UNIVERSIDADE DE COIMBRA

Intellicare

Algoritmos de deteção de comportamento de indivíduos com autismo – Análise comparativa

Miguel Proença Brás Ferreira

Mestrado em Engenharia Biomédica

Faculdade de Ciências e Tecnologias – Departamento de Física

Julho 2013

Intellicare

Algoritmos de deteção do comportamento de autistas – Análise comparativa

Departamento de Física
Faculdade de Ciências e Tecnologias
Universidade de Coimbra

Dissertação orientada por:
Andreia Carreiro
Prof. Dr. Carlos Correia
Prof. Dr. José Basílio Simões

Miguel Proença Brás Ferreira

Agradecimentos

Chegou a altura de agradecer a todos os envolvidos neste projeto. Sem a ajuda destas pessoas e instituição não seria possível terminar com sucesso este ano de trabalho.

Em primeiro lugar quero agradecer à ISA *Intelligent Sensing Anywhere*, SA e aos seus colaboradores por me terem acolhido ao longo do desenvolvimento deste projeto e proporcionado todas as condições para que pudesse ser desenvolvido.

Um obrigado à Engenheira Andreia Carreiro que me indicou a direção certa sempre que me encontrava um pouco perdido. Obrigado por todas as orientações e soluções para os problemas que iam surgindo.

Agradeço também aos meus amigos que sempre me apoiaram em momentos de descontração e de alívio de *stress*. Aqui incluo também os meus colegas com os quais partilhei a casa ao longo do meu percurso académico.

Resta-me agradecer à minha família que tanto me apoiou, me ofereceu condições de estabilidade imprescindíveis ao desenvolvimento do trabalho ao longo do percurso académico.

Resumo

A elevada prevalência de doenças relacionadas com o autismo e a necessidade de descoberta desta patologia de difícil diagnóstico e convivência, levou a que seja imprescindível o desenvolvimento de mecanismos de deteção e avaliação da atividade de pacientes. Esta tese tem como principal objetivo colmatar estas necessidades.

Os algoritmos testados são direcionados à melhoria da monitorização de doentes com autismo e com necessidades especiais. Estes têm como principal objetivo a utilização numa casa direcionada à terapia e convivência de doentes com autismo. Através destas técnicas será possível melhorar a qualidade de vida dos pacientes e responder a emergências que possam surgir.

Os dados provenientes de sensores como a câmara *Kinect* vão servir para detetar movimentos estereotipados característicos de doenças do espectro do autismo. Assim podem ser monitorizadas atividades anormais e por consequência antecipar algum tipo de anomalia. Por outro lado será possível avaliar a evolução do doente ao longo de uma série de tratamentos e terapias.

Neste projeto foi avaliada a performance de vários algoritmos de deteção e avaliação de movimentos adaptados às necessidades presentes. Foi também conduzido um estudo sobre as tecnologias de reconhecimento de expressões faciais e identificação de pessoas pela face.

Os algoritmos foram testados com movimentos de pessoas sem doenças do espetro do autismo devido à maior facilidade de trabalhar neste contexto. Posteriormente transpõem-se as tecnologias para onde forem necessárias.

Abstract

The great prevalence of Autism Spectrum Disorders and the need of discover of new information about this kind of diseases of difficult coexistence led to the development of behavior detection mechanisms. This master's thesis aims to address these needs.

The tested algorithms are intended to improve the monitoring of Autism Spectrum Disorders patients. These are meant to be used in a house directed to therapy and cohabitation of autistic patients. Through these techniques should be possible to improve patient's quality of life and answer any emergencies that may arise.

The data collected by the sensors such as the Kinect by Microsoft will be useful to detect characteristic stereotyped movements. Therefore abnormal activity can be monitored and consequently anticipate some kind of anomaly. On the other hand it will be possible to evaluate the evolution of the patient over a series of treatments and therapies.

In this project we adapted and evaluated the performance of various algorithms for detection and evaluation of movements. Has also been conducted a study on facial expressions and face recognition.

The algorithms were tested with people without Autism Spectrum Disorders due to ease of working in this context. Later we will transpose the technologies to where will be necessary.

Índice

Agradecimentos ii

Resumo iii

Abstract iv

Índice v

Lista de acrónimos viii

Índice de Figuras ix

Índice de Tabelas x

1. Introdução 1

    1.1. Motivação 1
    1.2. Objetivos 2
    1.3. Âmbito 2
    1.4. Audiência 3
        1.4.1. Orientação 3
        1.4.2. Organizações envolvidas 3
    1.5. Estrutura da tese 3

2. Autismo 5

    2.1. Sintomas 5
        2.1.1. Dificuldades motoras 5
        2.1.2. Interações sociais e comportamentos 5
        2.1.3. Movimentos estereotipados 6
        2.1.4. Convulsões 7

3. Estado da Arte 8

    3.1. Soluções integradas 8
    3.2. Soluções isoladas 10
    3.3. Conclusão 13

4. Arquitetura e Especificações do sistema 14

    4.1. Arquitetura física 14
    4.2. Arquitetura lógica 15
    4.3. Arquitetura da aplicação 17
    4.4. Especificações Técnicas do sistema 19

4.5. *Kinect SDK* 21
5. Deteção de gestos 22
5.1.1. Recolha de dados 22
5.1.2. Tratamento de dados 25
5.1.3. Armazenamento dos dados 26
5.2. Algoritmo *Hidden Markov Model* 27
5.2.1. Algoritmo *Viterbi* 30
5.2.2. Algoritmo Baum-Welch (Forward-Backward) 33
5.3. Implementação do algoritmo HMM 37
5.3.1. Framework Accord.NET 37
5.3.1. Obtenção de resultados 38
5.3.2. Resultados obtidos 38
5.3.3. Desempenho *online* 44
5.4. Algoritmo DTW (*Dynamic Time Warping*) 45
5.4.1. DTW Standard 45
5.4.1. DTWMean 48
5.5. Implementação do algoritmo DTW 48
5.5.1. Análise dos dados 49
5.5.2. Obtenção de resultados 49
5.5.3. Treino com DTW Standard 50
5.5.4. Treino com DTWmean 52
5.5.5. Desempenho *online* 53
5.6. Discussão 54
6. Deteção de faces e expressões faciais 56
6.1. Deteção de expressões faciais 56
6.1.1. Recolha de dados 56
6.1.1. Resultados e discussão 59
6.2. Deteção de pessoas pela face 59
6.2.1. Técnicas de reconhecimento de faces 59
6.2.2. Recolha de *features* pelo *Kinect* 60
6.2.3. Extração de *features* com uma câmara 61
6.2.4. Tratamento dos dados 62
6.2.5. Resultados obtidos 64
6.2.6. Discussão 64
7. Conclusão 66

7.1. Trabalho futuro 67
7.2. Publicações associadas 67

Referências 68

Anexos i

Anexo A i
Anexo B xi

Lista de acrónimos

2D: Duas Dimensões

3D: Três Dimensões

AAL: *Ambient Assisted Living*

APCC: Associação de Paralisia Cerebral de Coimbra

ASD: *Autism Spectrum Disorders*

AU: *Animation Units*

DTW: *Dynamic Time Warping*

FN: Falsos Negativos

FP: Falsos Positivos

HMM: *Hidden Markov Models*

IBILI: *Institute for Biomedical Imaging and Life Sciences*

ISA: *Intelligent Sensing Anywhere*

ISR: Instituto de Sistemas e Robótica

LDA: *Linear Discriminative Analysis*

MHI: *Motion History Image*

OpenCV: *Open source computer vision*

PCA: *Principal Component Analysis*

RAM: *Random Access Memory*

RGB: *Red Green Blue*

SDK: Software Development Kit

VP: Verdadeiros Positivos

VN: Verdadeiros Negativos

XML: *eXtensible Markup Language*

Índice de Figuras

Figura 3.1: Mavhome [25] 9

Figura 3.2: Ubiquitous Home [27] 10

Figura 4.1: Arquitetura física 14

Figura 4.2: Fluxo de informação 16

Figura 4.3:Arquitetura da Aplicação 17

Figura 4.4:Arquitetura da Base de Dados 18

Figura 4.5:Especificações do *Kinect* 20

Figura 4.6: Funcionamento do sensor de profundidade [34] 21

Figura 5.1: : Motion History Image [39] 23

Figura 5.2: Articulações identificadas pelo *Kinect* [34] 24

Figura 5.3: Sistemas de coordenadas do *Kinect* [34] 25

Figura 5.4: Tratamento dos dados 26

Figura 5.5: Excerto XML 27

Figura 5.6: Dependências no modelo HMM [41] [42] 28

Figura 5.7 :Funcionalidades do algoritmo DTW [52] 45

Figura 5.8 : Diagrama de custos com caminho ideal [52] 46

Figura 5.9 :Diagrama de caminho ideal [52] 47

Figura 5.10: Esquema de análise da variável *tab* 49

Figura 6.1: Conjunto de pontos monitorizados [34] 57

Figura 6.2: Posição da cabeça [34] 58

Figura 6.3: Cara triste (AU4=+1) [34] 58

Índice de Tabelas

Tabela 3.1: Resumo das soluções 12

Tabela 5.1:probabilidades de emissão 31

Tabela 5.2: HMM Baum-Welch com 5 estados escondidos sem *threshold* 39

Tabela 5.3: HMM Baum-Welch com 10 estados escondidos sem *threshold* 39

Tabela 5.4 : HMM Baum-Welch com 5 estados escondidos com *threshold* de sensibilidade 5 . 40

Tabela 5.5: HMM Baum-Welch com 5 estados escondidos com *threshold* de sensibilidade 0.1 40

Tabela 5.6 : HMM Baum-Welch com 10 estados escondidos com *threshold* de sensibilidade 0.1 41

Tabela 5.7 : HMM Baum-Welch com 10 estados escondidos com *threshold* de sensibilidade 0.05 42

Tabela 5.8 : *Viterbi* com 5 estados escondidos com *threshold* de sensibilidade 0.1 43

Tabela 5.9 :*Viterbi* com 10 estados escondidos com *threshold* de sensibilidade 0.1 43

Tabela 5.10 : *Viterbi* com 10 estados escondidos com *threshold* de sensibilidade 0.05..... 44

Tabela 5.11 : Algoritmo DTW global *threshold*:2; first *threshold*:6; deslocamento máximo:5;distância mínima:20..... 50

Tabela 5.12 Algoritmo DTW global *threshold*:1; first *threshold*:3; deslocamento máximo:3;distância mínima:12..... 51

Tabela 5.13 : Algoritmo DTW global *threshold*:0,6; first *threshold*:2; deslocamento máximo:2;distância mínima:10..... 51

Tabela 5.14 : Algoritmo DTWmean global *threshold*:0,6; first *threshold*:2; deslocamento máximo:2;distância mínima:10..... 52

Tabela 5.15 : Algoritmo DTWmean global *threshold*:1; first *threshold*:3; deslocamento máximo:3;distância mínima:12..... 53

Tabela 5.16: Desempenho dos algoritmos..... 54

Tabela 6.1: Resultados na deteção de pessoas pela face 64

1. Introdução

1.1. Motivação

Este projeto foca-se no desenvolvimento e investigação de tecnologias que permitam a monitorização de autistas num ambiente caseiro.

A prevalência para o autismo em Portugal é de 10 pessoas em 10000, 2.5 em 10000 para a Síndrome de *Asparger* e 30 em 10000 para perturbações no campo do autismo [1].

O autismo não é apenas uma doença, mas um distúrbio complexo, com múltiplas etiologias e graus de desenvolvimento [2]. Esta condição evolui com a idade e prolonga-se para o resto da vida [3]. Assim podemos dizer que é importante recolher informações acerca deste tipo de pacientes de forma a melhorar a sua qualidade de vida ou até mesmo os métodos de diagnóstico.

Indivíduos com este tipo de patologia apresentam diferentes fraquezas e forças e mesmo o próprio doente pode apresentar diferentes comportamentos face à mesma situação em dias diferentes [4]. Assim torna-se essencial detetar vários parâmetros, portanto podemos dizer que a integração de informação a partir dos vários tipos de tecnologias de sensores é importante.

Com o aumento da quantidade de tecnologias que podem ser aplicadas à saúde, é necessário voltar as atenções para pacientes com distúrbios do espectro do autismo.

Os algoritmos moldados e testados neste projeto permitem analisar os comportamentos dos pacientes. Este processo é feito através da identificação e reconhecimento de movimentos e expressões faciais.

Finalmente o sistema irá identificar a ocorrência de um maior número de movimentos estereotipados ou comportamentos anormais e irá emitir um alarme para o cuidador da casa.

Este trabalho foi realizado no âmbito do projeto *HomeTech*, financiado pelo *QREN*, cujo objetivo é desenvolver um sistema de vigilância inteligente para uma casa habitada com pessoas com limitações, nomeadamente no espectro das doenças do autismo. Contribui assim

na investigação na área de algoritmos de análise de vídeo assim como na produção de documentação necessária.

1.2. Objetivos

Distúrbios do espectro do autismo são condicionantes que afetam a socialização, comunicação e comportamento dos pacientes. Nos últimos anos houve uma chamada de atenção para este tipo de doenças devido à evolução no diagnóstico e às óbvias consequências sociais.

A monitorização de pacientes em casa resulta da necessidade de se recolher dados durante um grande período de tempo de forma a melhorar a qualidade dos dados e consequentemente da sua análise. Este tipo de recolha de dados não pode ser feito no consultório do médico, e tem a vantagem de ser não invasivo.

Esta tese tem como objetivo encontrar soluções para analisar o comportamento natural dos pacientes e, consequentemente possibilitar descobertas acerca da doença, fornecer segurança e identificar problemas que advêm da condição.

Este projeto resume-se à monitorização completa de uma casa onde vivem pacientes autistas em conjunto com um cuidador.

1.3. Âmbito

O presente projeto foi desenvolvido de forma a finalizar o Mestrado Integrado em Engenharia Biomédica a carga da Universidade de Coimbra no ano letivo 2012/2013.

Este insere-se no âmbito do projeto *HomeTech* que conta com a participação da ISA que possui uma vasta experiência em projetos de investigação (nacionais e Europeus), desenvolvimentos industriais na área dos sistemas de tele-contagem e tele-atuação, eficiência energética e mais recentemente na área da saúde, segurança e domótica (*Ambient Assited Living*), com forte presença no mercado nacional e internacional (com enfâse no mercado Europeu). Pretende integrar os desenvolvimentos deste projeto no seu *roadmap* tecnológico, robustecendo as suas soluções para residências nas áreas da segurança, pretende ainda envolver a sua spin-off *Intellicare* na exploração da plataforma Cuidador integrando

com a sua plataforma *OneCare*, constituindo assim uma oferta mais completa na área do AAL (*Ambient Assisted Living*) [5].

1.4. Audiência

Os alvos principais desta tese serão os supervisores e júris envolvidos neste projeto, a comunidade de Engenheiros Biomédicos assim como da área da Informática.

1.4.1. Orientação

O presente projeto é direcionado para a elaboração de uma tese final de mestrado de um aluno do mestrado integrado em Engenharia Biomédica em colaboração com as entidades envolvidas.

Os membros envolvidos neste projeto foram:

Nome	**Tarefa**	**Contacto**
Miguel Ferreira	Estudante	miguelguarda1@gmail.com
Andreia Carreiro	Supervisora	acarreio@isa.pt
Prof. Dr. Carlos Correia	Orientador	correia@lei.fis.uc.pt
Prof. Dr. José Basílio Simões	Orientador	jbasilio@lei.fis.uc.pt

1.4.2. Organizações envolvidas

O projeto foi facultado pela empresa ISA Intelligent Sensing Anywhere, SA no âmbito do projeto HomeTech que resulta da pareceria com a BrainEyes e a Universidade de Coimbra com as Faculdades de Medicina (IBILI) e Faculdade de Ciência e Tecnologia (ISR), contando ainda com o apoio fundamental da APCC – Associação de Paralisia Cerebral de Coimbra para acompanhar o projeto e disponibilizar as instalações para o piloto em ambiente real.

1.5. Estrutura da tese

Nesta parte da introdução são brevemente descritos os capítulos que compõem esta tese.

Capítulo 1: É composto pela Introdução da tese na qual se incluem a Motivação, Objetivos, Âmbito, Audiência e Estrutura da tese.

Capítulo 2: É feito um enquadramento acerca da doença de Autismo.

Capítulo 3: Este capítulo inclui o estado da arte dos sistemas de monitorização inteligentes de forma a conseguir-se chegar à conclusão do *hardware* a utilizar.

Capítulo 4: Nesta fase é indicada uma possível arquitetura do sistema final a ser testado assim como um estudo acerca das especificações do sensor. Inclui a arquitetura lógica, física e da aplicação.

Capítulo 5: Neste capítulo são abordados os algoritmos de deteção de gestos de uma forma elaborada. De seguida é feita uma breve descrição da implementação e por fim os resultados obtidos são apresentados e discutidos.

Capítulo 6: É descrita uma maneira de reconhecer pessoas através da face assim como a deteção de expressões faciais.

Capítulo 7: São tiradas as conclusões finais do trabalho elaborado ao longo do projeto.

2. Autismo

Neste capítulo é efetuada a contextualização no âmbito das doenças do espetro do autismo.

Doenças do espectro do autismo ou *Autism Spectrum Disorders* (ASD) são uma preocupação urgente tanto em termos de saúde pública como em termos de despesa pública [6].

2.1. Sintomas

Apesar dos sintomas diferirem de pessoa para pessoa, na generalidade existem três áreas nas quais existem alterações. Estas áreas são tipicamente a interação social, comunicação e performance motora [7]. As dificuldades a nível social muitas vezes prolongam-se para o resto da vida do paciente [8] [9].

Em portadores de doenças ASD existe uma diminuição do uso da linguagem ou comunicação social não-verbal, baixo contato visual, falta de partilha de interesses, falta de reciprocidade emocional, linguagem e comportamentos motores repetidos e estereotipados, tratos obsessivos ou compulsivos, preferência por objetos em detrimento de pessoas e falta de imaginação [10].

2.1.1. Dificuldades motoras

As dificuldades motoras muitas vezes traduzem-se em falta de jeito na execução de tarefas, dificuldades no planeamento, défices no controlo fino e grosseiro dos movimentos e falta de fluência e coordenação [11].

2.1.2. Interações sociais e comportamentos

As crianças em desenvolvimento são seres sociais por natureza. Uma criança com autismo ou qualquer outra patologia do espetro do autismo não desenvolve capacidades de socialização de uma maneira tão fluente e natural [12].

Pelos 8 ou 10 meses, uma criança sem este tipo de patologias já responde quando se chama pelo seu nome, interessa-se pelo que as pessoas dizem ou fazem e conseguem emitir alguns sons de maneira a comunicar. Por outro lado, uma criança autista não desenvolveu nenhuma

destas capacidades até esta idade. Pelos 2-3 anos, uma criança autista tem dificuldade em jogar jogos sociais, não imita as ações de outras pessoas e prefere brincar sozinho [13].

As dificuldades ao nível da interpretação e demonstração de emoções também são evidentes o que se traduz em falhas na expressão facial e corporal. Por exemplo uma criança pode não conseguir perceber que fez algo de errado ao não conseguir interpretar que os pais estão aborrecidos. Da mesma maneira que não consegue exprimir de uma forma normal o carinho pelos pais. Estas debilidades podem também desencadear reações consideradas imaturas [13] [14].

Em termos de comunicação sabe-se que as crianças portadoras de ASD têm um desenvolvimento tardio da fala, dificuldades em iniciar ou manter uma conversa, repetem palavras ou frases ouvidas noutra conversa e não conseguem interpretar o tom da mensagem recebida, quer seja uma brincadeira ou ironia. Da mesma forma que não conseguem conferir expressividade às suas declarações [15].

Existem, por sua vez, muitos indicadores de ações em crianças que podem apontar para o diagnóstico de doenças ASD. Entre eles está a falta de contacto visual com outras pessoas, maneiras diferentes de brincar como bater com os brinquedos ou organizá-los por cores, a falha nos gestos de apontar com o dedo e a reprodução de movimentos estereotipados. Doentes de autismo costumam também revelar um interesse exacerbado por um certo tipo de assunto. Normalmente os autistas só comunicam com outra pessoa se esta partilhar o conhecimento sobre o assunto preferido do primeiro.

2.1.3. Movimentos estereotipados

Um movimento estereotipado tem como principais características a invariância, rigidez e repetição, costuma também ser inapropriado para a situação social do indivíduo. Este movimento pode ser traduzido num gesto ou numa fala [14].

A presença de comportamentos restritos e estereotipados é uma característica fundamental de indivíduos com autismo. Este tipo de comportamentos não serve objetivamente para nada, o que diminui o espaço para a aprendizagem de ações úteis. Os movimentos estereotipados são muitas vezes repetitivos e ao mesmo tempo muito debilitantes. Estes são estigmatizantes e são também causa da existência de dificuldades ao nível das interações sociais. As dificuldades aqui mencionadas vão ser determinantes na falta de integração deste tipo de

doentes em escolas normais e em sítios públicos, onde estes movimentos são considerados anormais [16].

Em termos de gestos estereotipados, entre os mais característicos existe o balançar e abanar do corpo e da cabeça, vocalizações irreconhecíveis, bater e acenar com as mãos. Estes estereótipos servem de auto estimulação para o autista [15].

2.1.4. Convulsões

Sabe-se também que a epilepsia com convulsões ocorre em crianças autistas, com idades entre os 5 e 10 anos, com uma taxa de incidência de aproximadamente 26% a 47% [17] [1].

3.Estado da Arte

Neste capítulo vão ser analisadas soluções na área de monitorização inteligente em ambientes especiais. Em primeiro lugar analisam-se as soluções integradas nas quais são utilizados sensores simples. De seguida serão analisadas soluções isoladas.

3.1. Soluções integradas

As soluções integradas consistem na aplicação de soluções isoladas em ambientes controlados como *Smart Homes* que são casas ou ambientes controlados que têm sistemas tecnologicamente avançados que permitem a automação de algumas tarefas, são mais seguras e dispõem de meios de comunicação avançados. Estas servem principalmente para diminuir custos de manutenção e melhorar a qualidade de vida de pessoas com necessidades especiais [18] [19]. Este é o principal objetivo do projeto no qual esta tese se insere.

Existem vários projetos de *smart homes* presentemente implementadas. Alguns exemplos destes projetos são a *Welfare Techno-House* no Japão, *Assisted Interactive Dwelling House* no Reino Unido ou como é o caso da *MavHome* [18] [20].

A *MavHome* é um projeto de pesquisa multidisciplinar na *Washington State University* e na *University of Texas* em Arlington focado na criação de um ambiente doméstico inteligente. O objetivo é ver as *smart home* como agentes inteligentes que se apercebem do seu ambiente através do uso de sensores e conseguem, através de atuadores, influenciar o ambiente da casa. Tem como principais objetivos minimizar os custos de manutenção e maximizar o conforto dos habitantes [21]. No esquema abaixo apresenta-se um resumo dos sensores e tecnologias usadas nesta *smart home* [22].

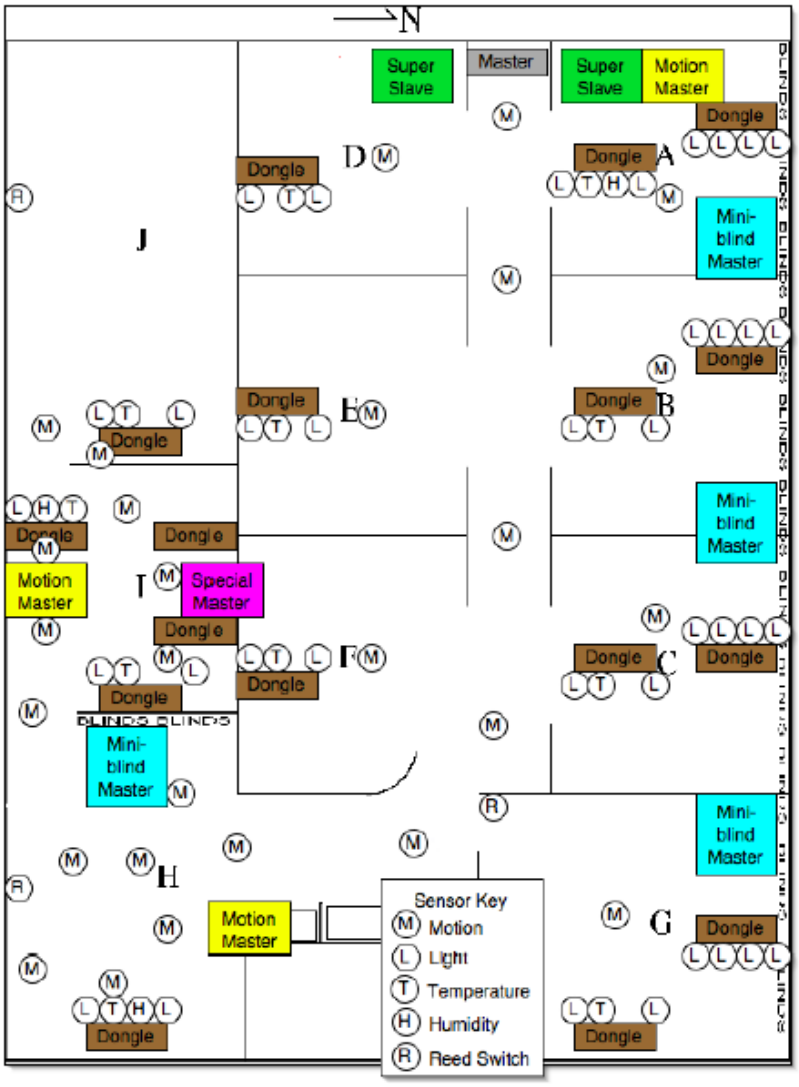


Figura 3.1: Mavhome [25]

A *Ubiquitous Home* é outro projeto de *Smart Home* na qual é utilizada a *smart forniture* que consiste na aplicação de sensores em objetos e mobília que detetam se existe interação com o indivíduo. Assim pode-se medir o nível de interesse do indivíduo por este determinado objeto. Esta tecnologia foi implementada na *Ubiquitous Home* no Japão [23]. Ainda nesta *smart home* são utilizadas câmaras, sensores de pressão, Radio Frequency Identification, sensores infravermelhos, microfones e colunas de som. O esquema abaixo permite uma perceção do esquema da casa [24].

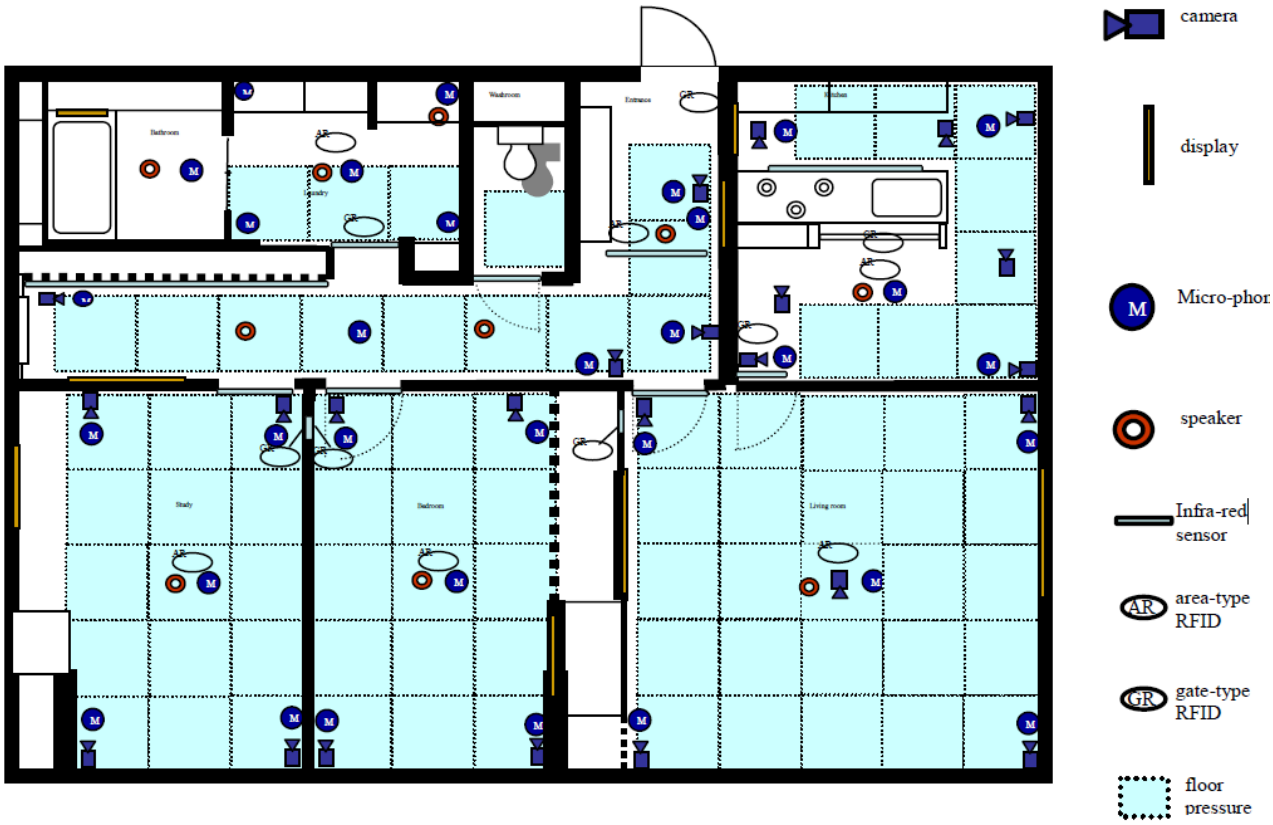


Figura 3.2: Ubiquitous Home [27]

O conceito de *Smart Home* abrange também o conceito de laboratório no qual existe uma evolução constante das tecnologias aplicadas.

3.2. Soluções isoladas

Indivíduos com patologias no espectro do autismo apresentam diferentes fraquezas e forças e mesmo o próprio doente pode apresentar diferentes comportamentos face à mesma situação em dias diferentes [4]. Assim torna-se essencial detetar vários parâmetros, portanto pode-se dizer que a integração de informação a partir dos vários tipos de tecnologias de sensores é importante.

Em termos de relações sociais, é muito difícil a interação entre autistas, sendo que isto só acontece aquando da partilha de um interesse. As amizades são construídas ao longo de muito tempo e com uma evolução lenta do nível de partilha pessoal [25]. Neste caso é possível

utilizar a tecnologia *Smart forniture* de forma a avaliar o nível de interesse que um autista revela por certos tipos de objetos e as tecnologias de *indoor tracking* de forma a identificar a interação entre indivíduos.

Devido à sua complexa interação social, os autistas não podem ser ensinados por pessoas, pois quando há interação entre um professor e um autista começam logo as dificuldades que impedem a concretização da aprendizagem. Por outro lado é possível a aprendizagem para o autista, desde que seja efetuada num ambiente computacional, pois este apresenta um comportamento consistente, estruturado e explícito. Os ensinamentos têm também que ser graduais, pois indivíduos com esta patologia têm dificuldades na generalização de casos. A realidade virtual oferece um cenário seguro, indulgente, controlado e computacional para desenvolver terapias relacionadas com o dia-a-dia (higiene pessoal, comer, atravessar uma passadeira) [4]. Através da utilização da monitorização por vídeo e áudio é possível interpretar movimentos e reações e promover as atividades em realidade virtual, além de que será possível detetar alguma necessidade do indivíduo.

Estima-se que 50% dos autistas nunca chegam a desenvolver capacidades de comunicação oral durante toda a sua vida [1]. Devido a estas falhas na comunicação pode-se dizer que a deteção de quedas também se torna importante em termos de segurança, pois o doente pode-se magoar e não se queixar.

Através da deteção de posturas, transição entre posturas e *tracking* pode-se detetar comportamentos próprios de indivíduos com esta patologia, pois estes apresentam alguma confusão mental que se pode traduzir em ações repetitivas (movimentos estereotipados), entradas e saídas frequentes de compartimentos da casa e irritabilidade [26].

Sabe-se também que a epilepsia com convulsões ocorre em crianças autistas, com idades entre os 5 e 10 anos, com uma taxa de incidência de aproximadamente 26% a 47% [17] [1], portanto seria importante não só a deteção destas situações utilizando o *smart watch* como a monitorização de ritmos cardíacos utilizando o monitor *Holter* e o sensor *Kinect* de forma a detetar movimentos [17].

Como se pode observar, existem muitos equipamentos que podem ser utilizados neste ambiente que permitem a recolha de informação. De forma a avaliar o comportamento dos ocupantes é necessário integrar este conjunto de dados e interpretar. Esta interpretação de dados tem que ser, por sua vez, útil para os profissionais de saúde se aperceberem da condição do paciente. Como a monitorização por vídeo através do sensor do *Kinect* é a que

oferece mais informação em termos de quantidade e diversidade, começa-se por implementar o sistema de videovigilância com estas câmaras. Na tabela abaixo pode-se observar o resumo das soluções isoladas e integradas.

Tabela 3.1: Resumo das soluções

Sensor	Objetivo
Monitor *Holter*	Monitorização ritmo cardíaco
I-*Button*	*Indoor tracking*
Radio frequency identification RFID	
Sistemas ultrassónicos	
Fingerprinting	
GPS	*Outdoor tracking*
Acelerómetro triaxial com giroscópio	Identificação de posturas e transições
Acelerómetro com giroscópio	Deteção de quedas
Acelerómetro *smart watch*	Deteção de convulsões
Câmara de ulta-baixa resolução	Deteção de perigos na cozinha
Sensores de movimento	Deteção de quedas, posturas, movimentos, *tracking*
Sensores de pressão	
Comutadores de contacto	
Sensores infravermelhos	
Monitorização áudio	
Monitorização vídeo	
Smart forniture	Medir o nível de interesse por algum objeto ou peça de mobília
Smart Condo	*Smart Homes* nas quais são testados todo o
TigerPlace	

Ubiquitous Home	tipo de sensores

3.3. Conclusão

O sensor *Kinect* trás muitas vantagens para a videovigilância, pois é muito mais que uma simples câmara, como é descrito na próxima fase do projeto. Em concorrência com o *Kinect*, temos um sensor parecido, o *Leap Motion*, que faz o mesmo que o *Kinect*, mas com mais precisão e é mais barato, mas tem um alcance muito inferior, logo é mais indicado para detetar movimentos das mãos. Estes movimentos detetados são mais utilizados para controlo de computadores ou sistemas *touch screen*, sem interação física entre a pessoa e o aparelho [27].

O *Kinect* apresenta ainda a vantagem de ser *open source* a nível comercial, o que trás uma grande vantagem no âmbito dos interesses da ISA [28].

Muitas vezes, a falta de uso de protocolos de comunicação limita o uso deste tipo de *smart homes* na área da saúde. Logo, de forma a integrar a informação em ambiente hospitalar seria interessante também implementar o protocolo de comunicação HL7 [29] [30].

4. Arquitetura e Especificações do sistema

Nesta fase pode-se começar a escolher algumas das soluções existentes no mercado e de seguida perspetivar uma forma de implementar o sistema de monitorização com o objetivo de, a partir das características do *hardware*, a extrair os melhores resultados.

4.1. Arquitetura física

O sistema destina-se à monitorização de pessoas com patologias do espectro do autismo que desta forma podem beneficiar de um acompanhamento mais adequado a partir da maior quantidade informação adquirida pelos profissionais de saúde. Os pacientes podem também, através de alguns níveis de interatividade, usufruir de terapia.

De forma a maximizar as potencialidades do sistema de vigilância, vai ser necessário instalar sensores *Kinect* ao nível da cabeça de forma a se usar o modo de proximidade que por sua vez é essencial para a deteção de expressões faciais. Os gestos são detetados no modo normal da câmara e a deteção de voz é independente do modo de operação da imagem.

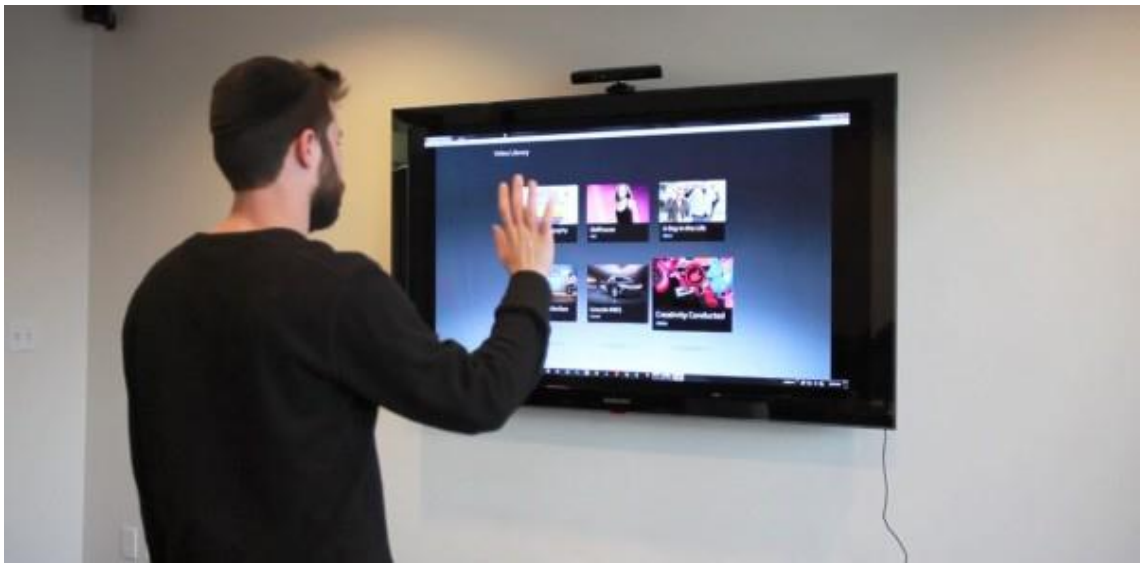


Figura 4.1: Arquitetura física

4.2. Arquitetura lógica

Os dados são recolhidos e processados de forma a possibilitar a formação de conclusões acerca do comportamento dos intervenientes. Para se poder recolher informações da câmara é necessário recorrer ao *Kinect SDK*, pois a partir deste é possível utilizar os serviços oferecidos.

Assim existem aplicações que permitem recolher os dados em bruto do sensor. Estas têm que correr num computador com os requisitos mínimos: processador *dual core 2.66-GHz* de 64 ou 32 bits, porta *USB* 2.0, 2GB *RAM* e é necessário um *Kinect* para Windows. Em termos de software é necessário o *Microsoft Visual Studio* 2010 e o *.NET Framework 4.0* alem do *Kinect SDK*. Assim conclui-se que é essencial a utilização de um computador sempre ligado ao sensor.

Os dados podem então, de seguida, ser submetidos a algoritmos de análise de vídeo que facilitam a obtenção dos dados que se pretende.

Na Figura 4.2 pode-se observar um esquema mais completo do que poderá ser o fluxo de informação desde o sensor *Kinect* até à aplicação final.

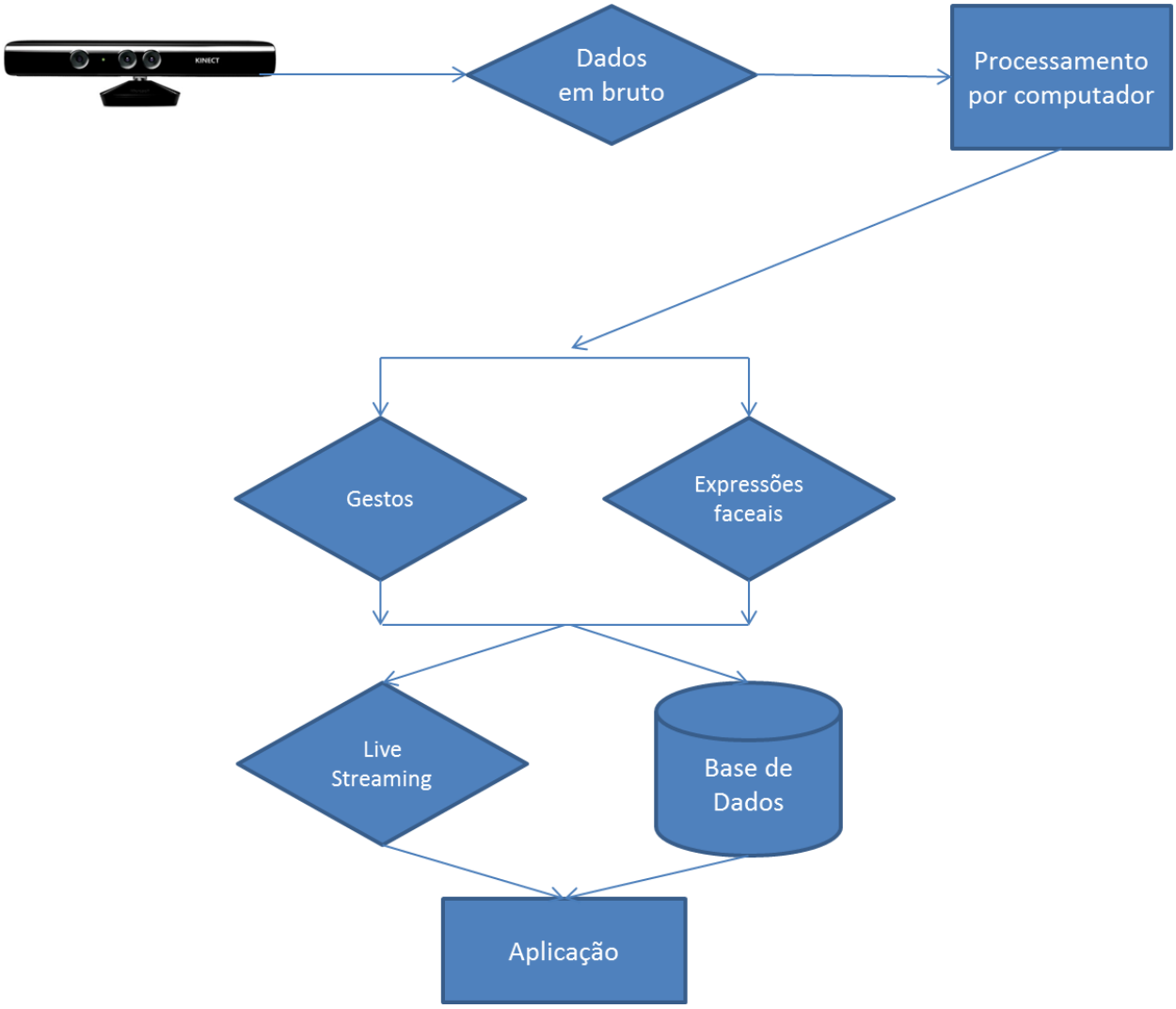


Figura 4.2: Fluxo de informação

Como se pode observar primeiramente os dados em bruto são recolhidos do sensor *Kinect* através de um computador onde se efetua o processamento dos dados. De seguida é necessário detetar os gestos e expressões faciais do paciente em frente do sensor.

As informações retiradas deste processo de reconhecimento são armazenadas numa base de dados, mas também podem ser observadas em tempo real.

4.3. Arquitetura da aplicação

De seguida é analisada a arquitetura da aplicação de forma a serem disponibilizados todos os serviços da Figura 4.. A interface da aplicação está disponível no Anexo B.

Figura 4.3:Arquitetura da Aplicação

A aplicação vai disponibilizar vários serviços como se pode observar na Figura 4.3. O utilizador deverá escolher que tipo de tarefa quer executar. Antes do reconhecimento de gestos é determinante recolher alguns dados dos pacientes que possam ser avaliados. Estes são guardados na tabela *Data* da base de dados da Figura 4.4. Como se pode observar, aqui são recolhidos os dados do nome do paciente, data de nascimento, e número de identificação. O campo *P_Id* é atribuído automaticamente sempre que se adiciona um novo paciente. Este representa também a chave primária da tabela que, por sua vez se relaciona com a coluna com o mesmo nome na tabela *Patient1* e na tabela *Expressions*. A relação é de um para

infinito, ou seja um paciente registado na tabela *Data* pode fazer muitos movimentos ou expressões que são gravados na tabela *Patient1* ou *Expressions*.

De seguida é necessário fazer pelo menos uma gravação de um gesto que se queira identificar, quanto mais gravações se fizerem do mesmo gesto mais fácil será obter bons resultados na deteção. Estes são guardados num ficheiro XML para que não se percam numa futura utilização. De seguida, escolhendo que paciente se está a avaliar (aquele que se encontra em frente da câmara) é possível proceder ao reconhecimento dos gestos, desde que previamente gravados. Estes dados são automaticamente guardados na tabela *Patient1* da base de dados na Figura 4.4. Em conjunto com o movimento detetado através do algoritmo é também guardada a data e hora a que este ocorreu em conjunto com o respetivo paciente que o efetuou. O mesmo acontece quando se avalia expressões faciais, sendo que nesta parte não é necessário qualquer tipo de treino de algoritmo.

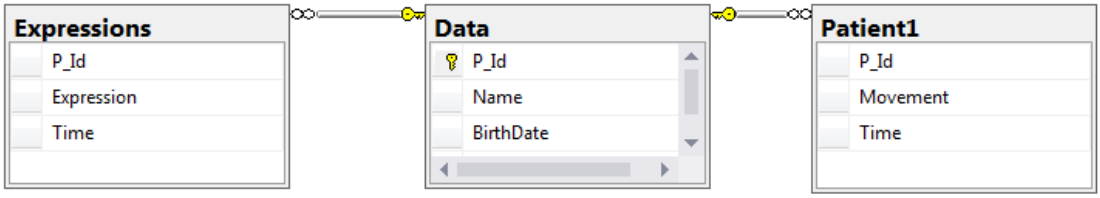


Figura 4.4:Arquitetura da Base de Dados

A aplicação permite também consultar informação da base de dados, ou seja pode-se ver a lista de pacientes registados com vários critérios. Podem ser visualizados pacientes com um certo nome, número de identificação ou num certo intervalo de idades. Por outro lado é possível consultar também os dados da tabela *Patient1*. A partir desta pode-se ver que movimentos efetuou um certo paciente num certo intervalo de tempo.

A identificação do paciente que se encontra focado no sensor pode ser feita de duas maneiras, ou pela identificação automática de faces ou pela seleção do paciente a partir da base de dados.

A identificação de faces é um processo que começa pela aprendizagem do algoritmo implementado com o rosto do paciente focado. De seguida esta face é guardada noutra base de dados onde está associada a um identificador, que por sua vez se refere ao identificador *P_Id* da tabela Data1.

A partir de agora tem-se a face do paciente registada. Sempre que este se encontrar em frente do sensor e o utilizador escolher a função de identificação automática do paciente, então este vai ser detetado. De seguida a aplicação envia uma mensagem de confirmação de paciente para que a identificação ganhe em robustez.

4.4. Especificações Técnicas do sistema

O sensor do *Kinect* tem cerca de 23 cm de comprimento horizontal e tem 7 componentes principais [31]:

- Camera RGB (*Red, Green, Blue)* que guarda os três canais numa resolução de 1280x960.
- Sensor de profundidade, que permite que o acessório tenha perceção do ambiente a sua volta em três dimensões.
- Microfone embutido que além de captar as vozes mais próximas, consegue diferenciar os ruídos externos. Dessa forma, barulhos ao fundo não atrapalham o andamento do *Kinect*. O microfone também é capaz de detetar várias pessoas diferentes numa sala (só não se sabe se a precisão é perfeita, já que é comum, por exemplo, irmãos com vozes parecidas).
- Tem um próprio processador e *software*.
- Sensor de inclinação que age automaticamente de forma a se enquadrar com o motivo em foco.
- Deteta 48 pontos de articulação do corpo, ou seja, possui uma precisão sem precedentes.
- Deteta 108 pontos na face.

Na figura abaixo pode-se observar o posicionamento dos vários componentes no sensor *Kinect*.

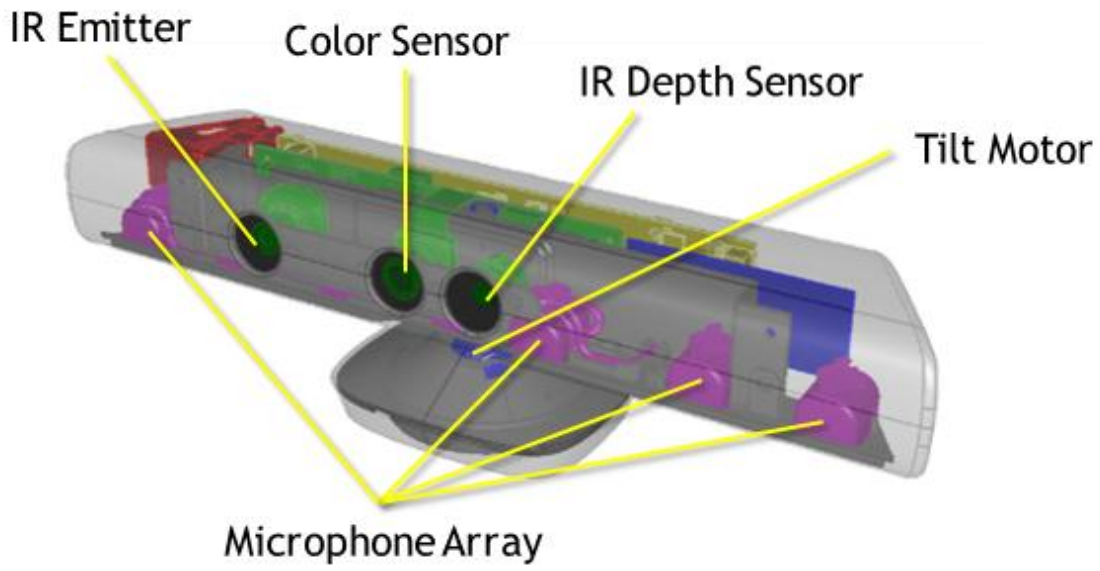


Figura 4.5: Especificações do *Kinect*

Através das potencialidades do sensor é possível identificar movimentos corporais do individuo produzindo um *input* de informação, sendo que existem inúmeros movimentos possíveis que têm que ser detetados e interpretados. Além disto pode-se também detetar vozes e expressões faciais que transmitem a emoção do indivíduo.

O *Kinect* para Windows *SDK* suporta até dois indivíduos que são monitorizados em simultâneo. O índice de um "jogador" é inserido nos 3 bits inferiores dos dados de profundidade para que se possa distinguir que pixéis de profundidade pertencem a cada jogador. Estes bits têm que se ter em conta aquando da verificação dos valores de profundidade [31].

Os dados de profundidade que se podem recolher do sensor são representados em 11 bits medidos a partir da posição da câmara como na Figura 4.6.

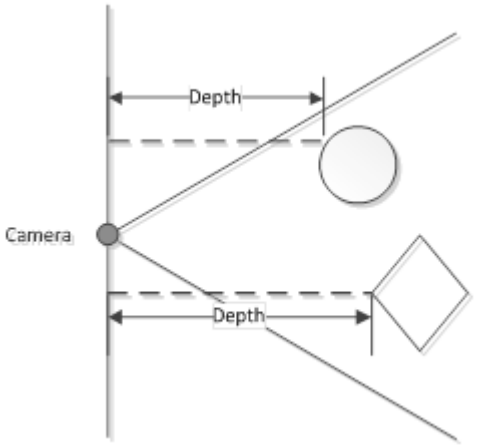


Figura 4.6: Funcionamento do sensor de profundidade [31]

Os pixéis de profundidade não alinham perfeitamente com os pixéis RGB por várias razões, mas serviço *Kinect* fornece uma maneira de mapear pixéis de profundidade em pixéis de cor (RGB), como se pode observar nas tabelas acima referidas [32].

Assim há a possibilidade de representar os esqueletos como um conjunto de articulações 3D no espaço. Para tirar partido total das potencialidades do sensor é necessário recorrer a todas as configurações possíveis. Desde definir o *frame rate* até à escolha do modo de operação passando pelo tipo de dados que se quer recolher e em que dimensões [31] [33].

4.5. *Kinect SDK*

A Microsoft fornece um *Kinect SDK* que suporta c++, c# e que permite o desenvolvimento de *software*. Existe uma biblioteca que fornece informação acerca do desenvolvimento de alguns programas como o *tracking* do esqueleto, da orientação das articulações, de caras e do reconhecimento de vozes através do microfone, mas este apenas suporta para já as línguas francês, japonês, espanhol, inglês, italiano e alemão.

O *Kinect* tem a capacidade de fazer o *tracking* de duas pessoas e a identificação de até seis.

A partir destas ferramentas é possível descobrir a postura da pessoa que se está a monitorizar, recolher informações sobre as suas movimentações e ações.

5. Deteção de gestos

Neste capítulo são abordados os algoritmos de deteção e reconhecimento de gestos e a forma de tratamento e armazenamento dos dados. Os algoritmos serão analisados, moldados, implementados e testados. De forma à obtenção de uma conclusão o mais precisa possível são analisados várias variações dos algoritmos propostos.

Um gesto tem a sua origem no movimento do corpo ou de uma parte do corpo e constitui uma parte importante da interação humana, indo desde expressões faciais a linguagem corporal e gestos com as mãos.

A deteção de gestos é muito importante também para a avaliação do estado e comportamento do paciente. Em concreto, o tipo de pacientes que vão ser observados desenvolve rituais compulsivos e maneirismos motores estereotipados e repetitivos [34].

Com este tipo de função é possível também criar uma biblioteca de linguagem gestual, a qual permite traduzir linguagem gestual para palavras de forma a facilitar a comunicação dos profissionais de saúde com os doentes, entre doentes ou mesmo como forma de ensinamento desta linguagem. A aprendizagem de novas formas de comunicação pode prevenir crianças mudas de se tornarem autistas ou mesmo o agravamento do estado de crianças autistas que possuem a capacidade da fala, mas não conseguem usá-la fora do ambiente mais conhecido [34].

Os algoritmos testados neste projeto implicam *Hidden Markov Models (5.2)* e *Dynamic Time Warping (5.4)*. Estes dois métodos foram selecionados, pois são os mais utilizados para reconhecer gestos humanos permitindo reconhecer inclusivamente gestos complexos [35]. Outra vantagem destes dois métodos utilizados reside na sua natureza de programação dinâmica, o que permite resolver problemas complexos através da divisão em problemas mais simples.

5.1.1. Recolha de dados

Neste capítulo inicia-se a interação com o sensor *Kinect* Figura 4.. A partir dos serviços fornecidos é necessário recolher os dados que interessam de forma ser possível a obtenção de informação medicamente relevante.

Os algoritmos de deteção de gestos implicam, normalmente as fases de recolha de [1]*features*, treino e teste. Em primeiro lugar, na recolha de *features* é necessário definir o tempo que normalmente se leva a concluir uma ação, desde o início que incluí a preparação, passando pelo núcleo do gesto e pela terminação. Aqui é necessário filtrar os pontos mais importantes na execução de um gesto, por exemplo se é melhor focar as atenções numa certa articulação, de forma a diminuir a quantidade de dados e a dimensão.

De seguida é necessário definir o nome do gesto e as várias maneiras que se pode reproduzir o gesto de forma a poder surgir alguma generalização. Na fase seguinte verifica-se qual a sensibilidade e especificidade do algoritmo.

Existem várias maneiras para recolher *features* dos movimentos. Por exemplo, através de *Motion History Image* (MHI) e através de recolhas de pontos das articulações 2D ou em 3D. O MHI consiste na recolha de uma imagem estática que evidencia a sequência de movimentos que eventualmente se traduz num gesto, a partir da diferenciação da intensidade dos pixéis sabe-se o início e o final do mesmo, como se pode observar na Figura 5.1 [36].

Figura 5.1: : Motion History Image [36]

Como se pode ver, existem algumas limitações na utilização de MHI, pois nalgumas posições, os gestos são irreconhecíveis dificultando a determinação da saída.

[1] *Feature*: pedaço de informação relevante para a conclusão de uma tarefa computacional (***Institute of Electrical and Electronics Engineers***) .

Por outro lado o sensor aqui utilizado possibilita, entre outras coisas, a recolha da posição das articulações de um individuo, tanto no espaço 2D como no 3D com uma taxa de até 30 *frames*[2] por segundo, independentemente da posição da pessoa em relação à câmara. Assim sendo este vai ser o método eleito. As articulações reconhecidas pelo *Kinect* estão explícitas na Figura 5.2.

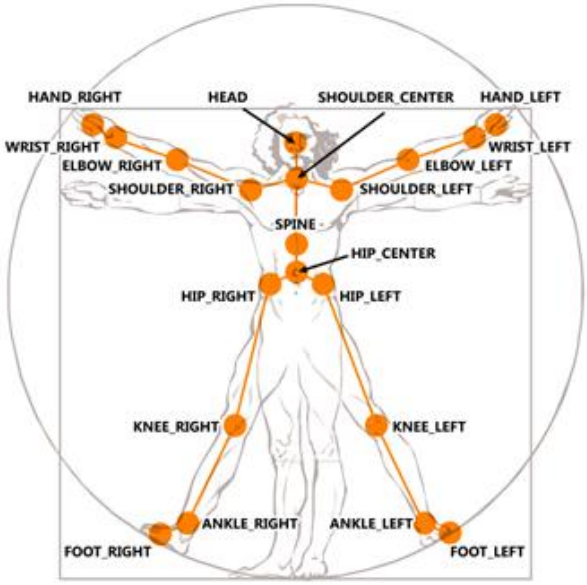


Figura 5.2: Articulações identificadas pelo *Kinect* [31]

O *Kinect* tem a capacidade de detetar múltiplas articulações, o que vai por um lado diminuir a fluidez de trabalho dos algoritmos utilizados e por outro lado dificultar a obtenção de bons resultados, pois a quantidade de informação a processar é demasiado grande. Nesse sentido é necessário guardar alguns pontos que sirvam de referência.

Como a maior parte dos movimentos que interessam detetar se efetuam com a parte superior do corpo e de forma a resolver os problemas acima, optou-se por recolher apenas informações dos pontos referentes às mãos, ombros e cotovelos.

A receção destes pontos obedece ao sistema de coordenadas do sistema descrito na Figura 5.3.

[2]Cada um dos quadros ou imagens fixas de um produto audiovisual (Jaques & Marie, Michael: "Dicionário teórico e crítico de cinema", ed. Papirus, 2001, pp. 136-137)

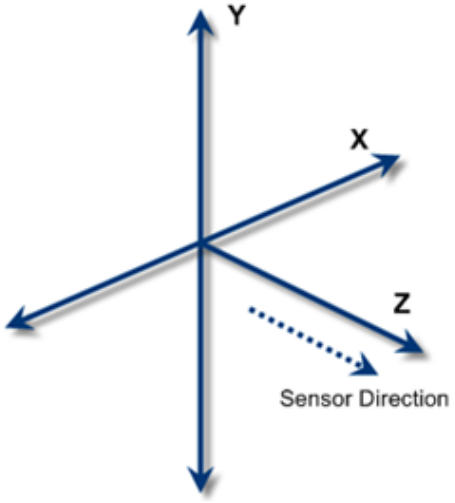


Figura 5.3: Sistemas de coordenadas do *Kinect* [31]

5.1.2. Tratamento de dados

O tratamento de dados em bruto recolhidos do sensor, inicia-se pela normalização e centralização em relação ao ponto médio entre as articulações referentes aos ombros. De seguida os pontos sofrem uma rotação de para que sejam detetados eficientemente, mesmo que não sejam realizados de frente para a câmara *Kinect*. Esta operação é eftetuada com recurso à expressão:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Esta permite efetuar uma rotação 2D em torno dum eixo Z.

Após esta primeira fase converte-se o tipo de dados *Skeleton*[3] provenientes do sensor para *Array*[4]. Finalmente os dados são enviados de volta para análise. Este processo está explícito na Figura 5.4.

[3] Tipo de dados produzidos internamente pelo *software* do sensor *Kinect* (msdn microsoft).
[4] Tipo de dados que consiste numa matriz (msdn microsoft).

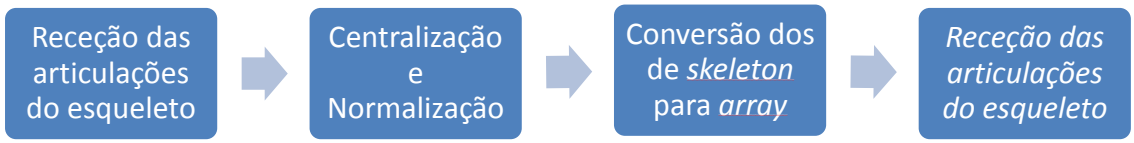


Figura 5.4: Tratamento dos dados

5.1.3. Armazenamento dos dados

Para o armazenamento dos dados depois de tratados e apostos para a aplicação dos algoritmos foi escolhido o formato XML, pois desta forma é possível que os dados sejam lidos e interpretados tanto pela aplicação como por humanos. Esta tecnologia permite também a identificação dos dados recebidos em cada uma das dimensões (X,Y,Z), em cada uma das articulações (*HandLeft*, *HandRight*, *WristLeft*, *WristRight*, *ElbowLeft*, *ElbowRight*) e em cada um dos gestos (levantar_dois_braços,…).

A mecânica implícita neste documento pode ser observada na Figura 5.5, onde se observa um excerto do documento XML.

```
<?xml version="1.0" encoding="utf-8"?>
<Gestures>
  <Gesture name="@levantar_dois_braços">
    <frame number="1">
      <joint name="HandLeft">
        <X>-0,710567575046418</X>
        <Y>-1,72207973418769</Y>
        <Z>-0,437885085983815</Z>
      </joint>
      <joint name="WristLeft">
        <X>-0,741539342126347</X>
        <Y>-1,56774950204795</Y>
        <Z>-0,38305947620941</Z>
      </joint>
      <joint name="ElbowLeft">
        <X>-0,748731924813793</X>
        <Y>-0,919355825669336</Y>
        <Z>-0,113894658796318</Z>
      </joint>
      <joint name="HandRight">
        <X>0,80577588642172</X>
        <Y>-0,840281911331331</Y>
        <Z>-0,188908173186808</Z>
      </joint>
      <joint name="WristRight">
        <X>0,860321129448492</X>
        <Y>-1,48684538205833</Y>
        <Z>-0,456864968689616</Z>
```

Figura 5.5: Excerto XML

5.2. Algoritmo *Hidden Markov Model*

O *Hidden Markov Model* (HMM) é um algoritmo de análise de vídeo que pode ser utilizado no reconhecimento de gestos. Um HMM consiste em N estados e uma matriz de transição. Cada estado tem associado também uma função de distribuição de probabilidades de output, o que indica a probabilidade de um estado N gerar a observação. [37].

O modelo HMM é descrito por:

$$H = (P_{ij}, e_i(a), w_i)\ , \begin{cases} i\ \ (1 \le i \le N) \text{: estado em que o modelo se encontra} \\ j\ \ (1 \le j \le N) : \text{estado para o qual o sistema transita} \\ a\ (1 \le a \le M), : \text{observações possíveis em cada estado} \\ e \text{: matriz de probabilidades de emissão de observações} \\ P_{ij} : \text{probabilidade de transição do estado } i \text{ para o } j. \end{cases}$$

As probabilidades de mudança de estado são:

$$p_{ij} = p(q_{t+1} = j | q_t = i), \begin{cases} q_t: estado\ escondido\ no\ tempo\ t \\ i\ \ (1 \leq i \leq N): estado\ em\ que\ o\ modelo\ se\ encontra \\ j\ \ (1 \leq j \leq N): estado\ para\ o\ qual\ o\ sistema\ transita \end{cases}$$

A probabilidade de emissão para a observação a no estado i é:

$e_i(a) = p(O_t = a | q_t = i)$, onde O_t é a observação no tempo t e a probabilidade do estado inicial é $w_i = p(q_1 = i)$.

Dada a sequência de observações $O = O_1, O_2 \ldots, O_t$, e o modelo $H = (P_{ij}, e_i(a), w_i)$, existem três problemas básicos relacionados com os HMM.

- Avaliação de $P(O|Q)$, que poderá ser resolvido com o algoritmos *forward-backward* (explicado mais a frente *5.2.2*).
- Dada a sequência de observações $O = O_1, O_2 \ldots, O_t$, e o modelo $H = (P_{ij}, e_i(a), w_i)$, o objetivo é encontrar a probabilidade máxima da sequência de estados $Q = Q_1, Q_2 \ldots, Q_t$.
- Treinar o modelo HMM para que se possa prever a sequência de estados e emissões.

Os problemas são resolvidos através da aplicação dos algoritmos Baum-Welch e *Viterbi* explicados mais à frente.

Na Figura 5.6 estão explícitas as dependências no modelo HMM. Sabe-se então que a saída x(t) depende apenas do estado x(t), este por sua vez depende apenas do estado X(t-1). Ou seja, os estados anteriores x(t-2), x(t-3)… são ignorados, esta é a propriedade de primeira ordem de *Markov*.

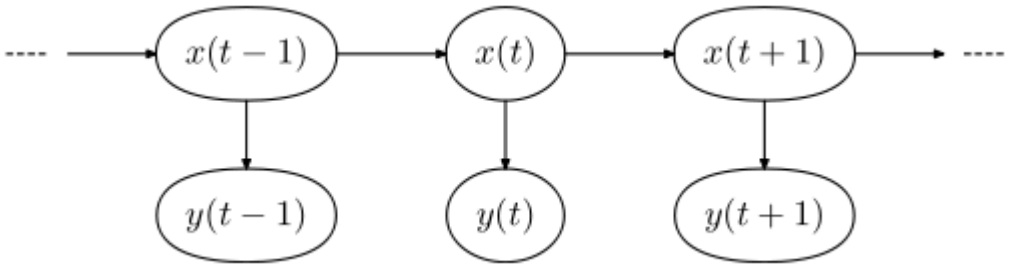


Figura 5.6: Dependências no modelo HMM [38] [39]

Na equação seguinte 5.1, observa-se que de fato este algoritmo funciona através de probabilidades, existe sempre um probabilidade de passagem de um estado y(t-1) para o seguinte y(t) e uma probabilidade de relação entre o estado e a respetiva saída [40] [41].

$$p(x,y) = \prod_{t=1}^{T} \underbrace{p(y_t|y_{t-1})}_{A} \underbrace{p(x_t|y_t)}_{B}$$

Equação 5.1: Dependências do algoritmo *Hidden Markov Models* [40]

Assim, o HMM representa o modelo típico para uma sequência estocástica de um número finito de estados. Isto implica que a saída do modelo no instante *i* depende apenas do estado no tempo *i*. Em cada estado, um output é emitido com uma certa probabilidade e um estado transita para outro com a mesma probabilidade. Com um número discreto de estados e outputs, este modelo é muitas vezes conhecido como HMM discreto [42]. No entanto as observações podem ser discretas, mas também contínuas, tipicamente através de uma distribuição gaussiana. As probabilidades de transição controlam a maneira como é escolhido o estado t tendo em conta o $t-1$. Além das probabilidades de transição existem probabilidades de emissão que dependem também da natureza da observação [39]. Como é tido em conta sempre o estado anterior, este algoritmo é usado em termos de reconhecimento de gestos, pois quando uma pessoa gesticula existe um encadeamento de palavras e portanto é tida em conta a dinâmica de passagem entre estados.

Enquanto que este algoritmo pode ser suficiente para determinados processos, normalmente resulta em baixas taxas de reconhecimento de gestos em processos que não respeitem a propriedade de primeira ordem de Markov. Como por exemplo no reconhecimento de vozes, estes problemas podem ser reduzidos ao estender os parâmetros dos vetores com as derivadas no tempo dos parâmetros originais. Também é possível derivar modelos de ordem superior, mas estes não partilham da eficiência computacional dos modelos de primeira ordem. Outra possível desvantagem dos modelos HMM é a pressuposição de que as funções de distribuição das observações podem ser modeladas como uma mistura de gaussianas, o que adiciona incerteza na determinação de gestos por exemplo [39] [42] .

Na prática o processo consiste na produção, em primeiro lugar, de modelos para cada gesto. De forma a preparar os dados é reduzida a dimensionalidade do vetor de características através de uma técnica de clustering. Cada frame, representado por 18 valores, contém 6

articulações. Após esta fase, é feito um treino através da elaboração de um HMM para cada gesto, possibilitando a adição de novos gestos desde que se forneçam amostras de forma a treinar estes [43].

5.2.1. Algoritmo *Viterbi*

Nesta fase procede-se à explicação do algoritmo de treino *Viterbi*. Dada a sequência de observações $O = O_1, O_2 \dots, O_t$, e o modelo $H = (P_{ij}, e_i(a), w_i)$, o objetivo é encontrar a probabilidade máxima da sequência de estados $Q = Q_1, Q_2 \dots, Q_t$. Isto pode ser feito através do algoritmo *Viterbi* seguindo o seguindo processo em [44] [45].

Sabendo que:

$$v_i(t) = max_{q_1 q_2 \dots q_t} P(q_1 q_2 \dots q_{t-1}, q_t = i, O_1 O_2 \dots O_t | H)$$

e w_i a probabilidade inicial dos estados i no tempo $t = 1$. Observa-se então que v_j pode ser calculado recursivamente usando:

$$v_j(t) = max_{1 \le i \le N}[v_i(t-1)p_{ij}]e_j(O_t)$$

em conjunto com a inicialização:

$$v_i(1) = w_i e_i(O_1), \quad 1 \le i \le N$$

E a terminação

$$P^* = max_{1 \le i \le N} |v_i(T)|$$

Pode-se então calcular a probabilidade máxima no último ponto e a partir daqui inverter a ordem de cálculo e encontrar o caminho de maior probabilidade.

É de notar que o caminho mais provável não é calculado através da avaliação ponto a ponto dos estados mais prováveis, mas sim através da avaliação do caminho em geral. Para que isto se proceda dessa maneira é necessário utilizar o algoritmo *Baum-Welch*.

5.2.1.1. Exemplo

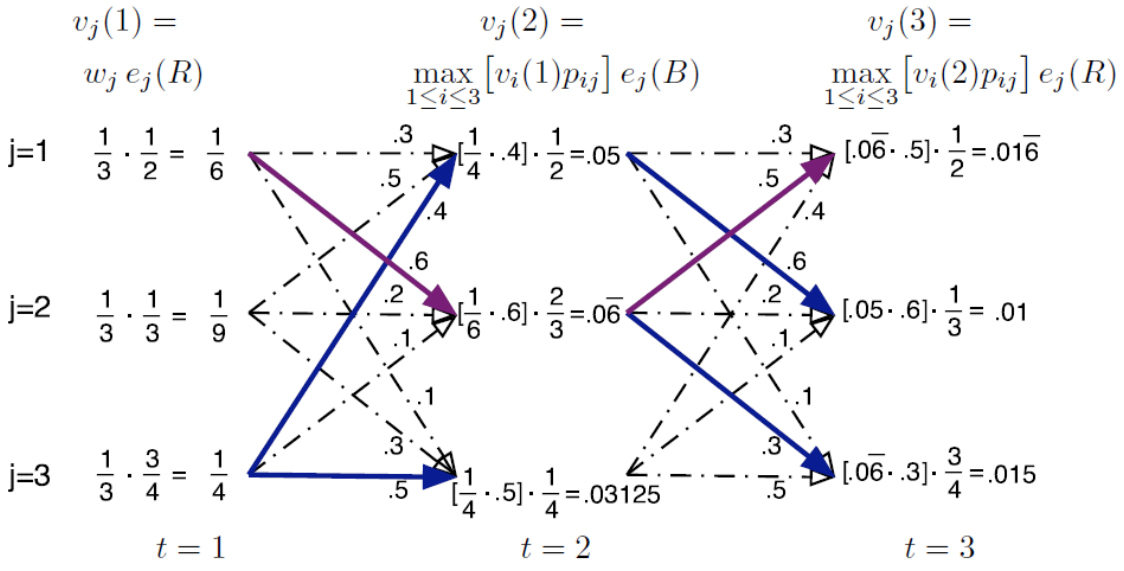
De forma a ilustrar este algoritmo pode-se dar um exemplo de um HMM com três estados e com as saídas R ou B emitidas em cada um (por exemplo existem três urnas com bolas vermelhas ou azuis). Este exemplo está disponível em [44]. As probabilidades de emissão são expressas na tabela abaixo.

Tabela 5.1:probabilidades de emissão

	R	B
e_1	$\frac{1}{2}$	$\frac{1}{2}$
e_2	$\frac{1}{3}$	$\frac{2}{3}$
e_3	$\frac{3}{4}$	$\frac{1}{4}$

Considerando também a matriz de transição $p_{ij} = \begin{pmatrix} 0.3 & 0.6 & 0.1 \\ 0.5 & 0.2 & 0.3 \\ 0.4 & 0.1 & 0.5 \end{pmatrix}$ e a probabilidade inicial dos estados é $w_i = \frac{1}{3}$, ou seja cada estado tem a mesma probabilidade de ocorrer.

Tendo em conta também a sequência RBR, de seguida pode-se encontrar a mais provável sequência de estados de forma a explicar a sequência de observações utilizando o algoritmo *Viterbi*.



No primeiro passo são utilizadas as probabilidades iniciais dos estados, ou seja em $t = 1$, de forma a calcular $v_j(t = 1) = w_j e_j(R)$ para cada estado $j = 1, j = 2, j = 3$.

$$\begin{cases} v_j(t=1) = w_j e_j = \frac{1}{6}, & j = 1 \\ v_j(t=1) = w_j e_j(R) = \frac{1}{9}, & j = 2 \\ v_j(t=1) = w_j e_j(R) = \frac{1}{4}, & j = 3 \end{cases}$$

De seguida é necessário utilizar a matriz de transição e as probabilidades já calculadas de forma a proceder ao calculo de $v_j(t=2)$. Por exemplo, para $j=1$, sabendo que: $v_i(1) \cdot p_{ij}$ e i corresponde aos estados iniciais, tem-se:

$$\begin{cases} v_i(1) \cdot p_{i1} = \frac{1}{6} \cdot 0.3 = 0.05, & \mathrm{i} = 1 \\ v_i(1) \cdot p_{i1} = \frac{1}{9} \cdot 0.5 = 0.0\overline{5}, & \mathrm{i} = 2 \\ v_i(1) \cdot p_{i1} = \frac{1}{4} \cdot 0.4 = 0.1, & \mathrm{i} = 3 \end{cases}$$

Desta operação vai prevalecer o maior resultado, ou seja o terceiro (obedecendo à escolha da seta azul). Assim conclui-se que $v_1(t=2) = \left(\frac{1}{4} \cdot 0.4\right) \cdot \frac{1}{2} = 0.05$.

Os restantes estados em $t=2$ processam-se da mesma maneira, sendo que para determinar $v_j(t=2)$ $j=2$, (considera-se as três quantidades $v_i(1) \cdot p_{i2}$ para $i=(1,2,3)$, sendo i os estados iniciais tem-se:

$$\begin{cases} v_i(1) \cdot p_{i1} = \frac{1}{6} \cdot 0.6 = 0.1, & \mathrm{i} = 1 \\ v_i(1) \cdot p_{i1} = \frac{1}{9} \cdot 0.2 = 0.0\overline{2}, & \mathrm{i} = 2 \\ v_i(1) \cdot p_{i1} = \frac{1}{4} \cdot 0.1 = 0.025, & \mathrm{i} = 3 \end{cases}$$

O maior valor é o primeiro (obedecendo à seta roxa) o que vai levar a que $v_2(t=2) = \left(\frac{1}{6} \cdot 0.6\right) \cdot \frac{2}{3} = 0.0\overline{6}$.

Em $v_3(t=2)$ o processo vai ser similar, calculando $v_i(t=1) \cdot p_{i3}$ para $i=(1,2,3)$,

$$\begin{cases} v_i(1) \cdot p_{i1} = \frac{1}{6} \cdot 0.1 = 0.01\overline{6}, & \text{i} = 1 \\ v_i(1) \cdot p_{i1} = \frac{1}{9} \cdot 0.3 = 0.0\overline{3}, & \text{i} = 2 \\ v_i(1) \cdot p_{i1} = \frac{1}{4} \cdot 0.5 = 0.125, & \text{i} = 3 \end{cases}$$

O maior valor é o terceiro o que vai levar a que $v_3(t=2) = \left(\frac{1}{4} \cdot 0.5\right) \cdot \frac{1}{4} = 0.0325$.

Finalmente, para os estados em $t = 3$ encontra-se uma solução similar ao que acontece nos estados em $t = 2$.

Para decidir qual o melhor caminho recorre-se à equação $P^* = max_{1 \le i \le N} |v_i(T)|$ correspondente à condição final. Conclui-se então que $P^* = 0.01\overline{6}$, correspondente ao primeiro estado de $t = 3$. A partir daqui conclui-se que o caminho mais provável de estados é $j = 1, j = 2, j = 1$ correspondente ao caminho assinalado pela linha roxa.

5.2.2. Algoritmo Baum-Welch (Forward-Backward)

O algoritmo *Baum-Welch* recorre à computação de dois conjuntos de probabilidades: forward e backward, retiradas de [46] [47].

Em primeiro lugar é calculada a probabilidade de existir um determinado estado tendo em conta as primeiras observações conhecidas até k. Este cálculo é dado pela expressão $P(X_k|O_{1:k})$, na qual X_k representa o estado escondido, $O_{1:k}$ o conjunto de saídas ou emissões e $k \in \{1,..,t\}$.

Na segunda fase é efetuado o processo contrário (*backward*), ou seja é calculada a probabilidade de observar o resto das saídas que ainda não são conhecidas dado um ponto de início k e o estado respetivo. A equação $P(O_{k+1:t}|X_k)$ representa este passo.

Após estas duas fases é possível descobrir a sequência de estados em qualquer ponto.

No último passo prende-se suavizar os dados através da regra de Bayes.

5.2.2.1. Sentido Forward

No sentido *Forward*, a fórmula que permite calcular a probabilidade de mudança de um estado (i) para outro (j) observando um dado evento é a seguinte:

$$f_{0:t} = f_{0:t-1} T O_t \quad \text{e} \quad f_{0:t}(i) = P(O_1, O_2, \dots, O_t, X_t = x_i|\pi)$$

Aplicando o fator de escala para que a soma das probabilidades do vetor seja 1, tem-se:

$$\hat{f}_{0:t} = c_t^{-1}\hat{f}_{0:t-1}TO_t$$

Na qual O se refere à observação ou saída, T é a matriz de probabilidades de transição, c_t^{-1} é um fator de escala que faz com que a soma dos vetores resultantes seja 1, $f_{0:t-1}$ é a probabilidade de mudança de estado no instante anterior e $f_{0:t}$ a probabilidade de mudança de estado no atual. π representa o estado inicial e x_i um estado específico.

A condição terminal é:

$$P(O_1, O_2, \dots, O_t|\pi) = \prod_{s=1}^{t} c_s$$

Isto permite interpretar $\hat{f}_{0:t}$ como sendo:

$$\hat{f}_{0:t}(i) = \frac{f_{0:t}(i)}{\prod_{s=1}^{t} c_s} = \frac{P(O_1, O_2, \dots, O_t, X_t = x_i|\pi)}{P(O_1, O_2, \dots, O_t|\pi)} = P(X_t = x_i|O_1, O_2, \dots, O_t, \pi)$$

Pode-se concluir que os fatores de escala fornecem a probabilidade total de se observar uma sequência até ao tempo t e que o vetor probabilidade em escala nos fornece a probabilidade de se estar num determinado estado no tempo t.

5.2.2.1. Sentido Backward

No sentido *Backward*, tem-se a seguinte equação:

$$b_{t:T}(i) = P(O_{t+1}, O_{t+2}, \dots, O_T|X_t = x_i)$$

Traduzindo, calcula-se a probabilidade de se observar os eventos futuros tendo em conta um determinado estado. Aqui tem-se a certeza de que se atinge o último evento. Neste caso usa-se um vetor coluna de probabilidade.

$$\hat{b}_{t-1:T} = c_t^{-1}TO_t\hat{b}_{t:T}$$

Aqui existem duas opções: ou usar o mesmo fator de escala usado no sentido *forward* ou usar um novo fator de escala que permita que a soma do vetor resultante seja 1.

Assim, encontra-se:

$$\hat{b}_{t:T}(i) = \frac{b_{t:T}(i)}{\prod_{s=t+1}^{t} c_s}$$

A partir desta fase pode-se calcular a probabilidade total de estar num determinado estado ao efetuar a operação:

$$\gamma_t(i) = P(X_t = x_i|O_1, O_2, \dots, O_t, \pi) = \frac{P(O_1, O_2, \dots, O_t, X_t = x_i|\pi)}{P(O_1, O_2, \dots, O_t|\pi)} = \frac{f_{0:t}(i) \cdot b_{t:T}(i)}{\prod_{s=1}^{t} c_s} = \hat{f}_{0:t}(i) \cdot \hat{b}_{t:T}(i)$$

A operação $\hat{f}_{0:t}(i) \cdot \hat{b}_{t:T}(i)$ permite calcular a probabilidade total de passar por um estado $X_t = x_i$ tendo em conta os eventos observados e o estado inicial. Tem-se então em conta os eventos até ao tempo t (através das probabilidades Forward) e os eventos futuros ocorridos até T (através das probabilidades Backward).

Através da operação $\frac{f_{0:t}(i) \cdot b_{t:T}(i)}{\prod_{s=1}^{t} c_s}$ obtêm-se os valores suavizados do estado mais provável. No entanto sabe-se que a sequência de estados em cada ponto não se traduz necessariamente na sequência de estados mais provável. Isto acontece devido ao fato de se calcular a sequência de estados individualmente em cada ponto. Não se tem em conta a sequência total de estados. Esta situação é apenas coberta pelo algoritmo *Viterbi*.

5.2.2.2. Exemplo

De forma a melhorar a compreensão deste processo, ilustra-se com um exemplo relacionado com o tempo. Este exemplo está disponível em [46]. Tenta-se inferir sobre o estado do tempo através da observação de um homem que leva um guarda-chuva ou não. Assim existem dois estados possíveis: ou chove (estado 1) ou não chove (estado 2). De seguida é necessário ter em conta alguns preconceitos como o tempo tem uma probabilidade de 70% de continuar igual ao dia anterior e 30% de probabilidade de mudar. Logo a matriz de probabilidades de transição é $T = \begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix}$.

Pode-se também dizer que existem dois eventos de saída: há guarda-chuva (evento 1) ou não há guarda-chuva (evento 2). A probabilidade de cada um destes ocorrer está expressa na matriz de probabilidade condicional $B = \begin{pmatrix} 0.9 & 0.1 \\ 0.2 & 0.8 \end{pmatrix}$. Daqui é possível concluir que existe uma probabilidade de 90% de haver guarda-chuva e estar a chover; uma probabilidade de 80% de não haver guarda-chuva e não chover; uma probabilidade de 10% de chover e não haver guarda-chuva; uma probabilidade de 20% de não chover e haver guarda-chuva.

Os eventos observados no intervalo $1 \leq t \leq 5$ respeitam a sequência: há guarda-chuva, há guarda-chuva, não há guarda-chuva, há guarda-chuva, há guarda-chuva. Ou seja observam-se

as seguintes matrizes de saída: $O_1 = \begin{pmatrix} 0.9 & 0.0 \\ 0.0 & 0.2 \end{pmatrix}$, $O_2 = \begin{pmatrix} 0.9 & 0.0 \\ 0.0 & 0.2 \end{pmatrix}$, $O_3 = \begin{pmatrix} 0.1 & 0.0 \\ 0.0 & 0.8 \end{pmatrix}$, $O_4 = \begin{pmatrix} 0.9 & 0.0 \\ 0.0 & 0.2 \end{pmatrix}$, $O_5 = \begin{pmatrix} 0.9 & 0.0 \\ 0.0 & 0.2 \end{pmatrix}$. Daqui conclui-se que não se sabe o que acontece antes das observações, ou seja em $t = 0$, logo pode-se assumir que existe uma probabilidade de 50% de haver guarda-chuva e 50% de não haver guarda-chuva.

Assim conclui-se que $f_{0:0} = (0.5 \quad 0.5)$. Não se pode aplicar diretamente a equação $\hat{f}_{0:t} = c_t^{-1} \hat{f}_{0:t-1} T O_t$, pois $f_{0:0}$ é um vetor linha, logo aplica-se $(\hat{f}_{0:t})^T = c_t^{-1} O_t T^T (\hat{f}_{0:t-1})^T$:

$$(\hat{f}_{0:1})^T = c_1^{-1} \begin{pmatrix} 0.9 & 0.0 \\ 0.0 & 0.2 \end{pmatrix} \begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix} \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} = c_1^{-1} \begin{pmatrix} 0.45 \\ 0.1 \end{pmatrix} = \begin{pmatrix} 0.8182 \\ 0.1818 \end{pmatrix}$$

$$(\hat{f}_{0:2})^T = c_2^{-1} \begin{pmatrix} 0.9 & 0.0 \\ 0.0 & 0.2 \end{pmatrix} \begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix} \begin{pmatrix} 0.8182 \\ 0.1818 \end{pmatrix} = c_2^{-1} \begin{pmatrix} 0.5645 \\ 0.0745 \end{pmatrix} = \begin{pmatrix} 0.8834 \\ 0.1166 \end{pmatrix}$$

$$(\hat{f}_{0:3})^T = c_3^{-1} \begin{pmatrix} 0.9 & 0.0 \\ 0.0 & 0.2 \end{pmatrix} \begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix} \begin{pmatrix} 0.8834 \\ 0.1166 \end{pmatrix} = c_3^{-1} \begin{pmatrix} 0.0653 \\ 0.2772 \end{pmatrix} = \begin{pmatrix} 0.1907 \\ 0.8093 \end{pmatrix}$$

$$(\hat{f}_{0:4})^T = c_4^{-1} \begin{pmatrix} 0.9 & 0.0 \\ 0.0 & 0.2 \end{pmatrix} \begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix} \begin{pmatrix} 0.1907 \\ 0.8093 \end{pmatrix} = c_4^{-1} \begin{pmatrix} 0.3386 \\ 0.1247 \end{pmatrix} = \begin{pmatrix} 0.7308 \\ 0.2692 \end{pmatrix}$$

$$(\hat{f}_{0:5})^T = c_5^{-1} \begin{pmatrix} 0.9 & 0.0 \\ 0.0 & 0.2 \end{pmatrix} \begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix} \begin{pmatrix} 0.7308 \\ 0.2692 \end{pmatrix} = c_5^{-1} \begin{pmatrix} 0.5331 \\ 0.0815 \end{pmatrix} = \begin{pmatrix} 0.8673 \\ 0.1327 \end{pmatrix}$$

Para as probabilidades *Backward* começa-se com $b_{5:5} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

$$\hat{b}_{4:5} = \alpha \begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix} \begin{pmatrix} 0.9 & 0.0 \\ 0.0 & 0.2 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \end{pmatrix} = \alpha \begin{pmatrix} 0.6900 \\ 0.4100 \end{pmatrix} = \begin{pmatrix} 0.6273 \\ 0.3727 \end{pmatrix}$$

$$\hat{b}_{3:5} = \alpha \begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix} \begin{pmatrix} 0.9 & 0.0 \\ 0.0 & 0.2 \end{pmatrix} \begin{pmatrix} 0.6273 \\ 0.3727 \end{pmatrix} = \alpha \begin{pmatrix} 0.4175 \\ 0.2215 \end{pmatrix} = \begin{pmatrix} 0.6533 \\ 0.3467 \end{pmatrix}$$

$$\hat{b}_{2:5} = \alpha \begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix} \begin{pmatrix} 0.1 & 0.0 \\ 0.0 & 0.8 \end{pmatrix} \begin{pmatrix} 0.6533 \\ 0.3467 \end{pmatrix} = \alpha \begin{pmatrix} 0.1289 \\ 0.2138 \end{pmatrix} = \begin{pmatrix} 0.3763 \\ 0.6237 \end{pmatrix}$$

$$\hat{b}_{1:5} = \alpha \begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix} \begin{pmatrix} 0.9 & 0.0 \\ 0.0 & 0.2 \end{pmatrix} \begin{pmatrix} 0.3763 \\ 0.6237 \end{pmatrix} = \alpha \begin{pmatrix} 0.2745 \\ 0.1889 \end{pmatrix} = \begin{pmatrix} 0.5923 \\ 0.4077 \end{pmatrix}$$

$$\hat{b}_{0:5} = \alpha \begin{pmatrix} 0.7 & 0.3 \\ 0.3 & 0.7 \end{pmatrix} \begin{pmatrix} 0.9 & 0.0 \\ 0.0 & 0.2 \end{pmatrix} \begin{pmatrix} 0.5923 \\ 0.4077 \end{pmatrix} = \alpha \begin{pmatrix} 0.3976 \\ 0.2170 \end{pmatrix} = \begin{pmatrix} 0.6469 \\ 0.3531 \end{pmatrix}$$

Os vetores encontrados acima representam a probabilidade de cada estado dado os eventos futuros. No entanto como se pode observar, foram utilizados os fatores α em vez dos c_t. Isto implica que na última fase vai-se submeter mais um fator de escala final.

A fase final pode ser observada de seguida:

$$(\gamma_0)^T = \alpha \begin{pmatrix} 0.5000 \\ 0.5000 \end{pmatrix} \times \begin{pmatrix} 0.6469 \\ 0.3531 \end{pmatrix} = \alpha \begin{pmatrix} 0.3235 \\ 0.1765 \end{pmatrix} = \begin{pmatrix} 0.6469 \\ 0.3135 \end{pmatrix}$$

$$(\gamma_1)^T = \alpha \begin{pmatrix} 0.8182 \\ 0.1818 \end{pmatrix} \times \begin{pmatrix} 0.5923 \\ 0.4077 \end{pmatrix} = \alpha \begin{pmatrix} 0.4846 \\ 0.0741 \end{pmatrix} = \begin{pmatrix} 0.8673 \\ 0.1327 \end{pmatrix}$$

$$(\gamma_2)^T = \alpha \begin{pmatrix} 0.8834 \\ 0.1166 \end{pmatrix} \times \begin{pmatrix} 0.3763 \\ 0.6237 \end{pmatrix} = \alpha \begin{pmatrix} 0.3324 \\ 0.0728 \end{pmatrix} = \begin{pmatrix} 0.8204 \\ 0.1796 \end{pmatrix}$$

$$(\gamma_3)^T = \alpha \begin{pmatrix} 0.1907 \\ 0.8093 \end{pmatrix} \times \begin{pmatrix} 0.6533 \\ 0.3467 \end{pmatrix} = \alpha \begin{pmatrix} 0.1246 \\ 0.2806 \end{pmatrix} = \begin{pmatrix} 0.3075 \\ 0.6925 \end{pmatrix}$$

$$(\gamma_4)^T = \alpha \begin{pmatrix} 0.7308 \\ 0.2692 \end{pmatrix} \times \begin{pmatrix} 0.6273 \\ 0.3727 \end{pmatrix} = \alpha \begin{pmatrix} 0.4584 \\ 0.1003 \end{pmatrix} = \begin{pmatrix} 0.8204 \\ 0.1796 \end{pmatrix}$$

$$(\gamma_5)^T = \alpha \begin{pmatrix} 0.8673 \\ 0.1327 \end{pmatrix} \times \begin{pmatrix} 1.0000 \\ 1.0000 \end{pmatrix} = \alpha \begin{pmatrix} 0.8673 \\ 0.1327 \end{pmatrix} = \begin{pmatrix} 0.8673 \\ 0.1327 \end{pmatrix}$$

Após estes cálculos é possível inferir acerca da probabilidade de chover ou não chover. A probabilidade de chover foi sempre superior à de não chover exceto no terceiro dia. Agora também é possível quantificar a probabilidade de ocorrência de um estado em tempos diferentes. O valor calculado com mais importância é o γ_5 pois quantifica a probabilidade observada no final da sequência. Este pode ser utilizado para prever os estados do tempo no futuro assim como a existência de um guarda-chuva.

5.3. Implementação do algoritmo HMM

Nesta parte do capítulo é descrita a implementação do algoritmo HMM em *c#* através do *Visual Studio 2012*.

Além da recolha dos dados através do *SDK Kinect* da Microsoft, aqui foi utilizado a framework Accord.NET.

5.3.1. Framework Accord.NET

É uma *framework* para computação científica em .NET. Esta foi construída a partir da famosa AForge.NET que é utilizada em aplicações de processamento de imagem [40]. As bibliotecas disponibilizadas permitem fazer processamento estatístico de dados e reconhecimento de padrões incluindo de visão e de audição computacional.

Aqui são disponibilizadas, todas as ferramentas que permitem implementar o algoritmo. Através da classe *HiddenMarkovClassifier* é possível inicializar os modelos de *Markov* com um certo número de estados, outputs e tipo de dados que se poderão utilizar.

Através da classe *HiddenMarkovClassifierLearning* define-se qual o algoritmo de aprendizagem mais conveniente para que se possa efetuar o treino a partir dos dados iniciais. Aqui existem algoritmos supervisionados e não supervisionados. Nesta fase, um algoritmo supervisionado implica que a sequência de estados escondidos esteja disponível durante o treino e com a utilização de um algoritmo não supervisionado não vai haver esta disponibilidade. Outra diferenciação que poderá existir em relação à escolha do algoritmo prende-se com a quantidade de dados disponível para o treino. Se houver muitos dados de treino é mais produtivo a utilização de um algoritmo supervisionado. Neste caso, devido à quantidade de dados a ser analisados, serão utilizados algoritmos não supervisionados.

5.3.1. Obtenção de resultados

Os testes foram efetuados com gestos gravados por 19 pessoas diferentes, que efetuaram uma vez cada um dos gestos indicados. Quando não se está a reproduzir nenhum gesto ou não é reconhecido, este resultado é guardado na variável *Fazer Nada*. Neste caso a sensibilidade corresponde à capacidade de um gesto ser reconhecido corretamente e a especificidade corresponde à capacidade de um não gesto ser reconhecido corretamente.

No processo é treinado o algoritmo com todos os gestos menos um, este gesto deixado de fora do treino é o que vai ser testado contra o resto.

5.3.2. Resultados obtidos

De seguida é necessário testar se o algoritmo consegue realmente distinguir gestos efetuados por várias pessoas.

1.1.1.1. Treino Baum-Welch

Com o treino *Baum-Welch* associado aos *Hidden Markov Models* foram produzidos os resultados descritos neste capítulo.

Nas tabelas seguintes pode-se observar o desempenho dos algoritmos. Nesta tabela estão em correspondência os gestos reconhecidos pelo algoritmo e o gesto executado pelo individuo em frente da câmara.

Tabela 5.2: HMM Baum-Welch com 5 estados escondidos sem *threshold*

Execução / Reconhecido	Levantar dois braços	Levantar braço direito	Levantar braço esquerdo	Bater palmas	Sentar	Levantar	Deslize Direita	Deslize esquerda
Levantar dois braços	81,82%	9,09%	9,09%	0,00%	25,00%	0,00%	0,00%	0,00%
Levantar braço direito	0,00%	63,64%	0,00%	0,00%	0,00%	16,67%	22,22%	0,00%
Levantar braço esquerdo	9,00%	0,00%	72,73%	0,00%	0,00%	0,00%	0,00%	40,00%
Bater palmas	0,00%	0,00%	0,00%	100,00%	0,00%	0,00%	0,00%	0,00%
Sentar	0,00%	0,00%	0,00%	0,00%	37,50%	50,00%	0,00%	0,00%
Levantar	0,00%	0,00%	0,00%	0,00%	37,50%	33,33%	0,00%	0,00%
Deslize Direita	0,00%	27,27%	0,00%	0,00%	0,00%	0,00%	77,78%	0,00%
Deslize Esquerda	0,00%	0,00%	9,09%	0,00%	0,00%	0,00%	0,00%	60,00%

Tabela 5.3: HMM Baum-Welch com 10 estados escondidos sem *threshold*

Execução / Reconhecido	Levantar dois braços	Levantar braço direito	Levantar braço esquerdo	Bater palmas	Sentar	Levantar	Deslize Direita	Deslize esquerda
Levantar dois braços	100,00%	9,09%	18,18%	0,00%	25,00%	0,00%	0,00%	0,00%
Levantar braço direito	0,00%	81,82%	0,00%	0,00%	0,00%	0,00%	22,22%	0,00%
Levantar braço esquerdo	0,00%	0,00%	72,73%	0,00%	0,00%	0,00%	0,00%	40,00%
Bater palmas	0,00%	0,00%	0,00%	100,00%	0,00%	0,00%	0,00%	0,00%
Sentar	0,00%	0,00%	0,00%	0,00%	50,00%	50,00%	0,00%	0,00%
Levantar	0,00%	0,00%	0,00%	0,00%	12,50%	50,00%	0,00%	0,00%
Deslize Direita	0,00%	9,09%	0,00%	0,00%	0,00%	0,00%	77,78%	0,00%
Deslize Esquerda	0,00%	0,00%	9,09%	0,00%	0,00%	0,00%	0,00%	60,00%

A partir dos primeiros resultados obtidos pode-se concluir que a utilização de um *threshold* é muito importante, pois sem este é impossível analisar se uma pessoa não está a efetuar

nenhum gesto em frente da câmara. Mesmo assim, no Tabela 5.3 é observável um melhor comportamento quando se aumentam o número de estados escondidos, ou seja o número de estados que existem disponíveis, dos quais é produzida uma observação.

Tabela 5.4 : HMM Baum-Welch com 5 estados escondidos com *threshold* de sensibilidade 5

Execução / Reconhecido	Levantar dois braços	Levantar braço direito	Levantar braço esquerdo	Bater palmas	Sentar	Levantar	Deslize Direita	Deslize esquerda	Fazer nada
Levantar dois braços	30,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Levantar braço direito	0,00%	10,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Levantar braço esquerdo	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Bater palmas	0,00%	0,00%	0,00%	10,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Sentar	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Levantar	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Deslize Direita	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	20,00%	0,00%	0,00%
Deslize Esquerda	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	20,00%	0,00%
Fazer nada	70,00%	90,00%	100,00%	90,00%	100,00%	100,00%	80,00%	80,00%	100,00%

Tabela 5.5: HMM Baum-Welch com 5 estados escondidos com *threshold* de sensibilidade 0.1

Execução / Reconhecido	Levantar dois braços	Levantar braço direito	Levantar braço esquerdo	Bater palmas	Sentar	Levantar	Deslize Direita	Deslize esquerda	Fazer nada
Levantar dois braços	89,47%	0,00%	0,00%	5,26%	0,00%	0,00%	0,00%	0,00%	0,00%
Levantar braço direito	0,00%	15,79%	0,00%	0,00%	0,00%	0,00%	10,53%	0,00%	0,00%
Levantar braço esquerdo	0,00%	0,00%	47,37%	0,00%	0,00%	0,00%	0,00%	5,26%	0,00%
Bater palmas	0,00%	0,00%	0,00%	52,63%	0,00%	0,00%	0,00%	0,00%	0,00%
Sentar	0,00%	0,00%	0,00%	0,00%	31,58%	0,00%	0,00%	0,00%	0,00%
Levantar	0,00%	0,00%	0,00%	0,00%	0,00%	21,05%	0,00%	0,00%	0,00%
Deslize Direita	0,00%	10,53%	0,00%	0,00%	0,00%	0,00%	26,32%	0,00%	0,00%
Deslize Esquerda	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	5,26%	0,00%
Fazer nada	10,53%	73,68%	52,63%	47,37%	68,42%	78,95%	63,16%	89,47%	100,00%

Após a aplicação de um *threshold* é possível uma melhor noção da sensibilidade e especificidade a partir do desempenho do algoritmo, ou seja a quantidade de vezes em que um gesto é bem identificado e a quantidade de vezes em que um gesto não é executado e também não é identificado. Dependendo da sensibilidade do *threshold* aplicado controla-se a especificidade e sensibilidade geral do algoritmo. Ao diminuir a sensibilidade do *threshold* é possível observar um aumento na sensibilidade na deteção de gestos sem a diminuição da especificidade. No entanto e analisando mais profundamente, existe um aumento de casos em que o gesto é classificado como sendo outro e não como não sendo nenhum. Conclui-se que, o pior cenário encontra-se no aumento dos casos em que o gesto é classificado como sendo outro, pois assim há engano no movimento executado em vez de se avaliar como se não tivesse acontecido nada ocupando a base de dados desnecessariamente.

Tabela 5.6 : HMM Baum-Welch com 10 estados escondidos com *threshold* de sensibilidade 0.1

Execução / Reconhecido	Levantar dois braços	Levantar braço direito	Levantar braço esquerdo	Bater palmas	Sentar	Levantar	Deslize Direita	Deslize esquerda	Fazer nada
Levantar dois braços	47,37%	0,00%	0,00%	0,00%	20,00%	0,00%	0,00%	0,00%	0,00%
Levantar braço direito	0,00%	15,79%	0,00%	0,00%	0,00%	0,00%	5,26%	0,00%	0,00%
Levantar braço esquerdo	0,00%	0,00%	26,32%	0,00%	0,00%	0,00%	0,00%	5,26%	0,00%
Bater palmas	0,00%	0,00%	0,00%	15,79%	0,00%	0,00%	0,00%	0,00%	0,00%
Sentar	0,00%	0,00%	0,00%	0,00%	57,89%	5,26%	0,00%	0,00%	0,00%
Levantar	0,00%	0,00%	0,00%	0,00%	0,00%	42,11%	0,00%	0,00%	0,00%
Deslize Direita	0,00%	15,79%	0,00%	0,00%	0,00%	0,00%	47,37%	0,00%	0,00%
Deslize Esquerda	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	10,53%	0,00%
Fazer nada	52,63%	68,42%	73,68%	84,21%	42,11%	52,63%	47,37%	84,21%	100,00%

O número de estados escondidos é calculado através do método de tentativa e erro, portanto agora a aumenta-se este valor de forma a analisar o desempenho do algoritmo.

Como se observa na tabela acima, com o aumento do número de estados beneficia-se a identificação de alguns movimentos que anteriormente não estavam a revelar uma boa performance.

Tabela 5.7 : HMM Baum-Welch com 10 estados escondidos com *threshold* de sensibilidade 0.05

Execução / Reconhecido	Levantar dois braços	Levantar braço direito	Levantar braço esquerdo	Bater palmas	Sentar	Levantar	Deslize Direita	Deslize esquerda	Fazer nada
Levantar dois braços	42,11%	0,00%	0,00%	5,26%	5,26%	0,00%	0,00%	0,00%	0,00%
Levantar braço direito	0,00%	10,53%	0,00%	0,00%	0,00%	0,00%	5,26%	0,00%	0,00%
Levantar braço esquerdo	0,00%	0,00%	26,32%	0,00%	0,00%	0,00%	0,00%	5,26%	0,00%
Bater palmas	0,00%	0,00%	0,00%	10,53%	0,00%	0,00%	0,00%	0,00%	0,00%
Sentar	0,00%	0,00%	0,00%	0,00%	31,58%	10,53%	0,00%	0,00%	0,00%
Levantar	0,00%	0,00%	0,00%	0,00%	0,00%	42,11%	0,00%	0,00%	0,00%
Deslize Direita	0,00%	10,53%	0,00%	0,00%	0,00%	0,00%	36,84%	0,00%	0,00%
Deslize Esquerda	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	15,79%	0,00%
Fazer nada	57,89%	78,95%	73,68%	89,47%	68,42%	47,37%	57,89%	78,95%	100,00%

A partir da tabela acima conclui-se que não existem ganhos em performance quando se aumenta o número de estados escondidos e se diminui a sensibilidade do *threshold* utilizado.

1.1.1.2. Treino com o algoritmo *Viterbi*

Com o treino *Viterbi*, associado aos *Hidden Markov Models* produzem-se os resultados descritos neste capítulo.

Tabela 5.8 : *Viterbi* com 5 estados escondidos com *threshold* de sensibilidade 0.1

Execução / Reconhecido	Levantar dois braços	Levantar braço direito	Levantar braço esquerdo	Bater palmas	Sentar	Levantar	Deslize Direita	Deslize esquerda	Fazer nada
Levantar dois braços	84,21%	0,00%	0,00%	0,00%	5,26%	0,00%	0,00%	0,00%	0,00%
Levantar braço direito	0,00%	78,95%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Levantar braço esquerdo	0,00%	0,00%	84,21%	0,00%	0,00%	0,00%	0,00%	5,26%	0,00%
Bater palmas	0,00%	0,00%	0,00%	10,53%	0,00%	0,00%	0,00%	0,00%	0,00%
Sentar	0,00%	0,00%	0,00%	0,00%	31,58%	10,00%	0,00%	0,00%	0,00%
Levantar	0,00%	0,00%	0,00%	0,00%	0,00%	10,00%	0,00%	0,00%	0,00%
Deslize Direita	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	94,74%	0,00%	0,00%
Deslize Esquerda	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	68,42%	0,00%
Fazer nada	15,79%	21,05%	15,79%	84,21%	63,16%	80,00%	5,26%	15,79%	100,00%

Tabela 5.9 :*Viterbi* com 10 estados escondidos com *threshold* de sensibilidade 0.1

Execução / Reconhecido	Levantar dois braços	Levantar braço direito	Levantar braço esquerdo	Bater palmas	Sentar	Levantar	Deslize Direita	Deslize esquerda	Fazer nada
Levantar dois braços	94,74%	0,00%	0,00%	5,26%	5,26%	0,00%	0,00%	0,00%	0,00%
Levantar braço direito	0,00%	89,47%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Levantar braço esquerdo	0,00%	0,00%	84,21%	0,00%	0,00%	0,00%	0,00%	5,26%	0,00%
Bater palmas	0,00%	0,00%	0,00%	15,79%	0,00%	0,00%	0,00%	0,00%	0,00%
Sentar	0,00%	0,00%	0,00%	0,00%	36,84%	0,00%	0,00%	0,00%	0,00%
Levantar	0,00%	0,00%	0,00%	0,00%	0,00%	21,05%	0,00%	0,00%	0,00%
Deslize Direita	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	94,74%	0,00%	0,00%
Deslize Esquerda	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	68,42%	0,00%
Fazer nada	5,26%	10,53%	15,79%	78,95%	57,89%	78,95%	5,26%	15,79%	100,00%

Observando a Tabela 5.8 e a Tabela 5.9 conclui-se que, aumentando o número de estados escondidos obtém-se uma melhor performance geral do algoritmo HMM com o treino *Viterbi*.

Tabela 5.10 : *Viterbi* com 10 estados escondidos com *threshold* de sensibilidade 0.05

Execução / Reconhecido	Levantar dois braços	Levantar braço direito	Levantar braço esquerdo	Bater palmas	Sentar	Levantar	Deslize Direita	Deslize esquerda	Fazer nada
Levantar dois braços	100,00%	0,00%	0,00%	5,26%	5,26%	0,00%	0,00%	0,00%	0,00%
Levantar braço direito	0,00%	94,74%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Levantar braço esquerdo	0,00%	0,00%	47,37%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Bater palmas	0,00%	0,00%	0,00%	21,05%	0,00%	0,00%	0,00%	0,00%	0,00%
Sentar	0,00%	0,00%	0,00%	0,00%	73,68%	31,58%	0,00%	0,00%	0,00%
Levantar	0,00%	0,00%	0,00%	0,00%	0,00%	26,32%	0,00%	0,00%	0,00%
Deslize Direita	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	94,74%	0,00%	0,00%
Deslize Esquerda	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	89,47%	0,00%
Fazer nada	0,00%	5,26%	52,63%	78,95%	21,05%	42,11%	5,26%	10,53%	100,00%

Diminuindo a sensibilidade e mantendo o número de estados em dez, obtêm-se os resultados explícitos na Tabela 5.10. Assim pode-se concluir que esta será a melhor combinação de estados escondidos com sensibilidade.

5.3.3. Desempenho *online*

Como é recolhida uma grande quantidade de dados em cada *frame* e cada segundo tem aproximadamente 31 *frames*, é importante que a avaliação do gesto a ser executado seja feita em tempo real. No entanto isso pode colocar entraves na aplicação do algoritmo, pois pode começar a haver uma acumulação de tarefas entupindo o sistema.

Tendo em conta estas situações foi verificado o desempenho do algoritmo, quando se grava e identifica os gestos em tempo real.

Conclui-se que, a partir do momento que o algoritmo se treinou, não existe grande dificuldade em que o reconhecimento se faça em tempo real.

5.4. Algoritmo DTW (*Dynamic Time Warping*)

5.4.1. DTW Standard

O Algoritmo *Dynamic Time Warping* (DTW) mede similaridades entre padrões que podem variar em duração. O conceito principal do algoritmo é comparar características de um padrão com uma referência previamente registada. Como tal, este algoritmo tem sido utilizado em análise de som e vídeo. Este ganhou popularidade devido à sua enorme eficiência na medida da similaridade tempo/série, minimizando os efeitos de distorção no tempo e permitindo uma transformação elástica da série temporal. Assim detetam-se as similaridades independentemente da fase (Figura 5.7) [48] [49].

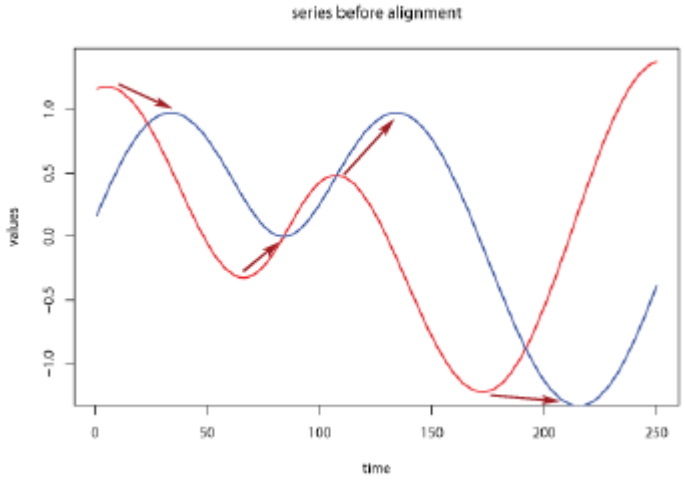


Figura 5.7 :Funcionalidades do algoritmo DTW [52]

O objetivo é comparar duas sequências dependentes do tempo $X = (x_1, x_2, x_3 \dots, x_n)$ e $Y = (y_1, y_2, y_3 \dots, y_m)$ de comprimentos N e M. Em primeiro lugar calcula-se o custo local que corresponde à distância Euclidiana entre x e y, sendo que o objetivo será obter o mínimo de custo total $C = \sum c(x_n, y_m)$ [50] [51].

A distância Euclidiana é calculada através da fórmula:

$$d(p,q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + (p_3 - q_3)^2}$$

Na qual $(p_1 p_2 p_3)$ é um ponto e $(q_1 q_2 q_3)$ é outro ponto.

O caminho ótimo para alinhar estas duas curvas será então definido por uma série de custos pequenos, como se pode observar na Figura 5.8, em que o *Reference index* é a curva de

origem e o *Query index* corresponde à curva que se quer comparar. Assim pode-se dizer que o caminho deverá obedecer aos "vales" [52].

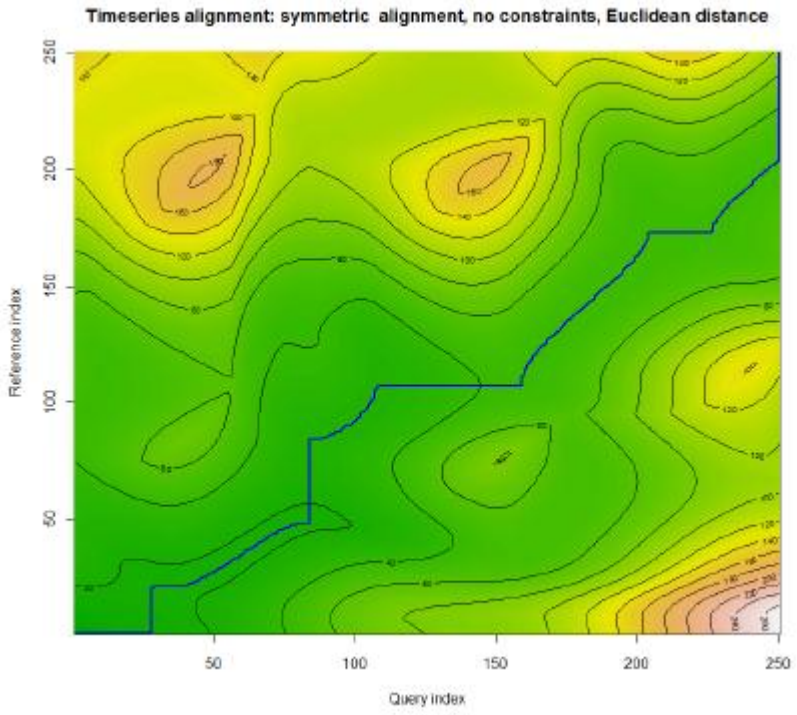


Figura 5.8 : Diagrama de custos com caminho ideal [52]

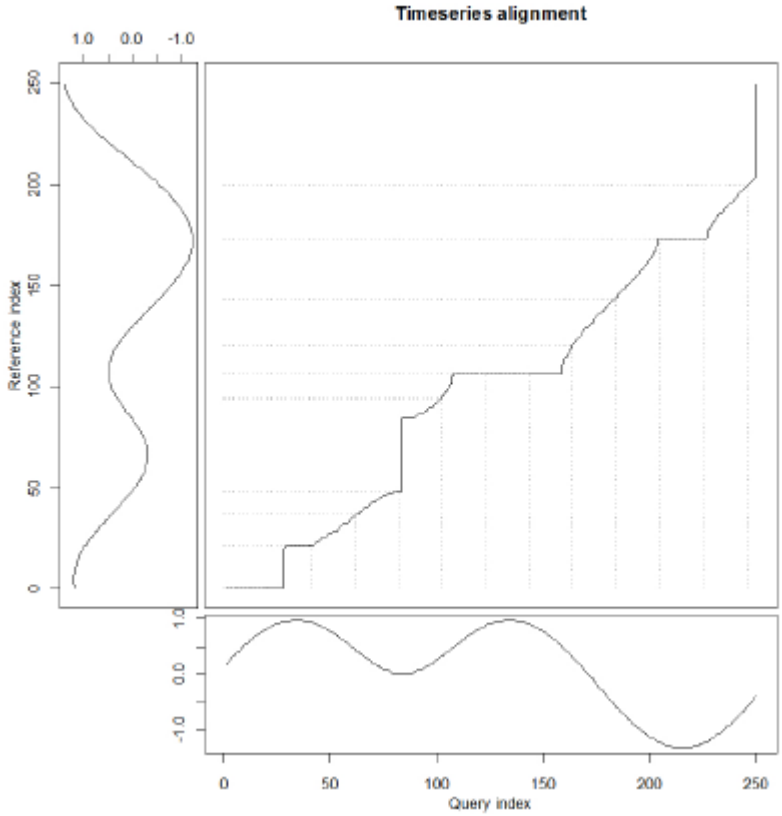


Figura 5.9 :Diagrama de caminho ideal [52]

Na Figura 5.9 é explícito o caminho ideal construído com as curvas correspondentes, a original e a que se quer comparar. Como se pode observar existem por vezes alguns saltos no tempo quando existe uma linha na horizontal e uma na vertical, o que corresponde ao facto de, num determinado tempo N ou M, a distância Euclidiana é inferior se se comparar, por exemplo $c(x_{n+1}, y_m)$ com $c(x_{n+1}, y_{m+1})$ [52]. Ou seja, tendo $c(x_{n+1}, y_m) < c(x_{n+1}, y_{m+1})$ encontra-se uma linha horizontal ou vertical conforme o sistema de coordenadas usado.

Para construir este caminho, para além do custo mínimo existem também mais três critérios [53]:

1- O ponto de início e final do caminho correspondem ao primeiro e ao último ponto das sequências alinhadas. [50]

2- A ordenação original do tempo não pode ser violada, ou seja, o tempo nunca pode andar para trás.

3- O tamanho dos saltos tem que ser definido de forma a não termos um caminho com muitas zonas verticais e horizontais e haver coerência na aplicação do algoritmo.

Como se pode observar na Figura 1 estão explicitas as diferentes condições violadas e o modo como é afetado o caminho ideal [48].

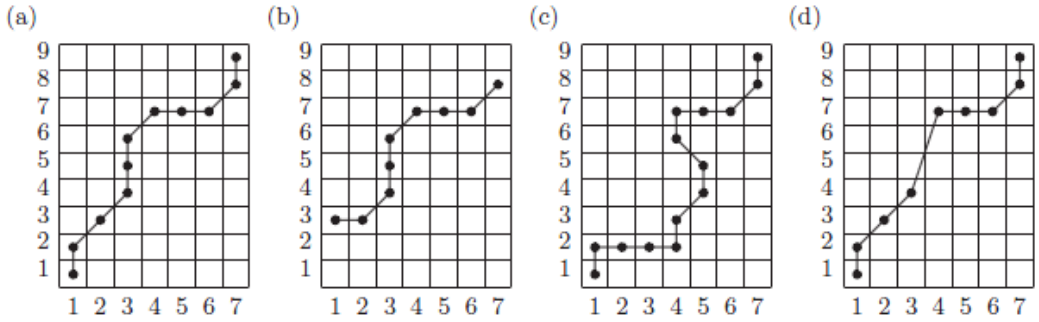


Figura 1 : (a) As condições são todas satisfeitas (b) 1ª condição violada (c) 2ª condição violada (d) 3ª condição violada [52]

Este algoritmo difere do HMM na medida em que a maior importância dá-se à distância entre duas observações.

5.4.1. DTWMean

O algoritmo DTWMean consiste num processo alternativo de medidas de distância. As distâncias entre o gesto a testar e o resto dos gestos foi medida como no algoritmo DTW, ou seja até aqui considera-se o processo standard. No entanto, depois desta fase foram agrupadas as distâncias referentes a gestos iguais e de seguida foi calculada a média dessas distâncias. No final foi encontrada a distância média mínima existente. Se esta se encontrar acima do *threshold*, há um gesto identificado, caso contrário não é identificado nenhum.

5.5. Implementação do algoritmo DTW

O algoritmo DTW foi implementado sem recurso a nenhuma *framework*, o que possibilita uma maior elasticidade de afinação mais simples de parâmetros ou até mudanças no processo de cálculo de distâncias entre gestos.

Os gestos têm a duração de 32 *frames*, sendo que a taxa de aquisição de frames utilizada é de 31 frames por segundo. De forma a maximizar a capacidade de reconhecimento dos gestos, estes devem iniciar-se na primeiro *frame* e terminar na última. Depois da gravação do gesto passa-se ao modo de leitura que, possivelmente vai reconhecer um gesto quando gravado previamente.

5.5.1. Análise dos dados

Por esta altura o tratamento dos dados está completo. De seguida vem o reconhecimento do gesto em si através da avaliação das distâncias Euclidianas no ponto (i-1,j-1) e nos restantes pontos (i-1,j), (i,j) e (i,j-1) de forma a se poder concluir qual o caminho mais provável.

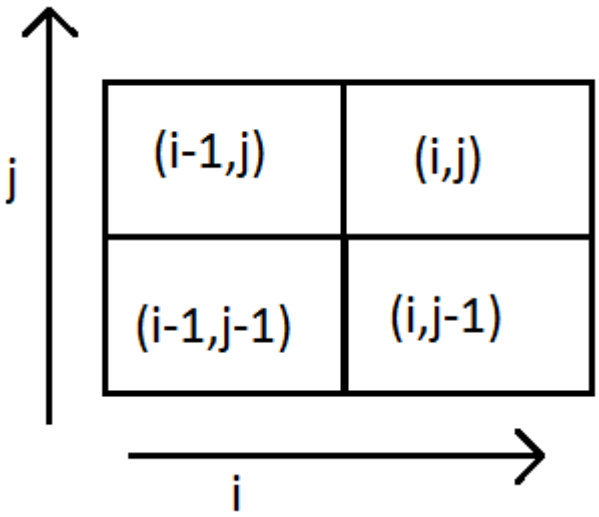


Figura 5.10: Esquema de análise da variável *tab*

Este processo vai apenas achar as distâncias entre vários gestos e encontrar qual o gesto que mais se aproxima do que se quer testar.

5.5.2. Obtenção de resultados

Os testes efetuados com o algoritmo DTW seguem o mesmo princípio do que os efetuados com o algoritmo HMM, mas neste caso não exige um processo de treino. Assim é necessário comparar sempre as distâncias entre o gesto efetuado e cada um dos gestos gravados.

5.5.3. Treino com DTW Standard

Nesta fase são analisados os resultados obtidos com o algoritmo DTW Standard.

Tabela 5.11 : Algoritmo DTW global *threshold*:2; first *threshold*:6; deslocamento máximo:5;distância mínima:20

Execução / Reconhecido	Levantar dois braços	Levantar braço direito	Levantar braço esquerdo	Bater palmas	Sentar	Levantar	Deslize Direita	Deslize esquerda	Fazer nada
Levantar dois braços	78,95%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Levantar braço direito	0,00%	100,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Levantar braço esquerdo	0,00%	0,00%	73,68%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Bater palmas	0,00%	0,00%	0,00%	100,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Sentar	0,00%	0,00%	0,00%	0,00%	89,47%	31,58%	0,00%	0,00%	36,84%
Levantar	0,00%	0,00%	0,00%	0,00%	10,53%	21,05%	0,00%	0,00%	36,84%
Deslize Direita	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	100,00%	0,00%	0,00%
Deslize Esquerda	0,00%	0,00%	21,05%	0,00%	0,00%	0,00%	0,00%	100,00%	0,00%
Fazer nada	21,05%	0,00%	5,26%	0,00%	0,00%	47,37%	0,00%	0,00%	26,32%

Tabela 5.12 Algoritmo DTW global *threshold*:1; first *threshold*:3; deslocamento máximo:3;distância mínima:12

Execução / Reconhecido	Levantar dois braços	Levantar braço direito	Levantar braço esquerdo	Bater palmas	Sentar	Levantar	Deslize Direita	Deslize esquerda	Fazer nada
Levantar dois braços	26,32%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Levantar braço direito	0,00%	47,37%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Levantar braço esquerdo	0,00%	0,00%	63,16%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Bater palmas	0,00%	0,00%	0,00%	100,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Sentar	0,00%	0,00%	0,00%	0,00%	73,68%	31,58%	10,53%	0,00%	0,00%
Levantar	0,00%	0,00%	0,00%	0,00%	15,79%	42,11%	0,00%	0,00%	0,00%
Deslize Direita	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	68,42%	0,00%	0,00%
Deslize Esquerda	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	89,47%	0,00%
Fazer nada	73,68%	52,63%	36,84%	0,00%	10,53%	26,32%	21,05%	10,53%	100,00%

Tabela 5.13 : Algoritmo DTW global *threshold*:0,6; first *threshold*:2; deslocamento máximo:2;distância mínima:10

Execução / Reconhecido	Levantar dois braços	Levantar braço direito	Levantar braço esquerdo	Bater palmas	Sentar	Levantar	Deslize Direita	Deslize esquerda	Fazer nada
Levantar dois braços	10,53%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Levantar braço direito	0,00%	15,79%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Levantar braço esquerdo	0,00%	0,00%	26,32%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Bater palmas	0,00%	0,00%	0,00%	15,79%	0,00%	0,00%	0,00%	0,00%	0,00%
Sentar	0,00%	0,00%	0,00%	0,00%	31,58%	0,00%	0,00%	0,00%	0,00%
Levantar	0,00%	0,00%	0,00%	0,00%	0,00%	42,11%	0,00%	0,00%	0,00%
Deslize Direita	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	36,84%	0,00%	0,00%
Deslize Esquerda	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	63,16%	0,00%
Fazer nada	89,47%	84,21%	73,68%	16/19	68,42%	57,89%	63,16%	36,84%	100,00%

Como se observa pelos resultados obtidos, quanto mais restritivos são os parâmetros aplicados aquando da criação do algoritmo, menor vai ser a sensibilidade do algoritmo. Este exercício

traz algumas vantagens, pois apesar de, muitas vezes não haver muita sensibilidade, temos especificidade de 100%, como se observa na Tabela 5.13 e na Tabela 5.12. Na Tabela 5.11, com parâmetros menos restritivos observa-se que na correspondência *Fazer nada – Fazer nada* encontra-se o valor diferente de 100%, ou seja, mesmo que uma pessoa não esteja a efetuar nenhum gesto, a base de dados fica cheia com gestos detetados. Segundo estes resultados, o melhor compromisso será a configuração explícita na Tabela 5.12.

5.5.4. Treino com DTWmean

De seguida são apresentados os resultados obtidos com a variação do algoritmo DTW Standard, o DTWMean.

Tabela 5.14 : Algoritmo DTWmean global *threshold*:0,6; first *threshold*:2; deslocamento máximo:2;distância mínima:10

Execução / Reconhecido	Levantar dois braços	Levantar braço direito	Levantar braço esquerdo	Bater palmas	Sentar	Levantar	Deslize Direita	Deslize esquerda	Fazer nada
Levantar dois braços	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Levantar braço direito	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Levantar braço esquerdo	0,00%	0,00%	5,26%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Bater palmas	0,00%	0,00%	0,00%	21,05%	0,00%	0,00%	0,00%	0,00%	0,00%
Sentar	0,00%	0,00%	0,00%	0,00%	90,00%	90,00%	0,00%	0,00%	0,00%
Levantar	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Deslize Direita	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	15,79%	0,00%	0,00%
Deslize Esquerda	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	31,58%	0,00%
Fazer nada	100,00%	100,00%	94,74%	78,95%	10,00%	10,00%	0,00%	68,42%	100,00%

Na forma com parâmetros mais restritos (Tabela 5.14) é observável a diminuição da sensibilidade em relação ao algoritmo DTW Standard nas mesmas condições, isto é consequência da natureza das comparações que o novo algoritmo faz.

Tabela 5.15 : Algoritmo DTWmean global *threshold*:1; first *threshold*:3; deslocamento máximo:3;distância mínima:12

Execução / Reconhecido	Levantar dois braços	Levantar braço direito	Levantar braço esquerdo	Bater palmas	Sentar	Levantar	Deslize Direita	Deslize esquerda	Fazer nada
Levantar dois braços	10,53%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Levantar braço direito	0,00%	42,11%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Levantar braço esquerdo	0,00%	0,00%	89,47%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Bater palmas	0,00%	0,00%	0,00%	80,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Sentar	0,00%	0,00%	0,00%	0,00%	31,58%	36,84%	15,79%	0,00%	0,00%
Levantar	0,00%	0,00%	0,00%	0,00%	15,79%	10,53%	0,00%	0,00%	0,00%
Deslize Direita	0,00%	0,00%	0,00%	20,00%	0,00%	0,00%	57,89%	0,00%	0,00%
Deslize Esquerda	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	78,95%	0,00%
Fazer nada	89,47%	57,89%	10,53%	0,00%	52,63%	52,63%	78,95%	21,05%	100,00%

Na Tabela 5.15 são modificados os parâmetros do algoritmo, mas mesmo assim, comparando com os resultados obtidos no treino DTW Standard, não existe grande diferença nem uma melhoria excecional. Isto acontece devido novamente à aplicação da média o que leva à perda alguma sensibilidade nas afinações que se poderão efetuar.

5.5.5. Desempenho *online*

No caso deste algoritmo, o desempenho *online*, em comparação com o algoritmo HMM é muito pior a partir do momento que existem, por exemplo mais que 12 gestos gravados no ficheiro. Isto acontece devido ao facto de termos de comparar o gesto a identificar com todos os gestos gravados, ou seja, quantos mais gestos gravados, mais tempo se vai despender neste processo.

Conclui-se assim que a identificação *online* com um maior número de gestos, será melhor usar o algoritmo HMM.

5.6. Discussão

A partir dos resultados obtidos é possível descrever objetivamente o desempenho dos algoritmos apresentados com as várias versões testadas. Na Tabela 5.16 pode-se observar a sensibilidade média e a especificidade alcançada em cada caso.

A medida de sensibilidade é obtida através da Equação 5.2, onde VP são os casos verdadeiros positivos (gestos bem identificados) e FN os casos falsos negativos (gestos mal identificados).

$$Sensibilidade = \frac{VP}{VP + FN}$$

Equação 5.2: Sensibilidade

A medida de especificidade é obtida através da Equação 5.3, onde VN são os casos verdadeiros negativos (gesto não identificado quando não existe nenhum gesto executado) e FP os casos falsos positivos (gesto não identificado quando não existe um gesto executado).

$$Especificidade = \frac{VN}{VN + FP}$$

Equação 5.3: Especificidade

Tabela 5.16: Desempenho dos algoritmos

		DTW	DTWmean	HMM Baum-Welch	HMM *Viterbi*
Set 1	**Especificidade**	26,32%	15,79%	100%	100%
	Sensibilidade	82,89%	78,95%	36,18%	68,42%
Set 2	**Especificidade**	100%	100%	100%	100%
	Sensibilidade	63,82%	50,13%	32,89%	63,16%
Set 3	**Especificidade**	100%	100%	100%	100%
	Sensibilidade	30,26%	20,46%	26,97%	57,83%

Cada set representa os resultados obtidos com as configurações cada vez mais restritivas. Através das conclusões obtidas na parte dos resultados e analisando o panorama geral na Tabela 5.16, conclui-se que os melhores resultados são obtidos com o uso do algoritmo HMM *Viterbi* com o *threshold* no mínimo, portanto este é o algoritmo escolhido para a aplicação.

Verificou-se, igualmente, que o movimento de levantar os dois braços simultaneamente pode ter inúmeras traduções, ou seja cada pessoa vai efetuar o movimento de maneira diferente. Por outro lado era possível que, com duas ou três instruções, houvesse uma melhoria significativa dos resultados obtidos chegando-se a obter sensibilidades de 90%. No entanto deixou-se que os indivíduos executassem os movimentos da maneira que achassem mais natural, sem intervenções, pois desta maneira há aproximação ao cenário mais realista possível.

6. Deteção de faces e expressões faciais

Neste capítulo são abordados os métodos de deteção e identificação de pessoas pela face e como se poderá detetar as expressões faciais.

6.1. Deteção de expressões faciais

As expressões faciais são importantes pois, a par da linguagem gestual, são uma maneira de comunicar emoções. Assim é necessário implementar um sistema que permita a perceção da expressão do paciente de modo a extrair mais informação útil.

Neste tipo de patologias, como referido anteriormente, os pacientes têm muita dificuldade na interpretação e transmissão de emoções. A deteção de expressões faciais, não só pode ser uma maneira de facilitar a monitorização, como pode ser usada a avaliar o desempenho dos pacientes ao longo de sessões de terapia.

A Microsoft disponibiliza um *SDK* de *FaceTracking* para o *Kinect* e em conjunto com o *Kinect SDK* possibilita a criação de projetos que fazem monitorização de faces humanas em tempo real.

O *SDK* permite analisar as informações oferecidas pelo sensor *Kinect* e deduzir expressões faciais e posições da cabeça.

6.1.1. Recolha de dados

As informações que se podem obter incluem a monitorização de 87 pontos em 2D da face de uma pessoa. Estes pontos incluem limitação da cara, olhos, nariz e sobrancelhas, como se pode verificar pela Figura 6.1.

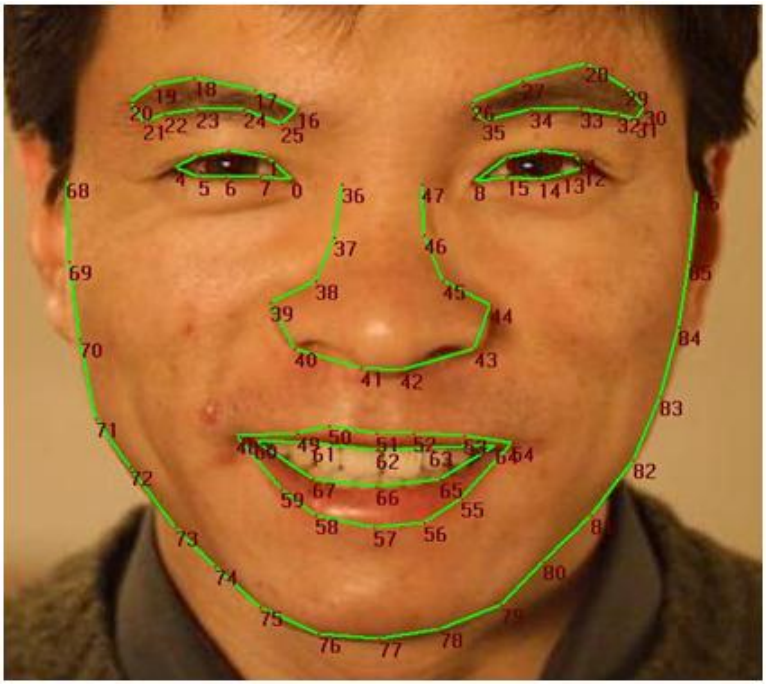


Figura 6.1: Conjunto de pontos monitorizados [31]

Além destes pontos ainda é possível ir buscar o centro dos olhos. Outra informação que se pode extrair é a inclinação 3D da cabeça a partir das coordenadas X, Y e Z. Assim é possível saber se a pessoa está a acenar positivamente, negativamente ou intermédio, como se pode verificar pela Figura 6.2.

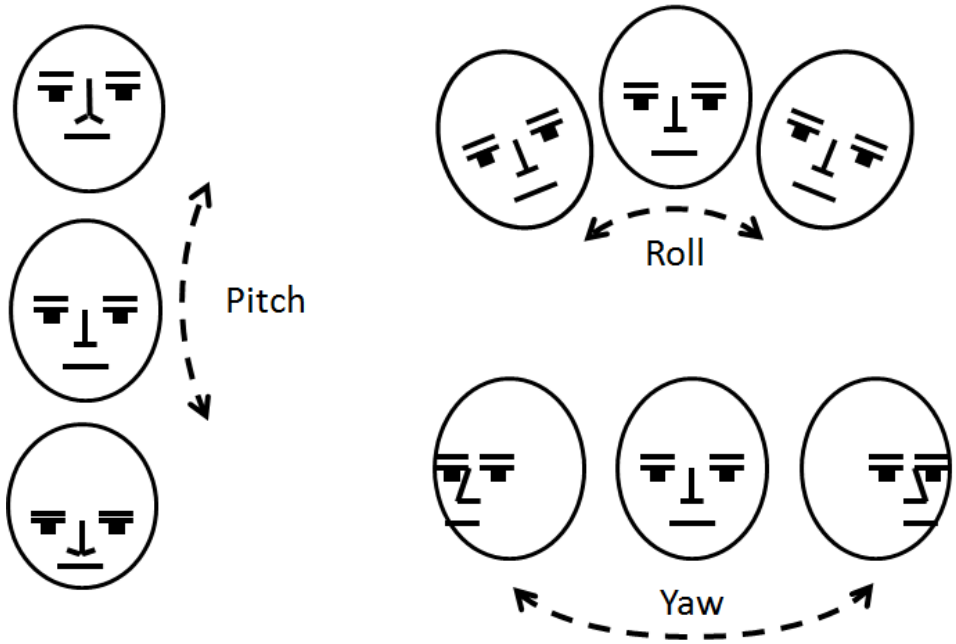


Figura 6.2: Posição da cabeça [31]

A saída *Animation Units (AU)* é principalmente usada quando é necessário animar um avatar. Existe à disposição um conjunto de seis AU que apresentam valores ente -1 e 1, sendo que 0 é uma posição neutra. Assim, por exemplo se o valor de AU4, correspondente à depressão do canto dos lábios, for -1 é indicação de um sorriso enquanto que se este valor for +1 representa uma cara triste (Figura 6.3) [31].

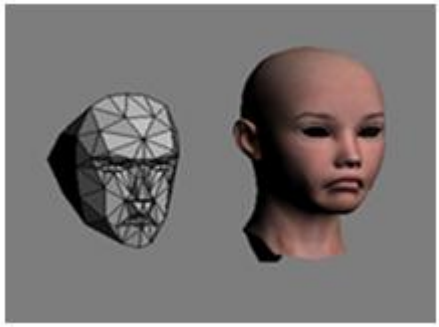


Figura 6.3: Cara triste (AU4=+1) [31]

6.1.1. Resultados e discussão

Após a análise das potencialidades do sensor *Kinect* no campo da deteção e identificação de expressões faciais, procede-se ao teste desta tecnologia.

Foi implementada uma simples aplicação de forma a recolher os *Animation Units* produzidos pelo *software* do *Kinect* aquando da identificação de uma face.

A conclusão deste simples teste é que realmente estas expressões faciais são detetadas conforme a variação dos índices das várias AU. Isto vai ao encontro do funcionamento descrito anteriormente. No entanto pode-se também tirar a conclusão que estes índices são apenas detetados quando a face se encontra perfeitamente enquadrada frontalmente em relação ao sensor.

Portanto, ao usar este tipo de tecnologia na vida real é essencial que o individuo que se queira avaliar se mostre disposto a direcionar a sua face para o sensor *Kinect*.

6.2. Deteção de pessoas pela face

Nesta fase é estudada a possibilidade de detetar faces humanas e fazer corresponder estas a um paciente da base de dados. Assim, sempre que um paciente se apresentar em frente do sensor, vai ser possível fazer a sua identificação automática.

A fim de se efetuar esta análise é necessário implementar algumas tecnologias além do sensor *Kinect*. O sensor vai ser usado para retirar pontos importantes da face que possam ser usados de forma a reconhecer pessoas. Uma simples câmara vai ser utilizada de forma a retirar uma imagem 2D da face [54].

6.2.1. Técnicas de reconhecimento de faces

Técnicas de reconhecimento de pessoas consistem na análise de vários parâmetros característicos. Estes parâmetros podem ser a cor da pele, olhos ou cabelo, a altura ou a etnia. No entanto estes parâmetros são pouco discriminativos aquando da distinção de duas pessoas [55]. Além de que estes podem alterar consoante exposição solar ou uso de calçado.

Por outro lado existem mais tipos de dados que se podem extrair de forma a identificar alguém de forma mais precisa.

Alguns algoritmos de reconhecimento de faces funcionam através da extração de características-chave. Normalmente estas representam a posição, tamanho ou forma dos olhos, nariz e boca. Estas são utilizadas em termos de comparação numa base de dados de forma a encontrar a pessoa [56] [57].

Existem outros algoritmos que consistem na composição de uma galeria de imagens de faces e de seguida estas são comprimidas e normalizadas de forma a restar apenas informação útil à identificação de pessoas. De seguida uma imagem teste é comparada com estas informações [58] [59].

Uma das últimas técnicas utilizadas com sucesso consiste na correspondência de modelos associado a propriedades faciais salientes, sendo que a partir daqui é possível construir o modelo da face da pessoa [58].

Os algoritmos de reconhecimento de faces podem ser divididos em dois grandes grupos, os que usam a geometria que se baseiam em distinção de características e os fotométricos que decompõem as imagens em números e usam técnicas estatísticas de forma a comparar modelos e eliminar variâncias. Os métodos de análise mais conhecidos são o PCA (*principal component analysis*), HMM (*hidden markov models*), e LDA (*linear discriminative analysis*) [56] [60].

A tecnologia mais atual em termos de análise de faces vem da computação 3D de imagens. Esta técnica consiste na recolha de características faciais como o contorno das formas do nariz e dos olhos. Com esta pode-se estar a aumentar as probabilidades de reconhecimento de uma pessoa em qualquer ambiente, pois a luminosidade é eliminada da equação. No entanto esta técnica continua a ter a desvantagem da sensibilidade às expressões faciais.

A análise da textura da pele também se começa a revelar como uma alternativa aos métodos convencionais. Esta consiste no reconhecimento de sinais, padrões e linhas únicas na pele [56].

6.2.2. Recolha de *features* pelo *Kinect*

Sabe-se então que existem vários *features* que se podem extrair. Através do sensor *Kinect* encontram-se as coordenadas de pontos-chave da face. Os pontos mais pertinentes de se observar na minha opinião serão os que alteram com menos facilidade nomeadamente através de expressões faciais.

O método *Get3DShape* do *toolkit FaceRecognision* do *Kinect SDK* permite recolher o vetor posição em 3D de todos os pontos faciais obtidos no processo de recolha de dados.

Muitos destes pontos permitem medir distâncias entre olhos e entre extremidades da face que podem servir para identificar a pessoa que se encontra em frente da câmara. Esta biometria da face em conjunto com todos os parâmetros recolhidos permitem também criar uma máscara virtual para cada pessoa, podendo isto ser usado para fins lúdicos ou terapêuticos.

6.2.2.1. Análise dos dados do *Kinect*

De forma a ser possível, através dos *features* recolhidos pelo *Kinect*, o reconhecimento de uma pessoa procede-se primeiro à normalização dos dados.

De seguida analisa-se a distância entre a ponta do nariz e o centro do olho direito. Esta é apenas uma das medidas que se podem fazer com a finalidade de distinguir duas faces.

No entanto concluiu-se que os dados obtidos não são bons para se continuar com o estudo, pois estas medidas entre dois pontos da face variam muito de *frame* em *frame*, mesmo que a pessoa mantenha uma posição estável em frente da câmara.

A precisão da câmara foi medida em [61] e [62] com resultados que variam entre o erro de alguns milímetros em distâncias de aproximadamente 0,5 metros e 4cm em distâncias até 5 metros da câmara. A precisão vai diminuir sempre com a distância à câmara.

Este comportamento do sensor vai comprometer a medida de distâncias da ordem de grandeza dos milímetros, dificultando muito a recolha de dados com qualidade suficiente para se distinguir duas faces. Desta forma, o sensor *Kinect* foi excluído desta tarefa e foi utilizada uma câmara.

6.2.3. Extração de *features* com uma câmara

Através da câmara é possível recolher outro tipo de dados diferente. Aqui recolhe-se uma imagem a preto e branco da face da pessoa de forma a minimizar a influência da iluminação e do fundo da imagem. Na técnica utilizada estão envolvidos os métodos de análise das componentes principais (PCA) e uma rede neuronal aplicados através duma biblioteca de visão computacional.

6.2.3.1. Análise dos dados da câmara

De forma a ser possível a análise de imagens recolhidas da câmara é necessário recorrer a uma biblioteca de visão computacional. A *Open source computer vision* (*OpenCV*) é uma biblioteca desenvolvida para as seguintes linguagens de programação: C, C++, *Python* e *Java*. É de livre uso tanto para fins académicos com para fins comerciais. Como o próprio nome indica esta trata-se de uma biblioteca que possibilita o tratamento de imagens na área da visão computacional. O *EgmuCV* é uma plataforma cruzada que representa um *wrapper* .NET da *OpenCV*. Este permite a chamada das funções da *OpenCV* através das linguagens compatíveis com .NET como C#,VB e VC++ [63].

Com esta biblioteca e este *wrapper* é possível desenvolver uma aplicação que identifique a cara das pessoas através de uma câmara normal.

Ao ser desenvolvido em C# é mais simples a interação com a aplicação que deteta gestos e expressões faciais. A ideia é juntar todas as aplicações de forma a serem disponibilizados mais serviços ao utilizador.

6.2.4. Tratamento dos dados

Nesta fase está escolhido o método que mais promete melhores resultados, ou seja, a extração de *features* com a câmara em conjunto com os métodos de análise disponibilizados pela OpenCV. Esta explicação está disponível no tutorial sobre reconhecimento de faces em [64].

O problema com a representação de imagens resulta da elevada dimensionalidade. Uma imagem apenas com cores na escala do cinzento e a 2D $p \times q$ com apenas 100×100 pixels, fica com um espaço dimensional de 10000. No entanto nem todas estas dimensões são uteis neste processo. Assim é necessário perceber quais as dimensões que fornecem mais informação. A análise dos componentes principais oferece um meio de resolver este problema.

Através da análise das componentes principais transforma-se um conjunto de variáveis possivelmente correlacionadas num conjunto de variáveis não correlacionadas. Um conjunto de dados com elevada dimensionalidade passa a ser descrito através de um conjunto com menos dimensões perdendo o mínimo de informação possível. Assim é necessário calcular as componentes principais que servem de guia para determinar as maiores variâncias nos dados.

Sendo $X = \{x_1, x_2, \dots, x_n\}$ um vetor de observações com $x_i \in R^d$

Primeiro é necessário calcular o valor médio

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i.$$

De seguida calcula-se a matriz covariância

$$S = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)(x_i - \mu)^T$$

Calcular os vetores próprios v_i e valores próprios λ_i de S correspondentes: $Sv_i = \lambda_i v_i$, com $i = 1{,}2, \dots, n$

No próximo passo ordenam-se os vetores próprios por ordem decrescente de valor principal. As K componentes principais são os vetores próprios correspondentes aos K maiores valores próprios.

Estas componentes principais do vetor X em causa são calculadas do seguinte modo:

$$y = W^T(x - \mu), \text{ onde } W = (v_1, v_2, \dots, v_k).$$

Assim conclui-se que a reconstrução a partir do PCA é dada por $x = Wy + \mu$

O método das faces principais reconhece as faces através dos seguintes passos: projetar todas as amostras treinadas no subespaço do PCA, projetar a face a determinar no subespaço PCA e encontrar o *nearest neighbor* entre a imagem de treino e a imagem que se quer identificar [54].

No entanto existe um problema para resolver, se houver 400 imagens de 100×100 pixels significa que a análise PCA resolve a matriz covariância $S = XX^T$, onde o tamanho $T = 1000 \times 400$, o que ocupa muito espaço. De forma a minimizar esta situação é necessário aplicar mais um passo adicional. Numa matriz $M \times N$ pode apenas haver $N - 1$ valores próprios. Então faz-se a decomposição de valores próprios $S = X^T X$ de tamanho $N \times N$. Tendo no final $X^T X v_i = \lambda_i v_i$.

Os vetores próprios originais $S = XX^T$ são obtidos por:

$$XX^T(Xv_i) = \lambda_i(Xv_i)$$

6.2.5. Resultados obtidos

O teste efetuado ao algoritmo escolhido foi efetuado com nove indivíduos. Em cada indivíduo foram feitas gravações da face nas mesmas condições. Depois de todas as faces gravadas posicionaram-se um a um em frente da câmara vinte vezes de forma a poder ter um espectro de resultados maiores.

Os resultados obtidos através do método sugerido acima são os seguintes:

Tabela 6.1: Resultados na deteção de pessoas pela face

É / Reconhecido	Individuo 1	Individuo 2	Individuo 3	Individuo 4	Individuo 5	Individuo 6	Individuo 7	Individuo 8	Individuo 9
Individuo 1	100,00%	5,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Individuo 2	0,00%	85,00%	0,00%	0,00%	10,00%	0,00%	0,00%	0,00%	0,00%
Individuo 3	0,00%	0,00%	100,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Individuo 4	0,00%	0,00%	0,00%	100,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Individuo 5	0,00%	10,00%	0,00%	0,00%	90,00%	0,00%	0,00%	0,00%	0,00%
Individuo 6	0,00%	0,00%	0,00%	0,00%	0,00%	95,00%	0,00%	5,00%	0,00%
Individuo 7	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	100,00%	0,00%	0,00%
Individuo 8	0,00%	0,00%	0,00%	0,00%	0,00%	5,00%	0,00%	90,00%	0,00%
Individuo 9	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	100,00%
Desconhecido	0,00%	5,00%	0,00%	0,00%	0,00%	0,00%	0,00%	5,00%	

6.2.6. Discussão

O algoritmo aqui utilizado permite a identificação de uma pessoa em frente de uma câmara. No entanto, como se verifica através da análise da Tabela 6.1, o desempenho do algoritmo é razoável.

Ao longo do teste deste algoritmo foram encontradas diversas dificuldades. Em primeiro lugar, a iluminação do rosto do individuo é crucial na identificação. As imagens da face que são gravadas podem ser obtidas duma maneira não muito eficiente o que influencia negativamente os resultados. Dois indivíduos com fisionomias parecidas são muito difíceis de

distinguir, fato que se pode corrigir quase na totalidade com regravação ou adição de imagens da face. A partir da Tabela 6.1 verifica-se que o indivíduo 2 e o indivíduo 5 são mais dificilmente distinguidos, isto deve-se à aparência semelhante.

No entanto, se dois indivíduos forem bastante diferentes em termos de forma da face e cor de pele, estes são bem distinguíveis. Um fator que também influencia a deteção e identificação certa de uma pessoa prende-se com o penteado que se possa ter.

De qualquer das maneiras, estes resultados foram obtidos de maneira ótima. Ou seja, sempre que foram precisas novas imagens da face para que o algoritmo se comportasse melhor, estas foram adquiridas. Além de que as condições de iluminação e ambiente foram iguais para todos os indivíduos e entre sessões.

7. Conclusão

A presente aplicação foi desenvolvida ao longo do ano letivo 2012/13 e incluída no projeto final de Mestrado Integrado em Engenharia Biomédica no âmbito do projeto *HomeTech*. Este tem como finalidade implementar, numa casa habitada com pacientes com doenças do espectro do autismo, um sistema de vigilância inteligente. Este sistema de vigilância inteligente permite enviar alertas para um cuidador. Este cuidador, a partir das informações recebidas consegue monitorizar de uma maneira ótima o comportamento dos pacientes assim como a segurança da casa em geral.

No decorrer do projeto foram desenvolvidas variantes de algoritmos de deteção e avaliação de sequências temporais de forma a ir ao encontro das necessidades. De maneira a ser escolhido o algoritmo com melhor performance foram utilizadas medidas de sensibilidade e especificidade.

Na análise de movimentos corporais, o algoritmo com melhor performance foi o *Hidden Markov Models* em conjunto com o tipo de treino *Viterbi*. O desempenho *online* do algoritmo foi satisfatório uma vez que, não exige a alocação de muitos recursos. Assim não há necessidade de armazenar grandes quantidades de dados correspondentes aos movimentos efetuados ao longo do dia, este processo atua em tempo útil.

Adicionando a análise e identificação das expressões faciais, dos gestos e das pessoas através da face e em conjunto com uma base de dados foi possível construir uma solução completa. Esta solução traduz-se numa aplicação que pode ser utilizada para detetar gestos, movimentações e expressões faciais de um determinado individuo e em seguida permite guardar estas informações numa base de dados. Esta pode ser consultada através da aplicação. Assim são oferecidos ao cuidador vários serviços que permitem uma vigilância completa dos pacientes a nível comportamental. Deste modo é também possível analisar objetivamente o progresso dos indivíduos ao longo do processo de terapias.

7.1. Trabalho futuro

Os algoritmos testados neste projeto representam os primeiros passos na direção da monitorização inteligente de pacientes com doenças do espetro do autismo. De seguida são sugeridos passos que permitirão obter um produto robusto e funcional.

Em primeiro lugar será necessário realizar um estudo acerca dos movimentos estereotipados individuais dos pacientes. Deste modo vai ser possível gravar os movimentos de forma eficiente e aumentar o desempenho do algoritmo em situações quotidianas.

A aplicação tem que ser testada no "mundo real" para que se possa ter uma noção do desempenho das tecnologias escolhidas. Desta maneira pode-se também estudar a viabilidade da solução e encontrar alternativas que melhorem o desempenho das várias partes.

Deverá também ser criado um sistema estatístico de forma a ser possível o acompanhamento da evolução do paciente ao longo das sessões de terapia.

7.2. Publicações associadas

O artigo do Anexo A foi aprovado para a *International Conference on Health and Social Care Information Systems and Technologies* a ser realizada entre 23 e 25 de Outubro de 2013 em Lisboa.

Referências

[1] "FPDA - Federação Portuguesa de Autismo," [Online]. Available: http://www.appda-lisboa.org.pt/federacao/autismo.php. [Acedido em 30 10 2012].

[2] C. A. Gadia, R. Tuchman e N. T. Rotta3, "Autismo e doenças invasivas de desenvolvimento," *Jornal de Pediatria,* 2004.

[3] "Federação Portuguesa de Autismo," [Online]. Available: http://www.appda-lisboa.org.pt/federacao/autismo.php. [Acedido em 15 10 2012].

[4] D. Strickland, "Virtual Reality for the Treatment of Autism," *Virtual Reality in Neuro-psycho-physiology: Cognitive, Clinical and Methodogical Issues in Assessment and Rehabilitation,* pp. 81-86, 1997.

[5] ISA, "Documento Interno," 2013.

[6] F. D. D. Reed, J. M. Hirst e S. R. Hyman, "Assessment and treatment of stereotypic behavior in children with autism and other developmental disabilities: A thirty year review," *Research in Autism Spectrum Disorders,* vol. 6, p. 422–430, 2012.

[7] S. Michelle e M. Ruud, "Effects of physical exercise on Autism Spectrum Disorders: A meta-analysis," *Research in Autism Spectrum Disorders,* vol. 6, p. 46–57, 2012.

[8] S. J. Spence, "The Genetics of Autism," *Seminars in Pediatric Neurology,* vol. 11, n.º 3, p. 196–204, 2004.

[9] C. K. Gurkan e R. J. Hagerman, "Targeted treatments in autism and fragile X syndrome," *Research in Autism Spectrum Disorders,* vol. 6, p. 1311–1320, 2012.

[10] M. G. Chez, S. Memon e P. C. Hung, "Neurologic Treatment Strategies in Autism: An Overview of Medical Intervention Strategies," *Seminars in Pediatric Neurology,* vol. 11, n.º 3, p. 229–235, 2004.

[11] N. Swangmuang e P. Krishnamurthy, "An effective location fingerprint model for wireless indoor localization," *Pervasive and Mobile Computing,* vol. 4, 2008.

[12] M. M. O., K. S. M. e M. J. H., "Predicting improvement in social–communication symptoms of autism spectrum disorders using retrospective treatment data," *Research in Autism Spectrum Disorders,* vol. 6, p. 535–545, 2012.

[13] A. S. Org., "Autism Speaks, It's time to listen," 2013. [Online]. Available: http://www.autismspeaks.org/. [Acedido em 10 1 2013].

[14] A. B. Cunningham e L. Schreibman, "Stereotypy in autism: The importance of function," *Research in Autism Spectrum Disorders,* vol. 2, p. 469–479, 2008.

[15] M. Rebecca, G. Gina, M. Renee, G. Amy, G. Nicole, A. Jennifer, H. William e S. June, "Stereotypy in young children with autism and typically developing children," *Research in Developmental Disabilities,* vol. 28, p. 266–277, 2007.

[16] T. N. Davis, S. Durand e J. M. C. b, "The effects of a brushing procedure on stereotypical behavior," *Research in Autism Spectrum Disorders,* vol. 5, p. 1053–1058, 2011.

[17] J. L. Matson e D. Neal, "Seizures and epilepsy and their relationship to autism spectrum disorders," *Research in Autism Spectrum Disorders,* vol. 3, p. 999–1005, 2009.

[18] Q. Lê, H. B. Nguyen e T. Barnett, "Smart Homes for Older People: Positive Aging in a Digital World," *Future Internet,* vol. 4, pp. 607-617, 2012.

[19] D. Dan, A. C. Rory, F. P. Paul e L. Fici-Pasquinad, "Sensor technology for smart homes," *Maturitas,* vol. 69, p. 131–136, 2011.

[20] L. C.DeSilva, C. Morikawa e I. M.Petra, "State of the art of smart homes," *Engineering Applications of Artificial Intelligence,* vol. 25, p. 1313–1321, 2012.

[21] "MavHome," [Online]. Available: http://ailab.wsu.edu/mavhome/information.html. [Acedido em 25 10 2012].

[22] G. M. Youngblood, L. B. Holder e a. D. J. Cook, "The MavHome Architecture".*Technical Report CSE-2004-18.*

[23] M. Ohashi, "Japanese Ubiquotous Network Project: Ubila," *Handbook of Ambient Intelligence and Smart Environments,* pp. 1229-1255, 2010.

[24] T. Yamazaki, "The Ubiquitous Home," *International Journal of Smart Home,* vol. 1, pp. 17-22, 2007.

[25] "Austism Society," [Online]. Available: http://www.autism-society.org/living-with-autism/lifespan/adulthood/social-relationships.html. [Acedido em 15 10 2012].

[26] M. J. L., H. M. A. e B. Brian, "Treating adaptive living skills of persons with autism using applied behavior analysis: A review," *Research in Autism Spectrum Disorders,* vol. 6, p. 271–276, 2012.

[27] L. Motion, "https://www.leapmotion.com/product," [Online]. [Acedido em 18 02 2013].

[28] Microsoft, "Kinect for Windows," [Online]. Available: http://www.microsoft.com/en-us/kinectforwindows/. [Acedido em 23 01 2013].

[29] [Online]. Available: http://www.hl7.org/about/index.cfm?ref=nav. [Acedido em 30 10 2012].

[30] "Integration of HL7 Compliant Smart Home Healthcare System and HMIS," *Impact Analysis of Solutions for Chronic Disease Prevention and Management Lecture Notes in Computer Science Volume,* vol. 7251, pp. 230-233, 2012.

[31] microsoft, "msdn," microsoft, [Online]. Available: http://msdn.microsoft.com/en-us/library/jj130970#ID4ENG. [Acedido em 08 12 2012].

[32] S. Jan, J. Michal e P. Tomas, "3D with Kinect," *Consumer Depth Cameras for Computer Vision Advances in Computer Vision and Pattern Recognition,* pp. 3-25, 2013.

[33] J. Giles, "Inside the race to hack the Kinect," *New Scientist,* vol. 208, pp. 22-28.

[34] R. Bautista, "A Criança Autista," em *Necesidades Educativas Especiales,* Dinalivro, 1997.

[35] Y. Wu e T. S. Huang, "Vision-Based Gesture Recognition: A Review," *GW'99, LNAI 1739,* pp. 103-115, 1999.

[36] O. State University, "Department of Science and Engineering," [Online]. Available: http://www.cse.ohio-state.edu/~jwdavis/CVL/Research/MHI/mhi.html.

[37] J. C. Hall, "Creative Distraction," [Online]. Available: http://www.creativedistraction.com/demos/gesture-recognition-kinect-with-hidden-markov-models-hmms/. [Acedido em 3 12 2012].

[38] Wikipédia, "http://en.wikipedia.org/wiki/Hidden_Markov_model#Architecture," [Online]. [Acedido em 28 1 2013].

[39] V. I. Pavlovic, R. Sharma e T. S. Huang, "Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review," *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE,* vol. 19, 1997.

[40] C. d. Souza, "Code Project," [Online]. Available: http://www.codeproject.com/Articles/541428/Sequence-Classifiers-in-Csharp-Part-I-Hidden-Marko. [Acedido em 15 04 2013].

[41] E. S. R, "Hidden Markov Models," *Current Opinion in Structural Biology,* vol. 6, pp. 361-365, 1996.

[42] "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE,* vol. 77, n.º 2, pp. 257 - 286.

[43] O. Dweik e H. Tamimi, "Real-Time Gesture Recognition Usind 3D Images".

[44] C. University. [Online]. Available: https://courses.cit.cornell.edu/info2950_2012sp/vit.pdf. [Acedido em 11 05 2013].

[45] J. G. David Forney, "The Viterbi Algorithm," *Proceedings of the IEEE,* vol. 61, pp. 268-279, 1973.

[46] S. Russel e P. Norvig, "Artificial Intelligence: A Modern Approach," Pearson Education, 2003, pp. 540-541.

[47] E. G. A. M. R. S. Wael Khreich, "On the memory complexity of the forward–backward algorithm," *Pattern Recognition Letters,* vol. 31, pp. 91-99, 2010.

[48] P. Senin, "Dynamic Time Warping Algorithm Review," Information and Computer Science Department, University of Hawaii at Manoa, Honolulu, USA, 2008.

[49] S. Salvador e P. Chan, "Toward accurate dynamic time warping in linear time and space," *Intelligent Data Analysis,* vol. 11, pp. 561-580, 2007.

[50] T. Giorgino, "Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package," *Journal of Statistical Software,* vol. 31, n.º 7, 2009.

[51] "On-line signature verification by dynamic time-warping," *Pattern Recognition,* vol. 3, pp. 38 - 42, 1996.

[52] M. Muller, "Information Retrieval for Music and Motion," 2007.

[53] D. J. G. Harris, "Isolated word, Speech recognition using Dynamic Time Warping towards smart appliances.," [Online]. Available: http://www.cnel.ufl.edu/~kkale/dtw.html. [Acedido em 4 12 2012].

[54] C. Johnson, "EMGU Multiple Face Recognition using PCA and Parallel Optimisation," 2012. [Online]. Available: http://www.codeproject.com/Articles/261550/EMGU-Multiple-Face-Recognition-using-PCA-and-Paral. [Acedido em 23 04 2013].

[55] F. R. D. M. Gian Luca Marcialis, "Group-specific face verification using soft biometrics," *Journal of Visual Languages and Computing,* vol. 20, p. 101–109, 2009.

[56] S. A. G. Rojas, "Multiple face detection and recognition in real time," 2012. [Online]. Available: http://www.codeproject.com/Articles/239849/Multiple-face-detection-and-recognition-in-real-ti. [Acedido em 01 04 2013].

[57] W. Zhao, A. Krishnaswamy, R. Chellappa, D. L. Swets e J. Weng, "Discriminant Analysis of Principal Components for Face Recognition," *NATO ASI Series,* vol. 163, pp. 73-85, 1998.

[58] M. A. Turk e A. P. Pentland, "Face Recognition Using Eigenfaces," *Computer Vision and Pattern Recognition IEEE Computer Society* , pp. 586 - 591, 1991.

[59] S. G. Kong, J. Heo, B. R. Abidi, J. Paik e M. A. Abidi, "Recent advances in visual and infrared face recognition—a review," *Computer Vision and Image Understanding,* vol. 97, n.º 1, p. 103–135, 2005.

[60] V. Bruce e A. Young, "Understanding face recognition," *British Journal of Psychology,* vol. 77, n.º 3, pp. 305-327, 1987.

[61] K. Khoshelham e S. O. Elberink, "Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications," *Sensor,* vol. 12, pp. 1437-1454, 2012.

[62] K. Khoshelham, "Accuracy Analysis of Kinect Depth Data," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences,* Vols. %1 de %2XXXVIII-5/W12, 2011.

[63] F. S. Foundation, "Emgu CV," [Online]. Available: http://www.emgu.com/wiki/index.php/Main_Page. [Acedido em 23 04 2013].

[64] o. d. team, "OpenCV 2.4.5.0 documentation," 5 4 2013. [Online]. Available: http://docs.opencv.org. [Acedido em 11 06 2013].

Anexos

Anexo A

Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Technology 00 (2013) 000–000

Procedia Technology

www.elsevier.com/locate/procedia

CENTERIS 2013 - Conference on ENTERprise Information Systems / HCIST 2013 - International Conference on Health and Social Care Information Systems and Technologies

Gesture Analysis Algorithms

Miguel Ferreira[a*], Andreia Carreiro[bc], António Damasceno[b]

[a]*University of Coimbra, Coimbra, Portugal*
[b]*ISA Intelligent Sensing Anywhere, S.A, Coimbra, Portugal*
[c] *Institute for Systems Engineering and Computers(INESCC), Coimbra, Portugal*

Abstract

In this paper we use a Kinect camera to monitor extensively human movements and evaluate facial expressions in order to implement a smart video surveillance system in a house inhabited with people with Autism Spectrum Disorders. The main problem here is to select the best approach by testing and analyzing several gesture recognition algorithms.
Despite the tests are made on people without Autism Spectrum Disorders, its known that people with Autism Spectrum Disorders usually reveal stereotypic motor behaviors and characteristic reactions to social interactions. In most cases the diagnosis techniques are subjective because they rely on the experience of the parents with the children and the time spent with the doctor is very little in order to be made a complete examination to the children's natural behavior [1] [2] [3].

Autism; Algorithms; C#; Detection of movements and facial expressions; Database; Kinect SDK

1. Introduction

There are many different approaches to diagnose Autism Spectrum Disorders (ASD's) and generally they are very subjective. Mainly there are several tables with questions about the behavior of the toddler. To fill these forms, the person in question must live together every day with the kid in question and be impartial when evaluating the behaviour [4]. Most parents may be in this situation but there is a factor of protection when talking about their kids. Because of this situation many times there are inaccurate answers [5] [6].

Our goal is to make this process less subjective via monitoring systems of the toddler's natural behaviour. In this way there is a need to create an intelligent video surveillance system in order to be possible for the caregivers to have more control over the toddler's behaviours by monitoring movements and reactions of the toddlers to stimuli.

2. Methods

In this case we found out that the Kinect sensor by Microsoft is the most suitable due to its ability to monitor the joints in three dimensions and also get facial expressions. This way we can access a set of features that can be used to detect and identify gestures made in front of the sensor and get the natural reaction to stimuli.

The main goal is to provide an innovative ASD's online monitoring. After the gestures are detected there must be a filtering method in order to differentiate the information that is important to this study and the information that can be discarded. All the data resulting from this process is saved in a database with the patient's personal information.

2.1. Data retrieval

The raw data taken from the Kinect sensor corresponds to the X, Y and Z coordinates of the joints of the hands, elbows and shoulders like we can observe in the green points on the Figure 1. The points are recorded obeying to the coordinate system below [7].

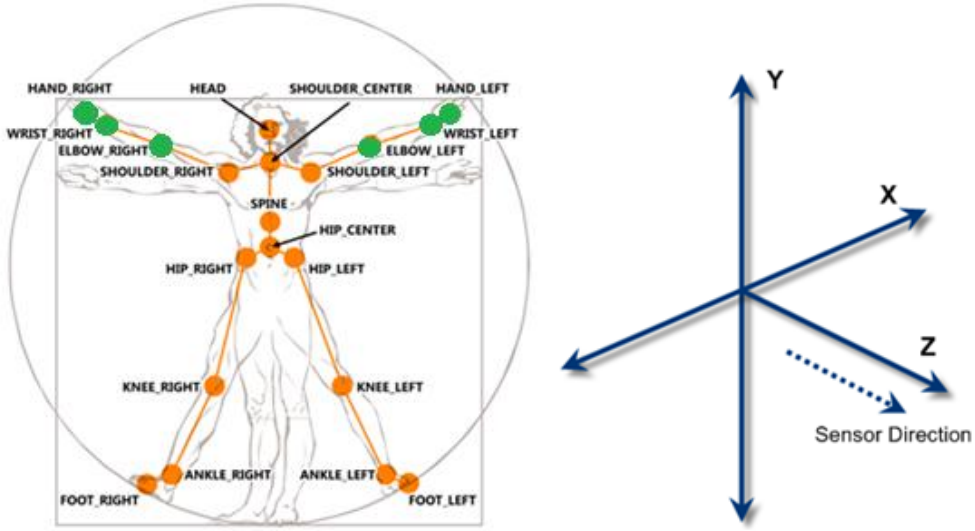


Figure 1: Points tracked by the Kinect sensor and Kinect sensor coordinate system [7]

It is needed to optimize the information in order improve the data processment and deliver an online recognition system. To achieve this we use a smaller set of joints than the Kinect sensor can offer. We need to use reference points, namely, the center between the shoulders to pre-process the data and to be able to simplify the recognition of the gestures that in most of the times involve the upper body members.

There is an importance to the pre-processing of the data by normalizing and centering because we need to eliminate the effect of the distance of the subject to the camera.

The data is retrieved at a rate of 31 frames per second and each gesture takes a minimum of 6 frames to be classified since the algorithms used are not sensitive to the duration of the gesture, but there must be a minimum number of frames to be classified.

We must consider the possibilities of the rotation movements in front of the sensor, since the gesture seems different to the point of view of the camera if a person is directly in front or sideways. In order to solve this issue, it is applied a 2D rotation to all the points recorded related to the Z and X coordinates, this way the line between the two shoulders is transformed in the X axis. This angle is calculated between the X axis and the line between the two shoulders and the rotation operation performed is explicit below.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Equation 1: 2D rotation

After this process there is a need to save the gesture data in XML. This way, the data saved is easily red by any human being and accessed throw the application. This way, we save each gesture with all the frames, on each frame we save each articulation and on each articulation we save the X, Y and Z values corresponding to the coordinate system.

2.2. Gesture detection

2.2.1. Dynamic Time Warping algorithm

In what concerns to the detection of gestures, first we need to find some algorithms and test them. In this paper we are testing the Dynamic Time Warping and the Hidden Markov Models [8]. These two algorithms are selected because of its nature of dynamic programming and they are mostly used in time series pattern recognition. This way we don't need to worry about how much time one person takes to do the gesture despite the type of gesture that it's made or its complexity.

The Dynamic Time Warping algorithm measures similarities between patterns that can change in terms of duration, so there is an elastic characteristic to this algorithm that may be very useful [8].

The main mechanics of this is explained below.

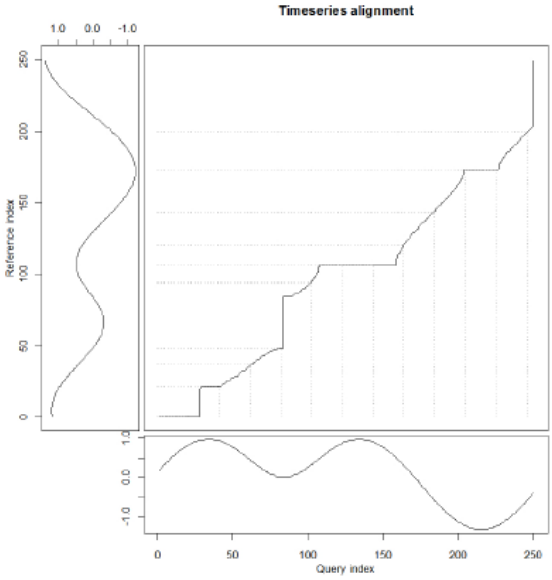


Figure 2: DTW example [9]

This algorithm works its way through the best line that allows the two time series alignment. This line is found by the measure of the Euclidean distance between each point of the reference sequence and the query sequence [9]. These distances are added and the best path is found by the least cost path or the path with the least distance. This is the main goal of the algorithm process but there are four rules to be respected:

- The first and last point of the path must coincide with the first and last point of the two sequences.
- The natural order of the time continuum must not be violated, in other words the path must never go backwards in the time series [10].
- The size of the horizontal and vertical steps must be defined by the user and respected.
- Each point of the query index must have at least one correspondence. No jumps in time are allowed.

In the graphs below we can see the first rule violated on b) the second rule violated on c), the third rule violated on d) and no rule violated on a) [9].

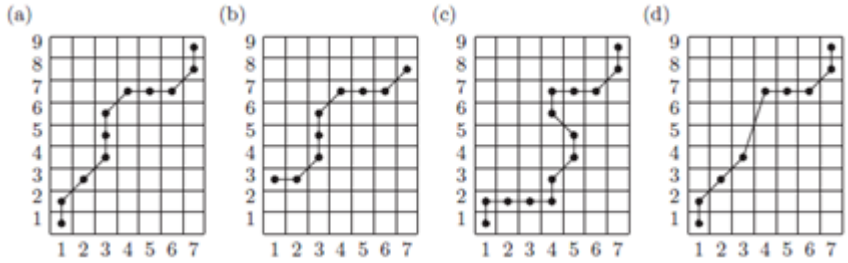


Figure 3: Dynamic Time Warping rules [12]

The mechanics of the algorithm is explicit is the schema below. Each time there is a comparison between two points, one of the original time series (recorded gesture) and the time series we want to compare (gesture performed by the person in front of the camera) there are three options: go diagonal, go up or go down.

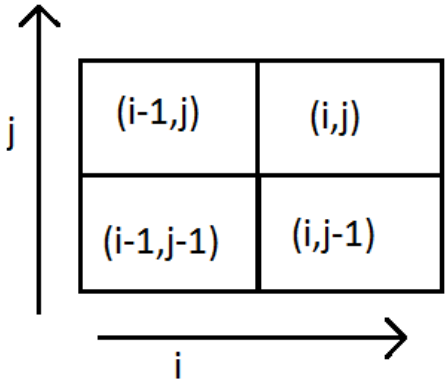


Figure 4: Dynamic Time Warping mechanics

In order to detect if no gesture is being made there is a threshold to define if the comparison is valid or invalid. So we can be doing nothing and there is no gesture recognized. This algorithm will be comparing the gesture made in front of the camera with a list of recorded gestures.

In a practical matter, in order to get the most interesting results with this algorithm, the gestures must be well recorded, because this way the recognition is made more effectively. Firstly we calculate how much time a person takes to perform an action, secondly we must record the same gesture several times in order to get some generalization. The amount of time we think it is the most commonly used by a person to perform an action is about one second.

The comparison with several sequences of the same gesture is inevitable. If there are many variations of the same gesture and to avoid the misclassification there must be a system to compare equal gestures. So we also implemented an alternative system that compares the gesture we want to classify with all the gestures but instead of choosing the most probable gesture like in the standard DTW, we make the mean probability between the gestures that are equal and choose from there. In order to find the best approach we tested both solutions.

2.2.2. Hidden Markov Models algorithm

The Hidden Markov Models algorithm are often used on voice recognition problems which are time series sequences. This algorithm works through states and outputs and each state depends on the previous one and each exit depends only on the correspondent state. In this type of algorithm the probabilities of transition between states ($a_{11}, a_{12}, a_{13\ldots}$) and the probability of a state to produce an output y ($b_{11}, b_{12}, b_{13\ldots}$) are visible. However the state X is not visible.

Because in this algorithm we only can observe the outputs, there is a need to determine all the variables only looking at these. The dynamics of the Hidden Markov Models are explicit in the figure below.

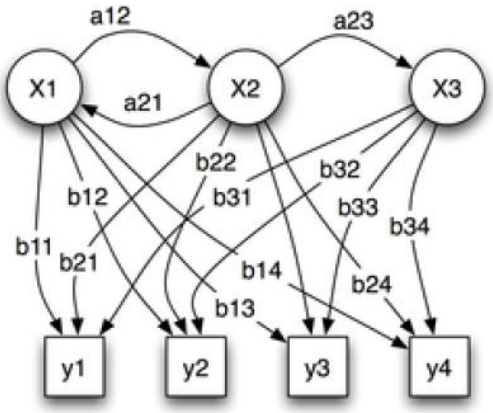


Figure 5: Hidden Markov Models schema [11]

As we can observe, all the states can produce all the outputs and each state only depends on the state before [12] [13].

There are three main problems that must be solved in this process:

- Determine the probability of a determinate state to produce an output.
- Determine the probability of a state transit to another.
- Determine the probability of existing a determinate sequence of states.

On this subject we trained two kinds of learning algorithms: the Viterbi algorithm [14] and the Baum-Welch algorithm [15]. The main reason of existence of these algorithms is to solve these issues. On one hand, on the Baum-Welch algorithm, the global transition and emission probabilities of the hidden states are calculated with each iteration [16]. On the other hand, with the Viterbi learning algorithm only the present iteration contributes to the present calculation of the probabilities reducing the computational complexity [17].

To apply this to the issue in matter we train each group of gestures as a Markov model in order to calculate all the variables. Each output represents one part of the movement and, consequently the sequence of hidden states affects directly this sequence.

After training all the models or in other words after training all the gestures, it is possible to determine which one is being performed in front of the camera.

3. Results

The results were obtained through a series of repetitions of the gestures made by 19 people. Each algorithm was tested using the method leave one out. The gestures performed, for now, are very simple in order to see the performance of the several algorithms.

The measures of specificity and sensibility are the most important on algorithm performance [18]. The sensibility reflects the fraction of times that a gesture was performed and well classified. The specificity is the fraction of times that no gesture was performed and was classified as such.

Each set of data is obtained with a more restrictive setting of the respective algorithm. As we can conclude from the section above, the HMM algorithm variations have a certain set of settings and the DTW algorithm variations have other set of settings (respecting the rules). For comparison, the DTW algorithms (first two columns) use the same settings and the same happens on the last two columns in the HMM algorithms test.

Table 1: Sensibility and Specificity of the various algorithms

		DTW	DTWmean	HMM Baum-Welch	HMM Viterbi
Set 1	Specificity	26,32%	15,79%	100%	100%
	Sensitivity	82,89%	78,95%	36,18%	68,42%
Set 2	Specificity	100%	100%	100%	100%
	Sensitivity	63,82%	50,13%	32,89%	63,16%
Set 3	Specificity	100%	100%	100%	100%
	Sensitivity	30,26%	20,46%	26,97%	57,83%

In the table above we have a measure of performance of the different algorithms used in this process. The parameters used on each set are in the table below.

Table 2: parameter used on each Set of the experimental step

	DTW	HMM
Set 1	5 maximum step 20 maximum Euclidean distance 6 threshold sensitivity	5 hidden states 0.1 threshold sensitivity
Set 2	3 maximum step 12 maximum Euclidean distance 3 threshold sensitivity	10 hidden states 0.1 threshold sensitivity
Set 3	2 maximum step 10 maximum Euclidean distance 2 threshold sensitivity	10 hidden states 0.05 threshold sensitivity

As we can see in the Table 2 we tested different numbers of hidden states and threshold sensitivities in order to find the best solution for the HMM algorithm and different numbers of maximum steps, Euclidean distance and threshold sensitivities in order to find the best combination for the DTW

algorithm. In the process of choosing the set of parameters to test the algorithms it was essential the trial and error phase. After trying several combinations of parameters these were the most promising.

Table 3 : HMM Viterbi ; Set1

Performed / Recognized	Raise both arms	Raise right arm	Raise left arm	Applaud	Sit	Raise	Swipe Right	Swipe Left	unknown
Raise both arms	100,00%	0,00%	0,00%	0,00%	5,26%	0,00%	0,00%	0,00%	0,00%
Raise right arm	0,00%	94,74%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Raise left arm	0,00%	0,00%	47,37%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
Applaud	0,00%	0,00%	0,00%	21,05%	0,00%	0,00%	0,00%	0,00%	0,00%
Sit	0,00%	0,00%	0,00%	0,00%	73,68%	31,58%	0,00%	0,00%	0,00%
Raise	0,00%	0,00%	0,00%	0,00%	0,00%	26,32%	0,00%	0,00%	0,00%
Swipe Right	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	94,74%	0,00%	0,00%
Swipe Left	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	89,47%	0,00%
unknown	0,00%	5,26%	52,63%	78,95%	21,05%	42,11%	5,26%	10,53%	100,00%

In the Table 3, we can see a sample of the results taken. This corresponds to the most interesting table we could get. The "unknown" line and column correspond to the situations when no gesture was performed and no gesture was recognized, in other word it corresponds to the specificity.

4. Discussion

As we can see in the last section, on each test there is an important component referring to the threshold applied. There is some importance on this subject because if the threshold is too permissive, the amount of data accumulated in the database is too much (false positives) and if the threshold is set too restrictive then we won't recognise the gestures. This is directly correlated to the measure of specificity. As such, the first set of the DTW and *DTWmean* algorithms are not interesting.

The rest of the tests resulted on 100% specificity which is good. Between these results, the best is the first set of the HMM Viterbi algorithm due to the 68,43% of mean sensibility.

After looking at the Table , there is a clear difference in the sensibility of the several gestures. This may be due to the more or less perfect recording of those. Most of the times was asked to the participants to act in the most natural way because, in the "real world" with people with ASD's, there isn't a specific way to perform gestures.

Another conclusion taken after analyzing human gestures is that every time I ask to raise both arms, there seems to be a slightly different way to perform the action. The first instinct in how to perform this sometimes leads people to raise the arms laterally or frontally.

5. Conclusions

A completely autonomous way of recording and analyse the patient's behaviours is the main reason of this study, but the results are not perfect and the surveillance will not give the most accurate results. However it is possible to implement a house with this surveillance system. In this house would be living a

caregiver with patients with ASD and through this system there would be an alarm mechanism to detect abnormal and stereotyped behaviours.

With the information retrieved from the intelligent system and the caregiver there is a chance to enhance the performance. In conclusion, the system would be one more mechanist that the caregiver would have to determine the evolution of the patients through the therapies sessions and the everyday life challenges.

For the future work there will be tests in patients with ASD for more accuracy of the results in the real life.

6. Acknowledgments

I would like to thank the ISA – Intelligent Sensing Anywhere, S.A. company for providing with a good environment and facilities to complete this project. Also I want to thank to the people involved in the HomeTech project that always supported me.

References

[1] D. Farrugia, "Exploring stigma: medical knowledge and the stigmatisation of parents of children diagnosed with autism spectrum disorder," *Sociology of Health & Illness,* vol. 31, p. 1011–1027, 2009.

[2] A. S. Org., "Autism Speaks, It's time to listen," 2013. [Online]. Available: http://www.autismspeaks.org/. [Accessed 10 1 2013].

[3] M. Rutter, "Autism and Pervasive Developmental Disorders: Concepts and Diagnostic Issues," *Journal of Autism and Developmental Disorders,* vol. 17 No.2, 1987.

[4] C. K. Gurkan and R. J. Hagerman, "Targeted treatments in autism and fragile X syndrome," *Research in Autism Spectrum Disorders,* vol. 6, p. 1311–1320, 2012.

[5] D. Robins, D. Fein and M. Barton, "Follow-up Interview for the Modified Checklist for Autism in Toddlers".

[6] "Autism diagnosis and screening: Factors to consider in differential diagnosis," *Research in Autism Spectrum Disorders,* vol. 6, p. 19–24, 2012.

[7] microsoft, "msdn," [Online]. Available: http://msdn.microsoft.com/en-us/library/hh438998.aspx. [Accessed 15 03 2013].

[8] P. Senin, "Dynamic Time Warping Algorithm Review," Information and Computer Science Department, University of Hawaii at Manoa, Honolulu, USA, 2008.

[9] T. Giorgino, "Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package," *Journal of Statistical Software,* vol. 31, no. 7, 2009.

[10] D. J. G. Harris, "Isolated word, Speech recognition using Dynamic Time Warping towards smart appliances.," [Online]. Available: http://www.cnel.ufl.edu/~kkale/dtw.html. [Accessed 4 12 2012].

[11] C. Fang, "From Dynamic Time Warping (DTW) to Hidden Markov Model (HMM)," *Final project report for ECE742 Stochastic Decision,* 2009.

[12] V. I. Pavlovic, R. Sharma and T. S. Huang, "Visual Interpretation of Hand Gestures for Human-Computer Interaction: A Review," *IEEE Transactions On Pattern Analysis And Machine Intelligence* , vol. 19, 1997.

[13] J. C. Hall, "Creative Distraction," [Online]. Available: http://www.creativedistraction.com/demos/gesture-recognition-kinect-with-hidden-markov-models-hmms/. [Accessed 3 12 2012].

[14] J. G. David Forney, "The Viterbi Algorithm," *Proceedings of the IEEE,* vol. 61, pp. 268-279, 1973.

[15] E. G. A. M. R. S. Wael Khreich, "On the memory complexity of the forward-backward algorithm," *Pattern Recognition Letters,* vol. 31, pp. 91-99, 2010.

[16] L. Rodríguez and I. Torres, "Comparative study of the Baum-Welch and Viterbi training algorithms applied to read an spontaneous speach recognition," *Pattern Recognition and Image Analysis Lecture Notes in Computer Science*, vol. 2652, pp. 847-857, 2003.

[17] C. d. Souza, "Code Project," [Online]. Available: http://www.codeproject.com/Articles/541428/Sequence-Classifiers-in-Csharp-Part-I-Hidden-Marko. [Accessed 15 04 2013].

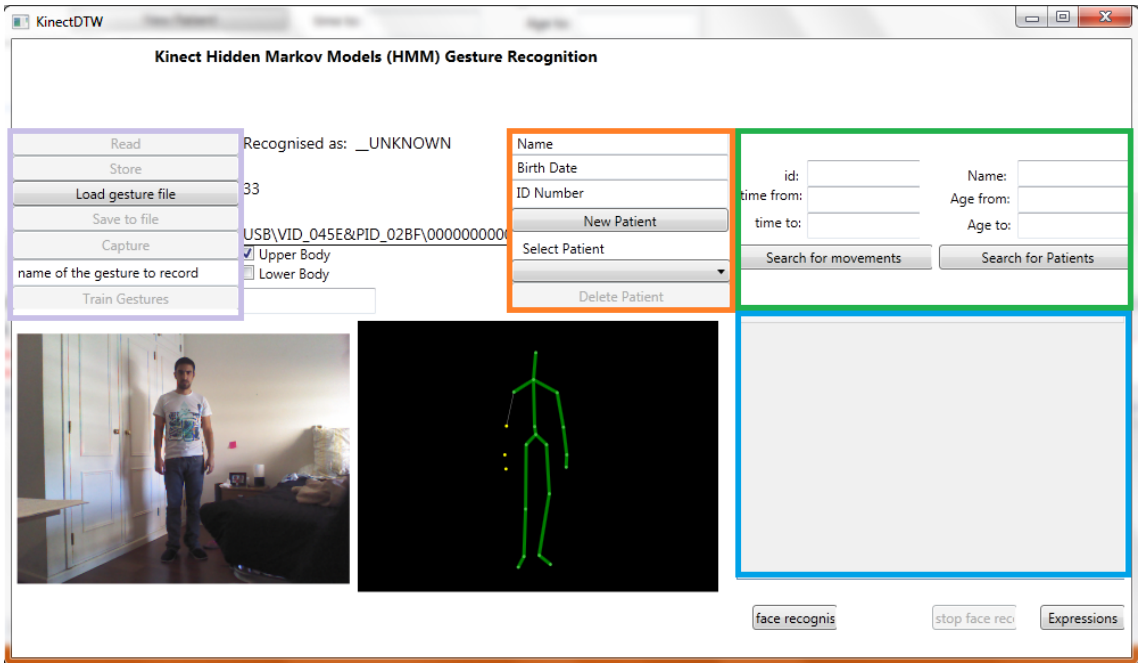
[18] A. G. Lalkhen and A. McCluskey, "Clinical tests: sensitivity and specificity," *BJA: CEACCP Clinical Education in Anaesthesia, Critical Care and Pain*, vol. 8, no. 6, pp. 221-223, 2008.

[19] "Centers for Disease Control and Prevention," 29 03 2012. [Online]. Available: http://www.cdc.gov/ncbddd/autism/data.html. [Accessed 11 12 2012].

[20] M. Muller, "Information Retrieval for Music and Motion," Springer Berlin Heidelberg New York, 2007.

Anexo B

Na aplicação desenvolvida é aplicado o algoritmo *Hidden Markov Models* com treino *Viterbi*.



No quadrado laranja existem todas as condições para inserir um novo paciente ou selecionar manualmente um existente. No quadrado verde estão as opções de pesquisar na base de dados os movimentos e os pacientes disponíveis; estas informações são disponibilizadas no quadrado azul. No quadrado roxo estão os botões referentes às operações de treino do algoritmo, reconhecimento de gestos em tempo real, gravação de um gesto e guardar um gesto.

Nas figuras na parte inferior da janela estão as imagens RGB e do esqueleto disponibilizadas pelo *Kinect* de forma a monitorizar o desempenho atual do sensor.

Entre os quadrados roxo e laranja é possível escolher que parte do corpo se quer registar (membros posteriores ou anteriores) e a tarefa atual da aplicação.

Inferiormente à área azul pode-se observar os botões que nos permitem reconhecer o paciente pela face e detetar expressões faciais.