensional. Assim, a aplicação foi implementada em linguagem FreePascal, recorrendo também à biblioteca de funções Sdpo Freenect. Sdpo Component Library (Sdpo) é um conjunto de componentes lançados pela Sdpo Robotic Soccer Team, uma equipa da FEUP (Faculdade de Engenharia da Universidade do Porto) que permitem o controlo de todos os componentes do Kinect em ambiente Linux, com várias funções de manipulação dos dados retirados em Lazarus.

Lazarus foi o *software* utilizado, devido à sua interface interativa e de fácil compreensão, que em conjunto com a biblioteca Sdpo nos possibilita o desenvolvimento de uma aplicação capaz de cumprir o objetivo deste trabalho.

3.3 Arquitetura do Sistema

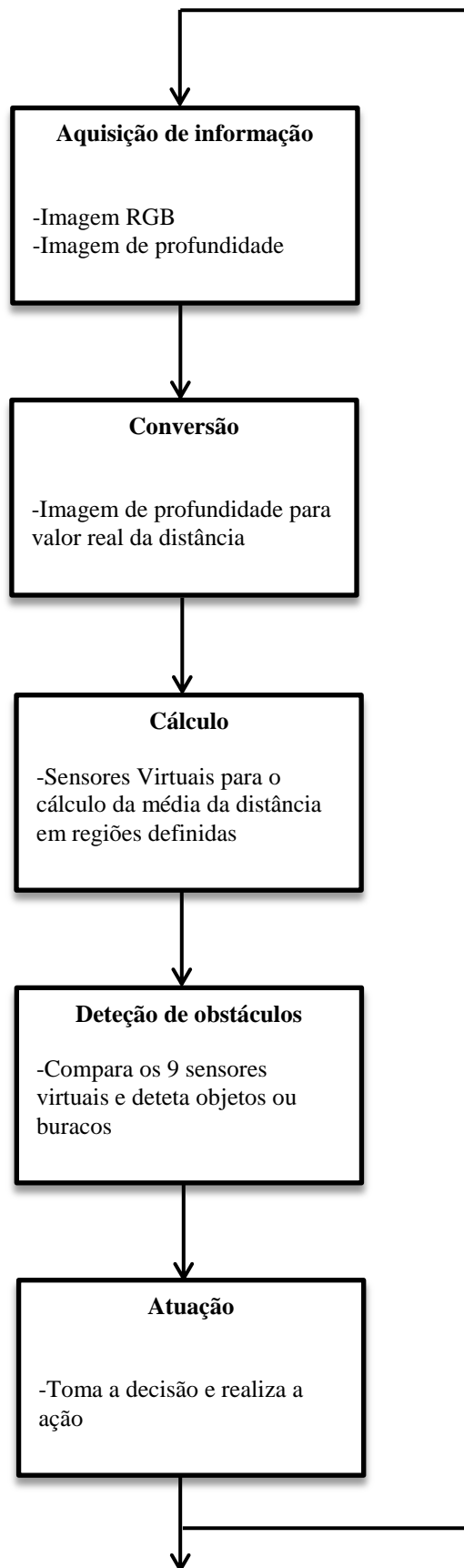


Figura 21 – Diagrama do sistema

Como referido anteriormente, este trabalho pretende aumentar a segurança numa cadeira de rodas através de um sistema baseado no Kinect. O Kinect captura informações do meio observado e uma aplicação analisa essa informação, fazendo as transformações e processos necessários para que no final o sistema tenha, em tempo real, uma autonomia de tomada de decisões consoante o que encontra no ambiente.

Para isso, a figura 21 representa a arquitetura do sistema, onde explica as várias fases de processamento de dados que o sistema possui: a aquisição de imagem, conversão, cálculo, deteção de obstáculos e atuação.

A aquisição de imagens corresponde às entradas de dados no sistema, resultado da captura de imagens por parte do Kinect. Essas imagens permitem a recolha de informação RGB e de profundidade, informação esta que está na base de todo o sistema e a partir da qual não era possível desenvolver um sistema deste género sem possuir estes dados.

Após a recolha das imagens, segue-se a conversão dos valores da profundidade que é dado pelo Kinect para valores de distância, ou seja, o Kinect retorna valores que necessitam de passar por uma fórmula matemática para que o resultado seja apresentado em unidade SI, metros. Este procedimento será explicado no capítulo seguinte em maior pormenor.

Já com os valores convertidos e possuindo distâncias reais, procede-se à fase de cálculo. Esta fase recorre a um método de sensores virtuais que mais não são que análises a regiões específicas da imagem de profundidade. Os valores das distâncias a cada *pixel* observado da imagem de profundidade estão presentes numa matriz com a dimensão da resolução da imagem, ou seja, 640x480. Os sensores virtuais são sub-matrizes, de dimensões 20x20, de regiões específicas da matriz das profundidades. Assim, os sensores virtuais calculam a média da vizinhança 20x20 desses pontos, retornando um valor médio da distância da região observada.

O sistema possui nove sensores virtuais e depois de calculados os valores dos sensores, o próximo passo é a deteção de obstáculos. O processo de deteção não é mais que fazer a comparação dos valores dos sensores. Assim, este passo é dividido em duas partes, a parte superior, onde os primeiros seis sensores são utilizados, e a parte inferior onde os últimos três sensores são utilizados. Numa vista de frente, os seis sensores fazem comparações entre si sempre que há alterações dos valores da profundidade. A deteção de objetos é efetuada se as diferenças de valores entre os sensores forem acentuadas. O mesmo

processo é utilizado para a parte inferior, onde são detetados desníveis ou buracos no chão, quando o Kinect está orientado nesse sentido.

Depois de todos estes passos o sistema, de acordo com os dados recebidos da detecção, tem uma resposta. Neste caso, na aplicação desenvolvida a interface possui um sistema de sinalização, ou seja, é possível ver em tempo real a distância que cada sensor está da região observada e se não existirem entaves, o sinal está verde, se for detetado algum objeto ou buraco, o sinal fica vermelho nessa zona. Assim, no desenvolvimento futuro desta aplicação, é possível efetuar uma decisão que movimento para que a cadeira de rodas possa prosseguir o trajeto. Outro detalhe é que todo o objeto que apareça a 60 cm de distância ou menos, em qualquer parte da imagem, o sistema responde também com sinal vermelho, podendo ser associado a uma ação de travagem da cadeira.

Capítulo 4

4 Implementação e Resultados

Neste capítulo será analisado todo o desenvolvimento da aplicação para o controlo deste sistema, sendo descritas todas as etapas da arquitetura do sistema anteriormente apresentado. Pretende-se que o sistema em tempo real analise o meio envolvente para assim detetar a existência de obstáculos e agir de acordo com essas deteções. Assim, a cadeira de rodas tem a sua segurança aumentada e realiza ações que lhe permitam o deslocamento evitando colisões.

4.1 Aplicação Desenvolvida

A aplicação desenvolvida permite a imediata visualização das imagens adquiridas nos diversos formatos, RGB, IR ou profundidade assim como a visualização de diversas informações extraídas dessas mesmas imagens capturadas pelo Kinect. A aplicação foi desenvolvida em Lazarus, um *software* livre que recorre ao FreePascal como linguagem de programação.

4.1.1 Interface

No desenvolvimento da aplicação teve-se o especial cuidado de criar uma aplicação simples de usar e de fácil controlo, de modo a poder ser usada pelo maior número de pessoas. Assim, a figura 22 apresenta a janela principal da aplicação, que permite observar diversas informações.

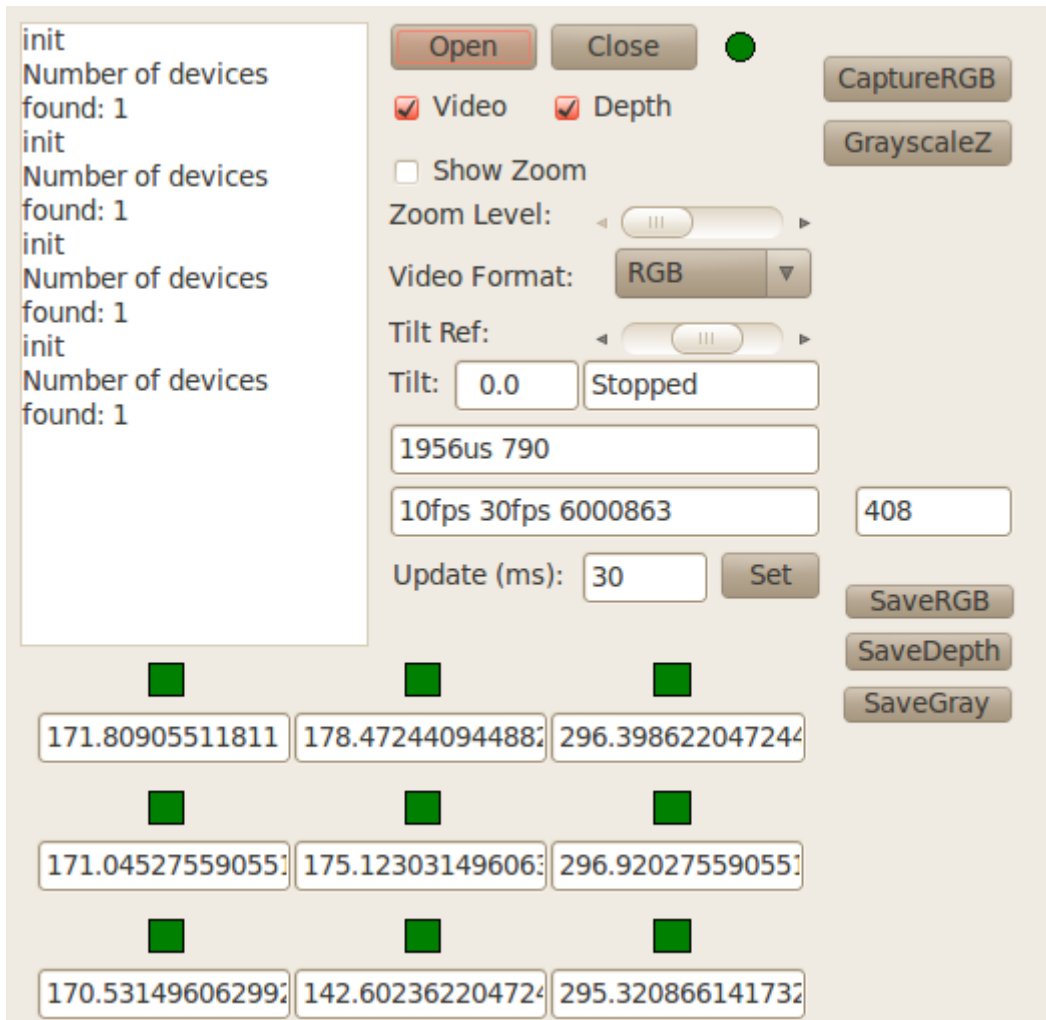


Figura 22 – Menu principal da aplicação

4.1.2 Funcionalidades da aplicação

De seguida são enumeradas as funcionalidades da aplicação desenvolvida.

1. Visualizar imagem vídeo em RGB, Bayer e IR.
2. Visualizar imagem de profundidade.
3. Fazer *zoom* de determinada região da imagem.
4. Capturar e guardar imagens de vídeo.
5. Capturar e guardar imagens de profundidade (colorida e em escala de cinza).
6. Variar o ângulo de inclinação do Kinect.
7. Variar a taxa de atualização de dados recebidos pelo Kinect.

8. Visualização em tempo real da distância dos sensores virtuais.
9. Sinalização dos sensores virtuais.

Assim, através destas funcionalidades é possível analisar várias informações da imagem. É possível visualizar tudo em tempo real, e a partir daí tomar decisões em relação ao que é visualizado.

Resumidamente, as principais funcionalidades desta aplicação possibilitam a visualização do meio observado pelo Kinect, mostrando uma janela em que é possível observar em RGB, Bayer e IR, e uma janela em que é possível observar o meio em profundidade. Podemos assim capturar e guardar o momento observado numa imagem em RGB ou em profundidade, aqui já transformada em escala de cinzas para que seja mais fácil a sua análise, já que os valores presentes nessa escala indicam o valor da distância real. É possível também a visualização dos valores das distâncias a que estão os sensores virtuais e a sua sinalização, ou seja, se o sensor estiver verde significa que essa região que está a ser calculada em comparação com os sensores vizinhos está de acordo com as distâncias mínimas do normal funcionamento do sistema, se o sensor estiver vermelho, o sistema toma uma decisão que deve mover a cadeira para uma posição em que volte a retornar o sensor verde para que esta possa realizar o seu trajeto.

Nos próximos subcapítulos serão abordados mais detalhadamente os meios e processos no desenvolvimento destas funcionalidades.

4.2 Conversão da profundidade

Como dito anteriormente no capítulo 3.3, o sistema obtém do meio observado valores que necessitam de ser convertidos para valores em unidades SI (metros). Deste modo, nesta aplicação foi usado uma função matemática que funciona como conversor dos dados retornados diretamente pelo Kinect. O modelo proposto para essa conversão usa a equação 4.1, desenvolvida por Stephane Magnenat [23].

$$p_m = 0.1236 \tan\left(\frac{p_k}{2842.5} + 1.1863\right) \quad (4.1)$$

Através da equação 4.1 é possível saber o valor em metros, pm , usando o valor retornado pelo Kinect, pk .

O gráfico da figura 23 permite também verificar os valores retornados pelo Kinect e os valores reais correspondentes. Os dados observados permitem também concluir o já referido anteriormente, que o Kinect obtém bons resultados desde 0.5 metros até 3.5 metros de distância, conseguindo depois medições até 10 metros mas com erros bastante grandes.

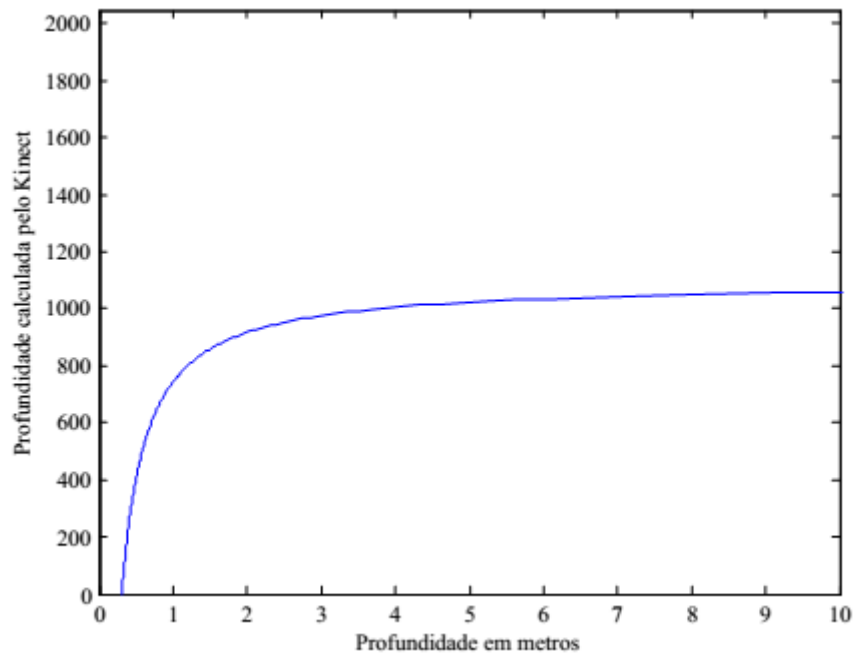


Figura 23 – Valores do Kinect e os valores em metros gerados pela equação 4.1

Como apresentado anteriormente, o erro de cálculo da profundidade aumenta proporcionalmente à distância a que se encontra o Kinect. Isto deve-se a determinadas características próprias do Kinect e ao seu método de cálculo. Assim, é necessário perceber a evolução do erro. Para isso, foi realizado um simples teste que consistiu na colocação de um objeto plano em frente ao Kinect e respetiva extração da medida de profundidade do dispositivo ao centro do plano.

Os resultados obtidos são apresentados na tabela 2.

Mundo	<0,5	0,50	0,75	1,0	1,25	1,50	1,75	2,00
Kinect	n.d	0,4977	0,7489	0,9989	1,2465	1,4985	1,75544	1,9944
Mundo	2,25	2,50	2,75	3,00	3,25	3,50	3,75	4,00
Kinect	2,2509	2,5087	2,7661	3,0289	3,2536	3,5143	3,7792	4,0403
Mundo	4,25	4,50	4,75	5,00	5,25	5,50	5,75	6,00
Kinect	4,2870	4,5658	4,8833	5,0959	5,3279	5,5819	5,8614	6,1704

Tabela 2 – Valores reais comparativamente aos valores calculados pelo Kinect.

De acordo com os resultados da tabela 2, obtém-se a forma do gráfico da figura 24, muito semelhante ao da figura 23, que representa a função matemática que permite a conversão dos dados obtidos pelo Kinect em valores SI (metros). É possível observar que a forma dos dois gráficos é muito semelhante, sendo esse resultado o esperado, o que demonstra também que o método matemático apresentado possui resultados semelhantes tanto teoricamente, como na implementação prática, sendo por isso uma boa escolha para o cálculo da profundidade.

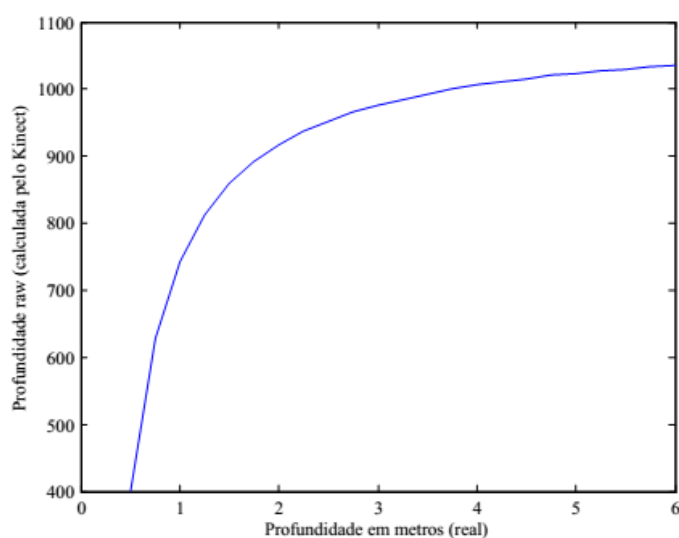


Figura 24 – Representação dos valores da tabela 2

Assim, de acordo com os dados apresentados na tabela 2, verifica-se o que já foi apresentado teoricamente, o Kinect não é capaz de calcular distâncias inferiores a 0.5 metros. Verifica-se também que a partir de 6 metros, vai perdendo gradualmente informação até ficar completamente impossibilitado de calcular distâncias.

Verifica-se também que o erro tem um crescimento mais evidente a partir dos 3 metros de profundidade, que vai de acordo ao apresentado anteriormente, na divisão do Kinect em três áreas distintas de precisão, onde a última zona, correspondente à área onde o Kinect começa a perder precisão, se situa nos 3.5 metros. É possível assim verificar que a partir desse valor o erro tende para valores que podem ultrapassar os 10 cm. Por este motivo, muitos investigadores que estudam o Kinect consideram que apenas se deve utilizar este dispositivo para medições até 3 metros, pois a partir deste valor, não se obtém uma boa precisão para o uso em aplicações que requerem valores mais precisos [5].

De seguida é apresentado um excerto de código onde é possível visualizar a implementação do método matemático anteriormente apresentado.

```
procedure TFMMain.initDepthTable;
var i: integer;
    k1, k2, k3: double;
begin
    // Stephane Magnenat version
    k1 := 1.1863; k2 := 2842.5; k3 := 0.1236;
    for i := 0 to 2047 do begin
        t_gamma[i] := round(256 * k3 * tan(i/k2 + k1));
    end;
```

Deste modo, é possível implementar o método de Stephane Magnenat para o cálculo de distâncias, ou seja, t_gamma vai obter todos os valores entre 0 e 2047 que são todos os valores que podem ser observados pelo Kinect. Depois através da multiplicação dos valores realmente obtidos pelo Kinect, $ImageZ$, com o t_gamma , obtém-se os valores que estão a ser observados em tempo real pelo Kinect, sendo esses valores os usados pelo sistema para os sensores virtuais. Estes valores permitem assim converter a imagem de profundidade a cores para uma imagem de profundidade em escala de cinza, sendo os valores de cinza, os valores das distâncias reais do Kinect ao objeto, figura 25.

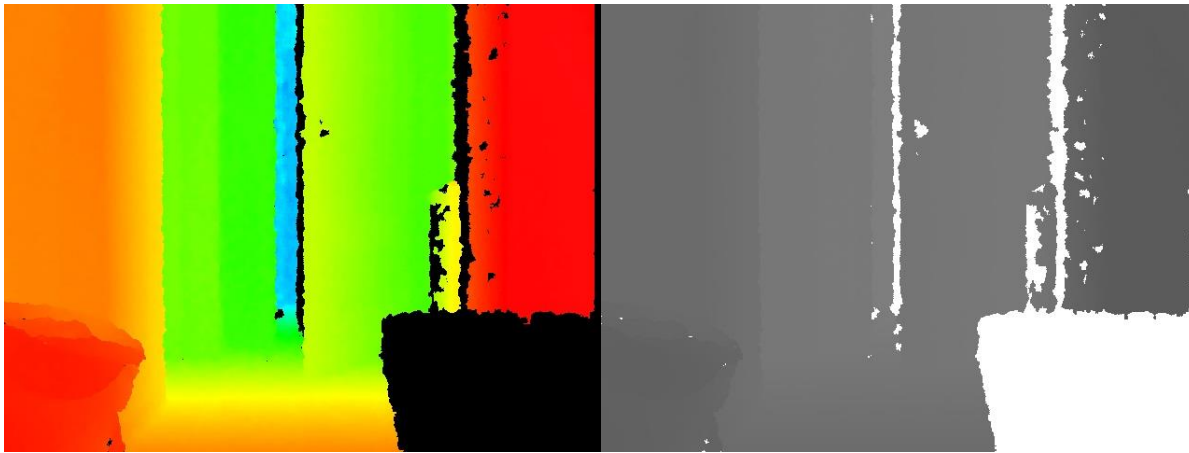


Figura 25 – Imagem de profundidade a cores (esquerda) e em escala de cinza (direita).

4.3 Sensores Virtuais

A implementação de sensores virtuais foi o método usado para o cálculo das distâncias. Um sensor virtual não é mais do que uma área selecionada da imagem que se observa., ou seja, o Kinect recolhe a imagem do meio observado e o sistema transforma a informação recolhida numa matriz. Esta matriz é convertida noutra mas já com os valores reais da distância. Com esta matriz, definem-se determinados *pixels* que correspondem a pontos na imagem. De seguida procede-se à definição de uma vizinhança de 20x20 em torno desse *pixel* e faz-se um cálculo da média desses valores. Esta média é o valor do sensor.

Nesta aplicação definiram-se nove sensores, sendo a sua localização a seguinte:

- Sensor 1 – S1 (120, 160)
- Sensor 2 – S2 (120,320)
- Sensor 3 – S3 (120,480)
- Sensor 4 – S4 (240,160)
- Sensor 5 – S5 (240,320)
- Sensor 6 – S6 (240,480)
- Sensor 7 – S7 (360,160)
- Sensor 8 – S8 (360,320)
- Sensor 9 – S9 (360,480)

Através deste método é possível o cálculo da distância de várias regiões definidas na imagem e usar essa informação para fazer a comparação entre si. Assim, consegue-se detetar mudanças bruscas ou diferenças de profundidade que nos podem indicar a existência de objetos ou desníveis no meio envolvente observado.

De seguida é apresentado um excerto de código que indica como um sensor virtual é calculado.

```
cm := 2.54; sensor := 400;

soma1:=0; media1:=0;

for x:=110 to 130 do
for y:=150 to 170 do

soma1:= soma1 + t_gamma[ImageZ[x,y]] ;

media1:= (soma1/sensor)/cm;

Sensor1.text := floatToStr(media1);
```

O código anterior calcula a distância do sensor 1. Assim, para todos os valores da vizinhança 20x20 do ponto (120,160) é feita a média desses valores. Um pormenor é o facto de este valor ser retornado em polegadas, sendo por isso necessário converter para metros, por isso a divisão da média por 2.54. Deste modo é apresentado na interface da aplicação, na posição do sensor 1 a distância calculada. Esta distância é atualizada sempre que a imagem adquirida pelo Kinect também é atualizada, obtendo assim novos valores e procedendo novamente ao cálculo do sensor virtual.

Este processo é repetido para todos os sensores definidos, modificando apenas os valores da posição do sensor.

4.4 Resultados

Até aqui foi apresentado todo o processamento de dados possibilitado pela aplicação. Assim, depois da fase de aquisição de imagem, de conversão da profundidade e do cálculo dos sensores virtuais, segue-se a fase de deteção de obstáculos.

A deteção de obstáculos, de modo simples, é a comparação feita entre os sensores virtuais, ou seja, sempre que exista uma diferença significativa entre os valores dos sensores, é muito provável a existência de um obstáculo ou de um desnível da superfície.

Apesar disso, é necessário ter em consideração as limitações do Kinect apresentadas anteriormente, nomeadamente a ocultação de objetos na emissão dos IR e os erros das regiões de medição.

Numa fase inicial da análise dos resultados recorreu-se ao *software* Matlab, um programa de análise e processamento de imagem bastante intuitivo e que nos permite facilmente desenvolver um algoritmo de análise a imagens retiradas de determinadas situações estáticas do meio observado pelo Kinect. Assim, recorrendo a este algoritmo facilmente se percebe de que modo se pode obter bons resultados numa fase posterior da análise em Lazarus. Partimos por isso da análise de momentos pontuais capturados pelo Kinect, de situações que possam acontecer em tempo real.



Figura 26 – Imagem de profundidade retirada pelo Kinect

Na figura 26 temos uma situação observada pelo Kinect. O Kinect está posto a aproximadamente 70 cm de uma parede em que do lado direito existe um obstáculo, com 12 cm de largura. Esta imagem está convertida em níveis de cinzentos e os valores da cor presente na matriz que representa a imagem correspondem à distância que foi detetada pelo Kinect.

Com isto, desenvolveu-se um algoritmo semelhante ao utilizado na aplicação em tempo real. Foram definidos nove sensores virtuais em posições específicas da imagem, figura 27, com uma vizinhança de 20x20, em que é feito o cálculo da média das distâncias presentes nesses *pixels*.

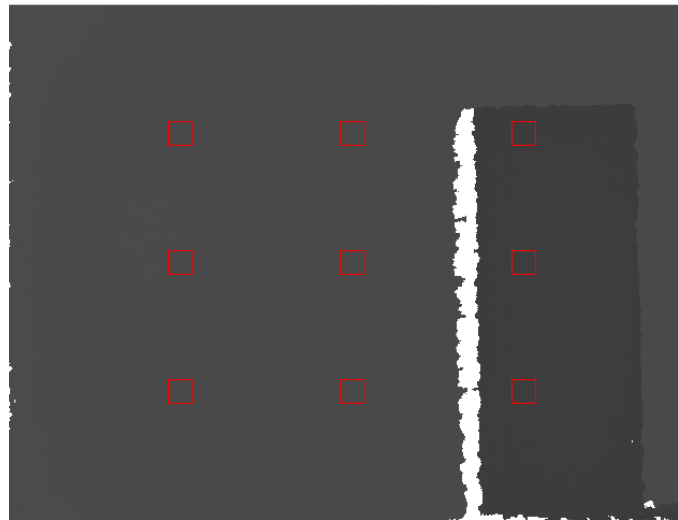


Figura 27 – Sensores virtuais na imagem de profundidade

Na figura 27 está representada a localização dos nove sensores virtuais, já anteriormente identificados no capítulo 4.3.

Os sensores virtuais calculam a média de todos os *pixels* numa matriz com dimensão 20x20 e com centro nos pontos acima identificados. Assim é possível obter a distância a que essas regiões estão do Kinect. Na tabela 3 estão as distâncias calculadas na figura 27.

Sensor 1	74.0023
Sensor 2	73.0000
Sensor 3	60.0204
Sensor 4	74.0091
Sensor 5	73.0000
Sensor 6	61.0000
Sensor 7	74.0000
Sensor 8	73.2336
Sensor 9	60.8821

Tabela 3 – Tabela dos valores das distâncias calculadas pelos sensores

Assim com estes valores podemos claramente identificar a existência de algum objeto no lado direito da imagem e os valores correspondem à distância real a que o Kinect se encontra do meio observado. Assim, os sensores S3, S6 e S9 detetam o objeto por comparação do resultado com os outros sensores.

A partir desta análise inicial procedeu-se ao próximo passo, a análise em tempo real da aplicação em Lazarus, contrariamente a este resultado que apenas analisa situações estáticas.

Assim, procedeu-se à análise num ambiente em que as condições pudessem ser as mais perfeitas possíveis, ou seja, para a deteção de objetos, o Kinect está numa posição central e paralela, com vista em frente a uma parede onde aparecem os objetos, e também numa vista para o chão, onde deteta desníveis ou buracos na superfície observada. Isto assegura que o plano de fundo tem praticamente a mesma distância ao Kinect para quando a comparação dos sensores é realizada, para não detetar um obstáculo quando ele não existe, por exemplo, se se colocar o Kinect numa posição que não seja paralela à parede, verifica-se, figura 28, que a distância é maior da parte esquerda da imagem em relação à parte direita, e apesar de não existir nenhum obstáculo, a aplicação deteta algo no sensor respetivo, figura 29.



Figura 28 – Posição não paralela do Kinect a uma parede

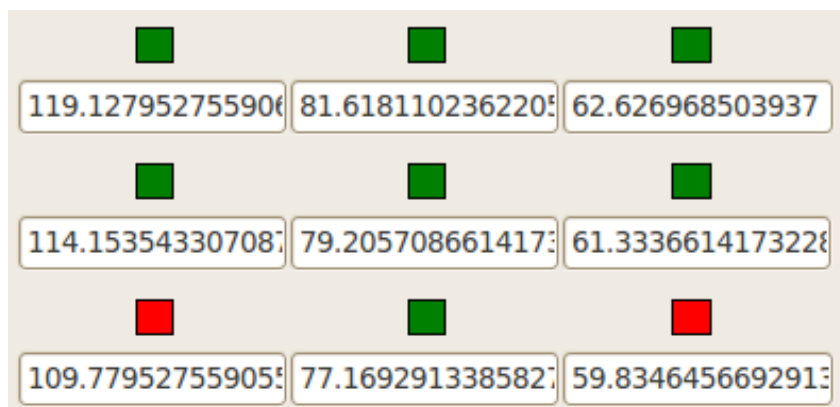


Figura 29 – Sinalização dos sensores

Podemos também verificar na figura 29, uma medida de segurança, ou seja, sempre que algum sensor calcular algum valor inferior a 60 cm, o sensor respectivo torna-se vermelho na *interface*.

Assim, feitas as comparações e dependendo dos resultados obtidos, a aplicação atua, neste caso, tornando o sensor verde se não existirem obstáculos, ou tornando o sensor vermelho, no caso de haverem problemas. Com isto pretende-se simular a tomada de decisão do sistema na cadeira de rodas, em que por exemplo, quando um sensor se torna vermelho, o sistema atua na mudança de direção da cadeira na direção oposta do sensor, até se obter uma distância que consiga voltar a torna o sensor verde e assim, poder seguir o seu trajeto normalmente.

É necessário também realçar o modo de comparação dos sensores. Esta comparação divide-se em duas partes, a parte superior e a inferior. Na parte superior, são usados os seis primeiros sensores. Estes sensores são usados quando é feita a deteção de objetos, ou seja, quando o Kinect está posicionado e observa a vista de frente. Na parte inferior são usados os últimos três sensores, aquando a análise da superfície do chão para a deteção de buracos ou desníveis.

São agora apresentados vários resultados da deteção e da atuação através da sinalização dos sensores.

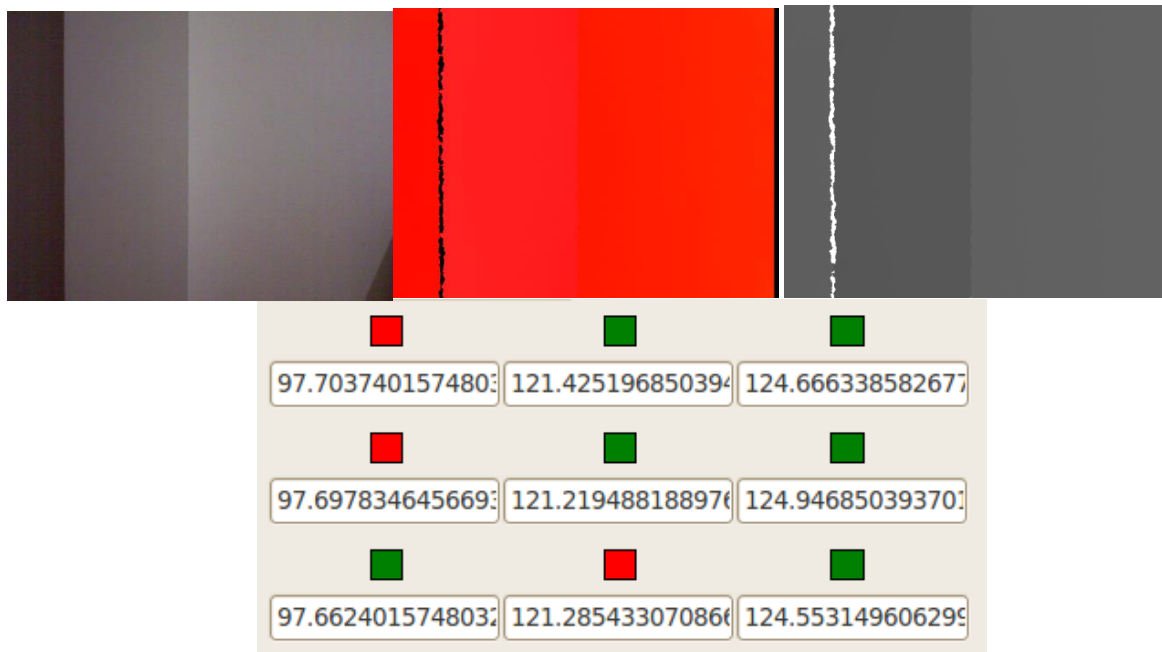


Figura 30 – Detecção e sinalização de um obstáculo do lado esquerdo

Como mostra a figura 30, existe um obstáculo do lado esquerdo da imagem. Anteriormente foi explicado também o modo de comparação entre os sensores, sendo esse modo a base de explicação desta situação. Assim, como se trata de uma análise a um objeto com uma vista de frente (é possível ter esse conhecimento observando os valores dos três sensores do lado direito, esses sensores tem todos o mesmo valor, o que significa que existe uma superfície paralela ao Kinect) vamos usar apenas os primeiros seis sensores. Deste modo, é possível observar os valores dos sensores, os do lado esquerdo tem valores por volta dos 97 cm, os do meio aproximadamente 121 cm e do lado direito 124 cm. O código seguinte representa o modo de comparação dos sensores.

```
if ((media2-media1)>20) then
ShapeS1.Brush.Color := clRed;
```

Neste excerto de código determina-se que se a diferença entre o sensor 2 e o sensor 1 for maior que 20 cm, então o sensor 1 torna-se vermelho, ou seja, se existir algo que faça uma diferença de distância superior a 20 cm em relação à superfície então existe um obstáculo na zona do sensor 1, fazendo assim com que o sinal se torne vermelho. Este procedimento é

aplicado a todos os sensores fazendo assim com que sempre que haja diferenças de 20 cm, é provável que exista algum tipo de objeto.

Os últimos três sensores já têm uma sinalização diferente, que se explica de seguida. Imaginemos agora que não se trata de uma vista em frente, mas que o Kinect está direcionado para o chão. A comparação dos sensores já é feita de modo diferente.

```
if ((media8-media7)>20) then  
ShapeS8.Brush.Color := clRed;
```

Este código faz a diferença entre o sensor 8 e o sensor 7, mas neste caso, se for superior a 20 cm é o sensor 8 que se torna vermelho, isto significa que o sensor 8 está mais distante o que poderá indicar a existência de um buraco na superfície. A figura 31 representa agora uma situação em que de facto o Kinect está direcionado para o chão.

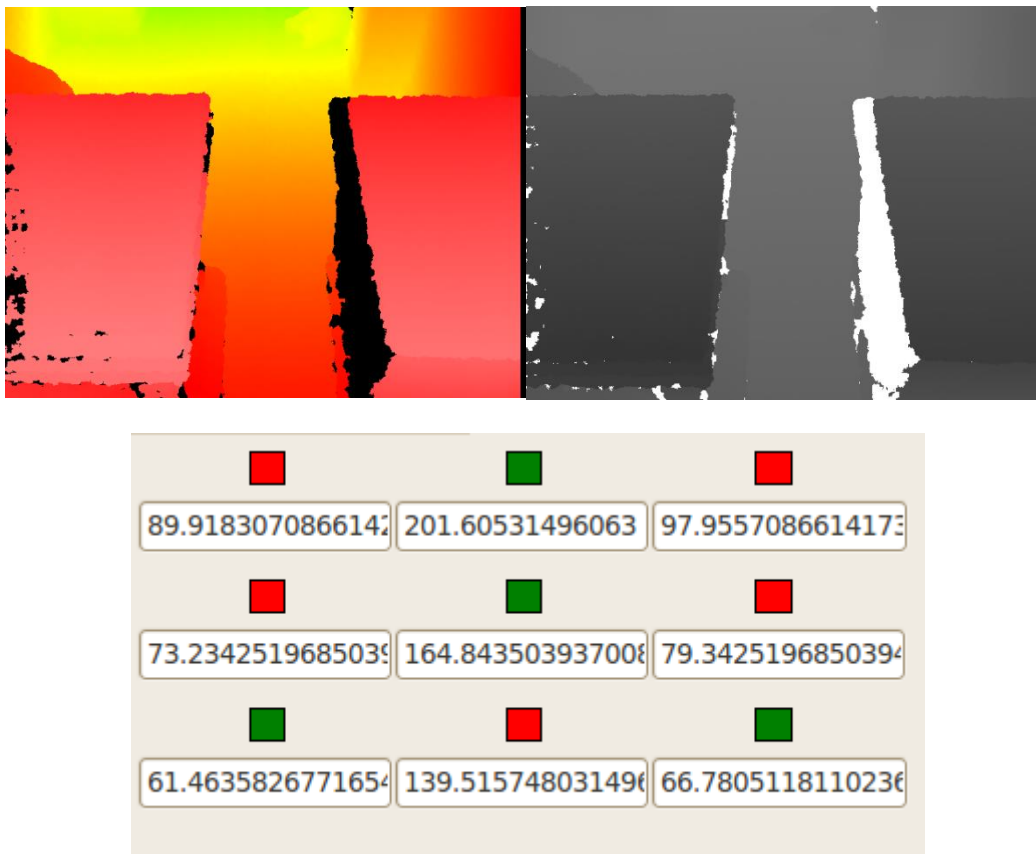


Figura 31 – Detecção e sinalização de um buraco

Nesta situação é possível verificar realmente o explicado anteriormente. Analisando apenas os últimos três sensores, verifica-se a existência de um buraco no centro da imagem onde se situa o sensor 8, assim em comparação com os seus vizinhos, verifica-se que existe uma diferença superior a 20 cm, o que sinaliza o sensor a vermelho. Pode-se também verificar que o Kinect está orientado para o chão ao analisar os valores dos outros sensores. É possível observar que a distância vai crescendo dos últimos sensores para os primeiros, o que significa que o Kinect não está paralelo à superfície.

Capítulo 5

5 Conclusão e Trabalho Futuro

Neste capítulo são apresentadas as principais conclusões e feito um levantamento de possíveis trabalhos e desenvolvimentos futuros para a aplicação desenvolvida neste projeto.

5.1 Trabalho Realizado

Cada vez mais são desenvolvidos projetos e trabalhos de visão computacional recorrendo ao Microsoft Kinect. Este dispositivo possui características bastante interessantes para o desenvolvimento deste tipo de trabalhos e por isso consolidou-se neste projeto todo o conhecimento teórico e modo de funcionamento do equipamento. Com isto foi possível perceber a sua construção e tirar o máximo possível das suas características através do estudo deste e de outros tipos de equipamentos existentes.

Para aplicações de visão computacional é importante que o sistema seja o mais simples possível, mais intuitivo para ser acessível a todos os tipos de utilizador e de rápido processamento em tempo real. O *software* Lazarus oferece essa possibilidade, que em conjunto com o Kinect proporcionam uma solução bastante interessante.

No contexto do problema estudado, foi possível o desenvolvimento de uma aplicação base para o aumento de segurança de uma cadeira de rodas. Assim, o sistema é autónomo e em tempo real consegue calcular distâncias e detetar a presença de obstáculos ou desníveis no meio observado, sinalizando assim o que se observa, podendo a partir dessa sinalização, tomar decisões sobre o que fazer seguidamente.

5.2 Trabalho Futuro

Como desenvolvimento futuro há alguns aspetos a melhorar e até a inovar. A maior precisão de cálculo com um aumento do número de sensores virtuais talvez possa ser uma boa solução se forem necessárias observações de mais regiões da imagem.

Mais concretamente para esta aplicação, a parte de deteção de obstáculos pode melhorar. Encontrar uma forma de juntar o processo de cálculo superior, de deteção de objetos, com o cálculo inferior, de deteção de buracos, e assim em conjunto o sistema conseguir analisar todas as situações em simultâneo.

Outra evolução do trabalho é a integração em ambiente real. Aplicar o Kinect e o sistema funcionar aplicado numa cadeira de rodas dará um avanço enorme à aplicação. Assim, com o sistema de sinalização presente, consegue-se manipular todas as ações da cadeira de rodas e definir o seu trajeto e movimento. Assim, dependendo do que o Kinect observa, o sistema processa os dados e aciona diretamente comandos no sistema de movimentação da cadeira de rodas, desviando-se assim autonomamente dos obstáculos, travando nas descidas, aumentando a velocidade nas subidas e parando com a presença súbita de algum objeto.

Referências

- [1] Yao-Jen Chang, Shu-Fang Chen, Jun-Da Huang, A Kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities, em Research in Developmental Disabilities 32, páginas 2566-2570, 2011.
- [2] Microsoft, Xbox 360 Kinect. www.xbox.com/kinect
- [3] Roland Siegwart e Illah R.Nourbakhsh, Introduction to Autonomous Mobile Robots. Bradford Company, Scituate, MA, USA, 2004.
- [4] Bernd Jahne, Digital Image Processing, 2005.
- [5] Paulo Costa, Operação de pick and place adaptativa em ambientes pouco estruturados, Faculdade de Engenharia da Universidade do Porto, 2012
- [6] Fernando Monteiro, notas de apoio da disciplina de Visão Artificial, Instituto Politécnico de Bragança, 2013.
- [7] S. Hsu, S. Acharya, A. Rafii, e R. New. Performance of a time-of-flight range camera for intelligent vehicle safety applications.
- [8] Paul Wootton. Out of control gaming. www.popsci.com/gear-gadgets/article/2008-05/out-control-gaming
- [9] PrimeSense www.primesense.com
- [10] iPiSoft. Asus xtion vs ms Kinect comparison. http://wiki.ipisoft.com/ASUS_Xtion_vs_MS_Kinect_Comparison.
- [11] Asus. Asus xtion pro live. http://www.asus.com/Multimedia/Motion_Sensor/Xtion_PRO_LIVE/.
- [12] Jan Smisek, Michal Jancosek, e Tomas Pajdla. 3d with kinect. Em Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on, páginas 1154–1160, nov. 2011.
- [13] Microsoft. MSDN - Kinect for Windows Sensor, 2012. <http://msdn.microsoft.com/en-us/library/jj131033.aspx>

- [14] The kinect patent - method and system for object reconstruction, 2005, <http://www.wipo.int/patentscope/search/en/WO2007043036>
- [15] L. Cruz, D. Lucio e L. Velho. Kinect and rgb-d images: challenges and applications. páginas 36 – 49, Los Alamitos, CA, USA, 2012.
- [16] OpenKinect. libfreenect drivers. http://openkinect.org/wiki/Main_Page
- [17] PrimeSense. Openni drivers. <http://openni.org/>.
- [18] F. Soltani, F. Eskandari, S. Golestan, Developing a gesture-based game for deaf/mute people using microsoft Kinect, em Sixth International Conference on Complex, Intelligent, and Software Intensive Systems, 2012.
- [19] YDreams. YScope : YDreams and Hospital Santa Maria da Feira Launch Revolutionary App in the Health and Technology Sectors, Julho 2012 <http://www.ydreams.com/index.php#/en/aboutus/media/whatsup/2012/YSCOPEYDREAMS/HOSPITALSANTAMARIA/>
- [20] Miguel Correia, Reconhecimento de elementos da lingua gestural portuguesa com Kinect, Faculdade de Engenharia da Universidade do Porto, 2013.
- [21] OpticalFlow. Kinect - insanely interesting, 2011.
<http://opticalflow.wordpress.com/2011/02/27/kinect-insanely-interesting>
- [22] Júlio Ferreira, Demonstrador de condução autónoma, Faculdade de Engenharia da Universidade do Porto, 2012.
- [23] Stephane Magnenat. Raw depth to meters.

