



**Inês Gomes Sá Neiva**

Licenciatura em Ciências de Engenharia Biomédica

## **Desenvolvimento de um tradutor de Língua Gestual Portuguesa**

Dissertação para obtenção do Grau de Mestre em  
Engenharia Biomédica

Orientador: Hugo Alexandre Ferreira, Professor Doutor,  
Faculdade de Ciências da Universidade de Lisboa

Co-orientadora: Maria Helena Caria, Professora Doutora,  
Departamento de Ciências Biomédicas,  
Instituto Politécnico de Setubal

Júri:

Presidente: Prof. Doutora Carla Maria Quintão Pereira

Arguente: Prof. Doutor Daniel Jorge Viegas Gonçalves

Vogal: Prof. Doutor Hugo Alexandre Ferreira



**Dezembro, 2014**





**Inês Gomes Sá Neiva**

Licenciatura em Ciências de Engenharia Biomédica

## **Desenvolvimento de um tradutor de Língua Gestual Portuguesa**

Dissertação para obtenção do Grau de Mestre em  
Engenharia Biomédica

Dissertação apresentada na Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa para obtenção do Grau de Mestre em Engenharia Biomédica.

A presente dissertação foi desenvolvida em colaboração com o Instituto de Biofísica e Engenharia Biomédica (IBEB/FCUL).

Orientador: Professor Doutor Hugo Alexandre Ferreira

Co-orientadora: Professora Doutora Maria Helena Caria



Dezembro, 2014



## **Desenvolvimento de um Tradutor de Língua Gestual Portuguesa**

Copyright © 2014 – Todos os direitos reservados. Inês Gomes Sá Neiva, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa.

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.



*In the beginner's mind, there are many possibilities,*

*But in the expert's mind, there are few.*

*- Shunryu Suzuki*





# Agradecimentos

Começo por agradecer aos anfitriões desta dissertação, o Instituto de Biofísica e Engenharia Biomédica por me terem recebido, ajudado e acarinhado neste tempo que lá estive.

Ao Professor Hugo Ferreira, meu orientador, um muito obrigado por me mostrar o melhor caminho para a conclusão desta dissertação e me introduzir novos temas. Agradeço a confiança e a ajuda dada fora do âmbito desta dissertação.

À Professora Maria Helena Caria, também agradeço pela ajuda dada para a obtenção de apoios para o projeto e futuro desenvolvimento.

Obrigado à D.<sup>a</sup> Beatriz e aos restantes membros do IBEB, pela simpatia demonstrada.

Agradeço aos meus pais e aos pais do meu namorado pelo apoio incondicional.

Ao Valadas, ao “O Sousa” e à Ana Morgado agradeço o companheirismo e amizade.

Ao Ricardo, agradeço a paciência e compreensão nos dias difíceis, mas agradeço principalmente o amor, sempre presente.

**FCT** Fundação para a Ciência e a Tecnologia  
MINISTÉRIO DA EDUCAÇÃO E CIÊNCIA

Este trabalho foi financiado pela Fundação para a Ciência e Tecnologia (FCT) e Ministério da Ciência e Educação (MCE) Portugal (PIDDAC) pois estava integrado no projeto PEst-OE/SAU/UI0645/2014.



O presente projeto que aqui se apresenta ganhou o concurso “Passaporte para o Empreendedorismo”, tendo a partir do dia 1 de Julho, apoio por parte das seguintes entidades:





# Resumo

A Língua Gestual é um meio de comunicação entre surdos e entre surdos e técnicos especialistas que não requer o uso da expressão/comunicação oral. Contudo, a interação de surdos com outras pessoas não surdas ou não especialistas é dificultada pela ausência de uma linguagem comum. Em particular, no contexto do dia-a-dia a utilização de serviços que envolvam uma interação mais complexa pode ser impossível sem um intérprete de Língua Gestual.

Com o intuito de resolver esta barreira na comunicação, foi vista uma oportunidade de desenvolvimento de uma aplicação que, em conjunto com o Leap Motion, deteta, grava e reconhece a fluência do movimento dos gestos na base da Língua Gestual.

Neste trabalho foram estudadas 4 diferentes metodologias com o intuito de iniciar o desenvolvimento do primeiro tradutor pessoal de Língua Gestual Portuguesa (LGP) usando a tecnologia recente e inovadora do Leap Motion. O trabalho foi desenvolvido sobre a *framework* Unity3D por motivos de desenvolvimento futuro multiplataforma, o desenvolvimento da interface de visualização gráfica em tempo real e a utilização de som para a tradução.

Foram obtidos bons resultados em 3 das 4 metodologias, sendo elas o LeapTrainerUI, e as duas aplicações de Reconhecimento de Numeração cardinal em LGP até 5, conseguindo uma correta classificação da numeração. A última metodologia, Classificador de 1 Gesto, com um grau de complexidade maior, foi a que obteve resultados menos satisfatórios.

Embora os resultados obtidos sejam preliminares, este trabalho abre as perspetivas ao desenvolvimento de um intérprete pessoal de LGP. Apesar desta língua ter uma complexidade superior à simples deteção de movimento das mãos e dos dedos, num futuro próximo, conseguir-se-á que pessoas surdas tenham consigo este dispositivo e consigam transmitir de forma verbalizada a sua intenção a pessoas que não entendam LGP.

**Palavras-chave:** Tradutor, LGP, Intérprete, Leap Motion, Unity 3D, Reconhecimento Gestual



# Abstract

Sign Language is a tool of communication between deaf and between deaf and technical experts, which do not require the use of oral expression/communication. However, the interaction with other non-deaf people is hampered by the lack of a common language. Particularly, in the context of day-to-day use of services involving a more complex interaction, communication may be impossible without a Sign Language interpreter.

In order to solve this communication barrier, an opportunity was seen to develop an application, using the Leap Motion device that detects, recognizes and records gestures in the base of Sign Language.

In this work, four different methodologies were studied in order to start the development of the first translator device of Portuguese Sign Language (PSL) using the innovative the Leap Motion device. The work was done using Unity3D *framework* for reasons of future multiplatform implementation, the development of graphical real-time visualization and the use of sound for the translation interface tested.

Good results were achieved in three of the four methodologies, which were the LeapTrainerUI, and two applications of Recognition cardinal Numbering in PSL until 5, getting a correct classification of the numbering. The last methodology, One Gesture Classifier, which has a greater degree of complexity, was the one with less satisfactory results.

Although the results are preliminary, this work opens the prospects for developing a personal interpreter of PSL. This language, inspect of having a higher complexity than simple detection of movement of the hands and fingers, it is expected that in the near future, will be achieved that deaf people can have this device and be able to communicate their intentions to people who do not understand PSL.

**Keywords:** Translator, PSL, Interpreter, Leap Motion, Unity 3D, Gesture Recognition





# Índice de Conteúdos

Agradecimentos.....	vii
Resumo .....	xi
Abstract .....	xiii
Índice de Conteúdos.....	xv
Índice de Figuras.....	xix
Índice de Tabelas .....	xxiii
Siglas e Acrónimos .....	xxv
1. Introdução .....	1
1.1. Motivação .....	1
1.2. Estrutura.....	2
2. Surdez.....	5
2.1. Definição e Classificação .....	5
3. Língua Gestual Portuguesa .....	7
3.1. História da Língua Gestual .....	7
3.2. Sistema SignWriting .....	9
3.3. Morfologia da LGP .....	10
3.3.1.1. Configuração da mão .....	10
3.3.1.2. Orientação Manual.....	11
3.3.1.3. Local de articulação .....	11
3.3.1.4. Movimento .....	12
3.3.1.5. Expressão facial e corporal.....	13
3.3.1.6. Sintaxe .....	13
3.4. Reconhecimento de Língua Gestual .....	14
4. Estado de Arte .....	15
4.1. Reconhecimento do Gesto Humano.....	15

4.2.	Tecnologias de Reconhecimento Gestual.....	17
4.2.1.1.	Enable Talk.....	21
4.2.1.2.	Microsoft Kinect.....	22
4.2.1.3.	Leap Motion.....	28
5.	Metodologia.....	33
5.1.	Raw Data.....	34
5.1.1.1.	LeapTrainer UI .....	34
5.2.	Reconhecimento de Numeração.....	40
5.2.1.1.	Interface de Utilização .....	41
5.2.1.2.	Funcionamento da Aplicação .....	44
5.3.	Reconhecimento de Numeração com HMM.....	45
5.3.1.1.	Accord.NET <i>Framework</i> .....	45
5.3.1.2.	Modelo para Reconhecimento de Gestos .....	45
5.3.1.3.	Modelo de Hidden Markov .....	47
5.3.1.4.	Interface de Utilização .....	49
5.3.1.5.	Funcionamento da Aplicação .....	51
5.4.	Classificador de 1 Gesto .....	52
5.4.1.1.	Interface de Utilização .....	52
5.4.1.2.	Funcionamento da Aplicação .....	55
6.	Resultados .....	57
6.1.	LeapTrainer UI .....	58
6.2.	Reconhecimento de Numeração.....	59
6.3.	Reconhecimento de Numeração com HMM.....	61
6.4.	Classificador de 1 Gesto .....	63
7.	Discussão dos Resultados .....	65
7.1.	LeapTrainerUI .....	65
7.2.	Reconhecimento de Numeração com e sem HMM.....	66

7.3.	Classificador de 1 Gesto.....	68
8.	Conclusão .....	69
8.1.	Conclusões.....	69
8.2.	Limitações .....	70
8.3.	Perspetivas Futuras .....	70
9.	Bibliografia .....	73



# Índice de Figuras

Figura 1.1: Estrutura da Dissertação .....	3
Figura 3.1: Exemplos de Gestos e os seus símbolos em SignWriting [76].....	9
Figura 3.2: Alfabeto gestual da LGP. [77] .....	10
Figura 3.3: Alfabeto gestual observado da posição do emissor e do recetor. [12] ....	11
Figura 3.4 Pontos de articulação na face, pescoço e cabeça. [12] .....	12
Figura 3.5: Pontos de articulação no corpo. [12] .....	12
Figura 3.6: Expressões faciais. Adaptado de [12] .....	13
Figura 4.1: Graus de liberdades da mão e punho. [15] .....	15
Figura 4.2: Classificação de Gestos por Aigner et al [16] .....	16
Figura 4.3: Esquema das técnicas de representação de gestos [16].....	17
Figura 4.4: 1 – Myo [19], 2 - Enable Talk glove [24], 3 – CyberGlove [18], 4 – Power Glove [23] .....	18
Figura 4.5: Comando da Wii [27].....	19
Figura 4.6: <i>Smartphone touchless</i> com 4 antenas de radiofrequência [29] .....	19
Figura 4.7: 1 – Kinect [33], 2 – Lazeeeye [34], 3 – Leap Motion [36], 4 - Intel RealSense [78].....	20
Figura 4.8: Comando Move da PlayStation [37] .....	20
Figura 4.9: Enable Talk - Especificações das luvas. [24] .....	21
Figura 4.10: Detalhes Técnicos da Luva Enable Talk. [41] .....	22
Figura 4.11: Componentes Técnicos da Kinect [33] .....	23
Figura 4.12: <i>Raw Data</i> fornecido pela Kinect. [42].....	24
Figura 4.13: <i>Skeleton Tracking</i> – informação sobre articulações. [79] .....	24
Figura 4.14: Face Tracking executado pela Kinect. [80] .....	25
Figura 4.15: Informação disponibilizada pela API da Kinect. [81] .....	25
Figura 4.16: Interface do protótipo Kinect Sign Language Translator. [82].....	26

Figura 4.17: Diagrama de blocos do tratamento de dados para reconhecimento gestual. [43] .....	27
Figura 4.18: Disposição das coordenadas espaciais e campo de visão do dispositivo Leap Motion. [36] .....	28
Figura 4.19: Informação extraída pelo dispositivo. [36] .....	28
Figura 4.20: Gesto <i>Circle</i> detetado pelo dispositivo. [83] .....	29
Figura 4.21: Gesto <i>Swipe</i> implementado na API do Leap Motion. [83] .....	29
Figura 4.22: Gesto <i>Key Tap</i> implementado na API do dispositivo. [83] .....	29
Figura 4.23: Gesto <i>Screen Tap</i> implementado na API do Leap Motion [83] .....	30
Figura 4.24: Leap Motion – Normal funcionamento do aparelho da V2 com <i>skeletal tracking</i> [84] .....	30
Figura 4.25: Sistema idealizado pelas Designers. [49] .....	31
Figura 4.26: MotionSavvy – Componentes de <i>hardware</i> e <i>software</i> . [51] .....	32
Figura 5.1: Interface Inicial do LeapTrainer.js .....	35
Figura 5.2: Treino Concluído. ....	35
Figura 5.3: Menu de Opções. Adaptado de [55] .....	36
Figura 5.4: GUI usada para a aplicação Reconhecimento de numeração até 5. ....	41
Figura 5.5: Interface com a mão direita representada. Esta interface mostra em tempo real os movimentos capturados pelo dispositivo. ....	42
Figura 5.6: Exemplo de postura da mão no início da aplicação. ....	42
Figura 5.7: Exemplo da conclusão da aplicação. Neste caso, o número 4 foi exibido com a mão esquerda. ....	43
Figura 5.8: Resultado da aplicação quando um punho é exibido. ....	43
Figura 5.9: Resultado da aplicação quando não existe uma mão no campo de visão do aparelho. ....	43
Figura 5.10: Esquema representativo do modelo da decisão usado na aplicação. ....	49
Figura 5.11: Interface da aplicação Reconhecer Numeração até 5. ....	50
Figura 5.12: Interface de apresentação da aplicação. ....	52

Figura 5.13: Interface apresentada após clicar “Gravar Gesto”.....	53
Figura 5.14: Reconhecimento de gestos (início). ....	53
Figura 5.15: Exemplo do reconhecimento positivo de um gesto.....	54





# Índice de Tabelas

Tabela 5.1: Esquema demonstrativo do funcionamento da aplicação. ....	44
Tabela 5.2: Resumo da literatura sobre Identificação de Gestos. Adaptado de [15], [62], [63].....	46
Tabela 5.3: Excerto da implementação do Classificador HMM usando a API da framework Accord.NET.....	50
Tabela 5.4: Esquema demonstrativo do funcionamento da aplicação. ....	51
Tabela 5.5: Esquema demonstrativo do funcionamento da aplicação Classificador de 1 Gesto. ....	55
Tabela 6.1: Matriz – Confusão para a aplicação LeapTrainerUI.....	58
Tabela 6.2: Análise do classificador do LeapTrainerUI.....	59
Tabela 6.3: Matriz – Confusão para a aplicação Reconhecimento de Numeração...	60
Tabela 6.4: Análise do classificador da aplicação Reconhecimento de Numeração .	60
Tabela 6.5: Matriz – Confusão para a aplicação Reconhecimento de Numeração – Accord <i>Framework</i> .....	62
Tabela 6.6: Análise do classificador da aplicação Reconhecimento de Numeração – Accord <i>Framework</i> .....	62
Tabela 6.7: Resultados obtidos da aplicação Classificador 1 Gesto .....	63



# Siglas e Acrónimos

API – *Application Programing Interface*

dB – Décibel

GUI – *Graphical User Interface*

HMM – *Hidden Markov Models*

Hz – Hertz

LGP – Língua Gestual Portuguesa

SDK – *Software Development Kit*



# 1. Introdução

A surdez de etiologia genética, congénita ou adquirida – profunda ou severa a moderada, resulta geralmente em dificuldades na expressão oral. Neste contexto, a Língua Gestual surge como um meio de comunicação entre surdos e entre surdos e técnicos especialistas que não requer o uso da expressão oral. Contudo, a interação de surdos com outras pessoas não surdas ou não especialistas é dificultada pela ausência de uma linguagem comum.

Com o intuito de resolver esta barreira comunicacional, foi identificada a oportunidade de iniciar o desenvolvimento de uma aplicação associada a um dispositivo móvel que deteta, grava e reconhece a fluência do movimento dos gestos e os associa a um som, dando assim voz a quem não a tem.

Apesar da Língua Gestual Portuguesa ter uma complexidade superior à simples deteção de movimento das mãos e dos dedos, neste trabalho será demonstrado com este que é possível, num futuro próximo, as pessoas surdas terem consigo um dispositivo associado a um smartphone e conseguirem transmitir a sua intenção a pessoas que não tenham fluência na Língua Gestual Portuguesa.

## 1.1. Motivação

Para as pessoas surdas, situações rotineiras como uma ida ao banco, ao serviço da Segurança Social, uma consulta médica ou mesmo uma entrevista de emprego podem ser um obstáculo para estas mesmas. Frequentemente, estas só se conseguem fazer entender com o apoio de um intérprete de Língua Gestual mas à custa da sua autonomia e

conveniência. Esta dependência também é um entrave à integração em sociedade. Por conseguinte, irá dar-se os primeiros passos para desenvolver um dispositivo móvel que será usado como um intérprete pessoal e portátil de Língua Gestual, traduzindo gestos em palavras e sons.

Pretende-se assim “dar voz” às pessoas surdas, permitindo-lhes comunicar de forma autónoma e sem barreiras com toda a gente, possibilitando-lhes o pleno exercício da cidadania.

Apesar de haver uma variedade enorme de dispositivos que podem ser usados para traduzir gestos, como serão discutidos nas secções seguintes, muitos destes não são discretos, móveis nem portáteis, colocando automaticamente um rótulo de pessoa diferente ao seu utilizador. De forma a contornar esta barreira, quis-se tirar partido de um dispositivo recente e prometedo no reconhecimento gestual, o Leap Motion. É visto como uma oportunidade inovadora de iniciar um projeto pioneiro na tradução da Língua Gestual Portuguesa com um dispositivo discreto, que virá ajudar as futuras gerações surdas a serem vistas como um igual na sociedade.

## 1.2. Estrutura

O Instituto de Biofísica e Engenharia Biomédica (IBEB) situado na Faculdade de Ciências da Universidade de Lisboa foi anfitrião deste trabalho, em parceria com a Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa. A estrutura seguida é a da Figura 1.1.

A dissertação está organizada em 7 capítulos de conteúdo e um último de bibliografia relevante para a realização da dissertação. Nos 3 primeiros capítulos encontra-se a descrição da surdez, língua gestual portuguesa e o estado da arte do reconhecimento gestual. Nos 3 capítulos seguintes estão descritas as abordagens concretizadas para iniciar o desenvolvimento do primeiro tradutor pessoal de LGP, resultados obtidos e a sua discussão. Para finalizar são debatidas as perspetivas futuras, as limitações deste trabalho e as conclusões.

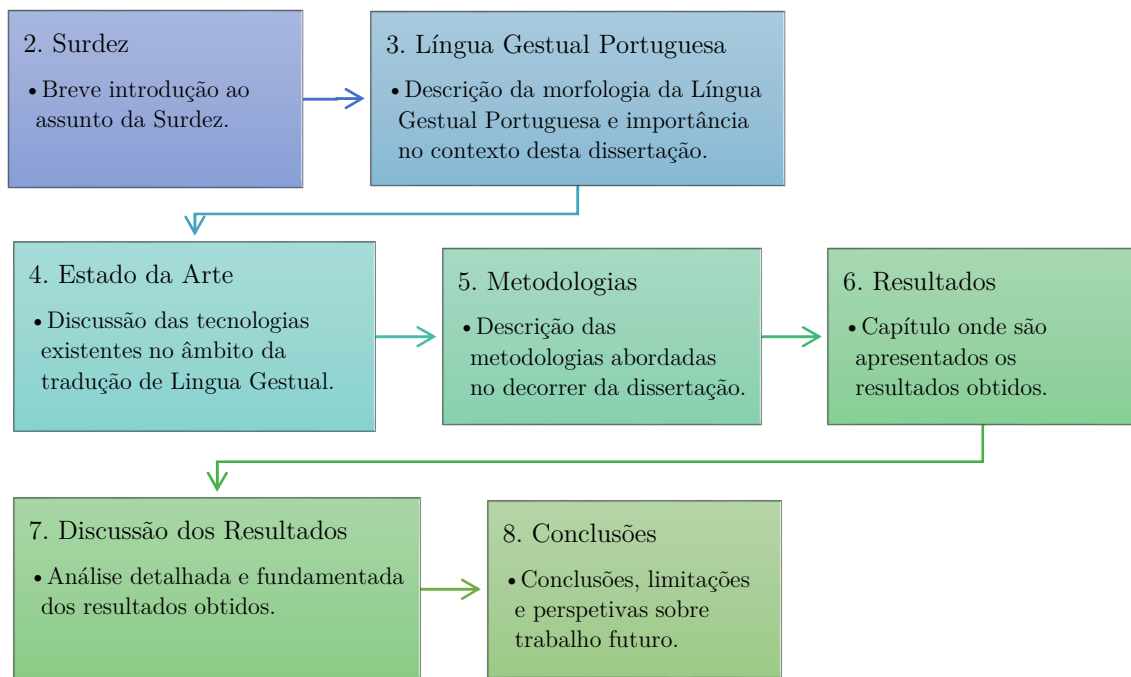


Figura 1.1: Estrutura da Dissertação





## 2. Surdez

A audição é a capacidade que o ouvido possui de captar as vibrações que compõem o som e transformá-las em impulsos nervosos que o cérebro vai decodificar e atribuir significado, fazendo corresponder a cada frequência captada um som diferente. A pessoa surda é aquela que não consegue ouvir e interpretar som, seja o som proveniente da fala ou meio envolvente.

Um som pode ser definido pelo seu ciclo (alternância entre compressão e expansão), frequência (números de ciclos por segundo), amplitude/intensidade, timbre (característica dependente da fonte sonora) e a velocidade (dependente do meio) [1].

A perda da percepção do som pode ser genética (o indivíduo nasce surdo, habitualmente) ou ser um fator adquirido com a idade ou resultante de algum fator ambiental (como um trauma). É importante assim fazer uma distinção entre surdez e perda de audição. A perda de audição, entende-se neste trabalho como a deterioração da função auditiva e a surdez como uma incapacidade profunda na percepção do som, o que leva a inexistência de função auditiva.

### 2.1. Definição e Classificação

A perda de audição sensorial (denomina-se sensorial quando afeta o ouvido interno) pode ser hereditária ou causada por trauma acústico, infecções virais ou por doenças como o síndrome de Ménière. Já a privação da função auditiva pela via neuronal (denomina-se neuronal quando afeta o nervo auditivo ou as suas vias) pode ser causada por tumores cerebrais que também danificam os nervos circundantes ou igualmente por infecções, várias

perturbações cerebrais e nervosas, como um acidente vascular cerebral, e algumas doenças hereditárias [2].

A perda da audição pode, ainda, ser provocada por um problema mecânico no canal auditivo ou no ouvido médio que obstrói a condução do som (perda condutiva de audição). Na infância, o nervo auditivo pode ficar danificado pela parotidite, pela rubéola, pela meningite ou por uma infecção do ouvido interno, causando também deficiência auditiva [2].

Os dois tipos de perda da audição podem ser diferenciados comparando como uma pessoa ouve os sons conduzidos pelo ar e como os ouve conduzidos pela via óssea, mas pode dar-se o caso da perda auditiva ser tanto neuronal como condutiva [2].

A audição é medida em decibel (dB). O limiar ou a marca dos 0 dB para cada frequência refere-se ao nível em que jovens adultos com audição normal percecionam um tom puro (som com uma frequência específica) em 50% do tempo. A audição é considerada normal se os patamares de um indivíduo são 25 dB dos limiares normais, todos estes parâmetros são aferidos clinicamente através de um audiograma.

A gravidade da perda auditiva é classificada da seguinte forma [3], [4]:

- Leve (26-40 dB);
- Moderada (41-55 dB);
- Moderada a Severa (56-70 dB);
- Grave (71-90 dB);
- Profunda (90 dB);

Por outro lado, a frequência de perda auditiva é designada como baixa (<500 Hz), média (501-2000 Hz) ou alta (> 2000 Hz), o que condiciona os sons que cada indivíduo deixa de percecionar, pois cada som está relacionado com as diferentes frequências, incluindo os sons das vogais e das consoantes [3], [4].

Assim sendo, uma pessoa surda que apresente uma perda auditiva grave ou profunda, a comunicação com o meio envolvente é afetada. Neste contexto, foi necessário criar uma forma para os surdos e pessoas com dificuldade em oralizar se conseguissem expressar. Surgiu assim a Língua Gestual. No capítulo seguinte será feita uma descrição deste tipo de Língua bem como a sua articulação e principais características.

## 3. Língua Gestual Portuguesa

A comunidade fluente em Língua Gestual Portuguesa ascendeu aos 60 000 indivíduos em 2013 e está difundida em todo o país, incluindo os arquipélagos da Madeira e Açores.[5] A grande maioria da comunidade surda em Portugal comunica entre si através da Língua Gestual Portuguesa. Como é usual em línguas gestuais, esta é captada visualmente e tem uma estrutura característica e sistematizada, comparável à Língua Falada Portuguesa e por isso distinta da Língua Gestual de outro país. Apesar de serem as pessoas surdas as que mais dependem da LGP para se expressarem, a comunidade envolvente, como familiares, educadores, profissionais de saúde e técnicos também falam LGP [6].

É importante perceber o que faz da LGP um recurso insubstituível para a comunidade surda.

### 3.1. História da Língua Gestual

A história da Língua Gestual singrou à custa de muito esforço e reivindicações das Associações de Surdos.

A Constituição da República Portuguesa consagra, nos Princípios Fundamentais, art.º74, ponto 2, alínea h, a necessidade de o Estado “ Proteger e valorizar a Língua Gestual Portuguesa, enquanto expressão cultural e instrumento de acesso à educação e da igualdade de oportunidades”, fazendo da LGP uma língua amparada pela legislação portuguesa, em 1997, numa altura em que apenas 5 países do mundo inteiro o tinham feito.[7], [8] Contudo, este reconhecimento só foi adquirido após reivindicação da Comissão Coordenadora de

Iniciativas e Ações em Prol da População Surda Portuguesa (CCIAPPSP), no documento Carta Social da Pessoa Surda [9].

A Língua Gestual tem um passado relativamente recente. A primeira abordagem feita para a educação e entendimento entre surdos foi feita aproximadamente no ano de 1500 por Girolamo Cardano que afirmava “...a surdez e mudez não é o impedimento para aprender e o meio melhor é através da escrita... e é um crime não instruir um surdo-mudo.”. Este médico filósofo conseguiu instruir surdos usando a expressão por sinais. Em “*Refugium Infirmorum*”, Fray de Melchor Yebra ilustra o alfabeto manual usado no ano de 1613, em Espanha. Em 1644, John Bulwer escreve sobre a importância dos gestos na expressão dos surdos e defende que a linguagem da mão é a única natural para o Homem. Publica “*Chironomia, or The Art of Manuall Rhetorique*” onde se podem encontrar gravuras de inúmeros gestos. Em 1749, George Louis de Buffon é considerado o primeiro professor de surdos em França, introduzindo o alfabeto gestual espanhol e o ensino da leitura. Baseava-se numa orientação natural do desenvolvimento da linguagem, recorrendo a exercícios para melhorar a observação, a leitura labial e o tato. Segundo Seguin, era também ele surdo. No livro “*Instruction de Sourds et Muets par la Voix des Signes Méthodiques*” publicado por Charles Michel de l’Épée, conhecido como Abbé de l’Épée, por considerar insuficiente a linguagem natural dos surdos, que ele próprio tinha aprendido, inventou os “*signes méthodiques*”, para integrar a gramática da língua francesa. Abriu a Escola para Surdos em Paris e desenvolveu o método para a Língua Gestual, que serviu como base para a língua de sinais americana e outras línguas de sinais mundiais. Em 1823, é fundado o Instituto de Surdos-Mudos e Cegos, na Casa Pia, e, por decisão do Rei D. João VI, é chamado o especialista sueco Pär Aron Borg para o orientar. Este teve um papel muito importante no ensino dos surdos em Portugal, que com ele aprenderam a comunicar através de um alfabeto manual e da Língua Gestual de origem sueca. O método de ensino do Instituto de Estocolmo foi adaptado para o ensino em Portugal. O alfabeto manual inventado por Borg foi adotado pela Suécia, Portugal e Finlândia [10].

Toda a evolução até ao ano de 1880, no campo da Língua Gestual e educação de surdos sofreu um abalo após o Congresso Internacional de Surdo-Mudez, em Milão – Itália. Foi votada a adoção do método oral (também chamado de método intuitivo) pelas escolas de surdos e a Língua Gestual foi proibida oficialmente, argumentando que os surdos preferiam a língua de sinais à oralidade arruinando a capacidade da fala dos surdos. Esta votação foi aceite pelas delegações alemã, italiana, francesa, inglesa, sueca e belga. Só o grupo americano, liderado por Edward Miner Gallaudet se opôs, e assim se inicia um

período negro de aproximadamente 100 anos para os surdos e a educação de surdos na Europa [10].

O reconhecimento da Língua Gestual foi incentivado, mais tarde, após a publicação de uma pesquisa realizada por William Stokoe em 1965. Sendo diretor do laboratório de pesquisas linguísticas do Gallaudet College, desenvolve o conceito de querema como equivalente gestual do fonema e publica “*Sign Language Structure*”. Nesta publicação trata-se a *American Sign Language* (ASL) como uma língua genuína com morfologia distinta e complexa [10].

Em 1982 é implementado como método de escrita da Língua Gestual o SignWriting, o qual é introduzido em baixo.

## 3.2. Sistema SignWriting

Criado em 1974 por Vallerie Sutton (Presidente da Deaf Action Committee-DAC) foi baseado na notação que Sutton usava para a dança. Consiste numa forma de expor na forma escrita os signos usados em qualquer Língua Gestual, uma vez que apenas depende da morfologia gestual. Isto vem colmatar a necessidade que os Surdos sentiam para exprimirem as suas ideias por escrito, uma vez que a sua língua materna é a Língua Gestual e a morfologia desta língua é diferente da língua oficial falada. Expressando o movimento, posição e forma das mãos, pontos de articulação e as características não manuais na escrita, evoluiu-se de gravação de imagem para uma forma de registo mais natural para quem fala Língua Gestual, como se pode observar na Figura 3.1 [11].

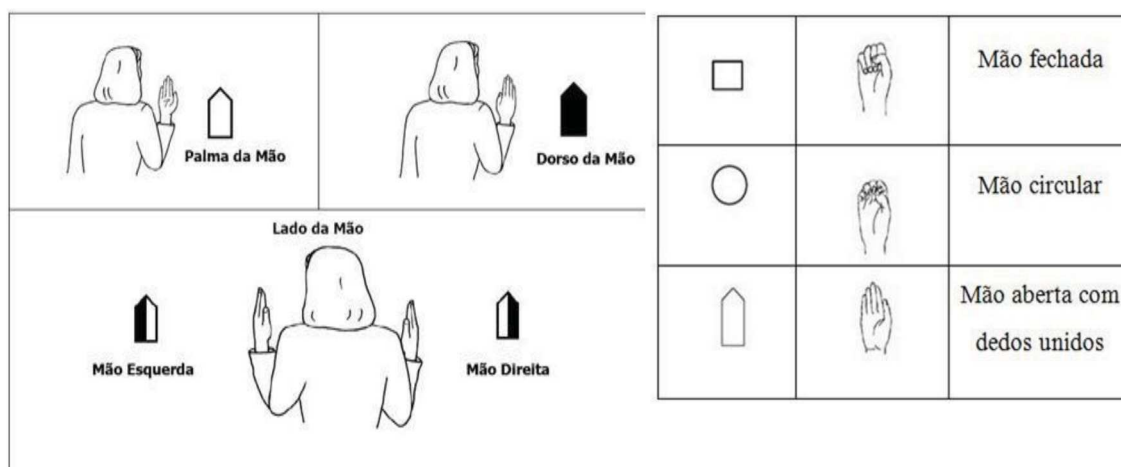


Figura 3.1: Exemplos de Gestos e os seus símbolos em SignWriting [76]

### 3.3. Morfologia da LGP

O significado de um gesto só é compreendido quando se apreende todas as características do gesto, sendo elas a configuração da mão, a orientação da mão, local da articulação, movimento, expressão facial e corporal e sintaxe. Em seguida é feita a descrição de cada uma das características.

#### 3.3.1.1. Configuração da mão

Considera-se configuração da mão todo o movimento executado com as mãos, podendo tomar múltiplas configurações. Em particular, é possível que a configuração da mão não dominante assuma o papel do local de articulação, em sinais produzidos com as duas mãos. A mão dominante é a mão direita para os destros e esquerda para os canhotos e a mão apoiante é a mão que não é dominante. As configurações das mãos são usadas de forma sequencial ou simultânea, tendo a mão apoiante um papel relevante. Pertence aqui a dactilologia ou ortografia gestual, isto é, a soletração com os dedos no espaço, utilizada apenas para representar o nome de pessoas, cidades ou países e o vocabulário técnico científico que não tem tradução direta da Língua Portuguesa para a LGP [12].

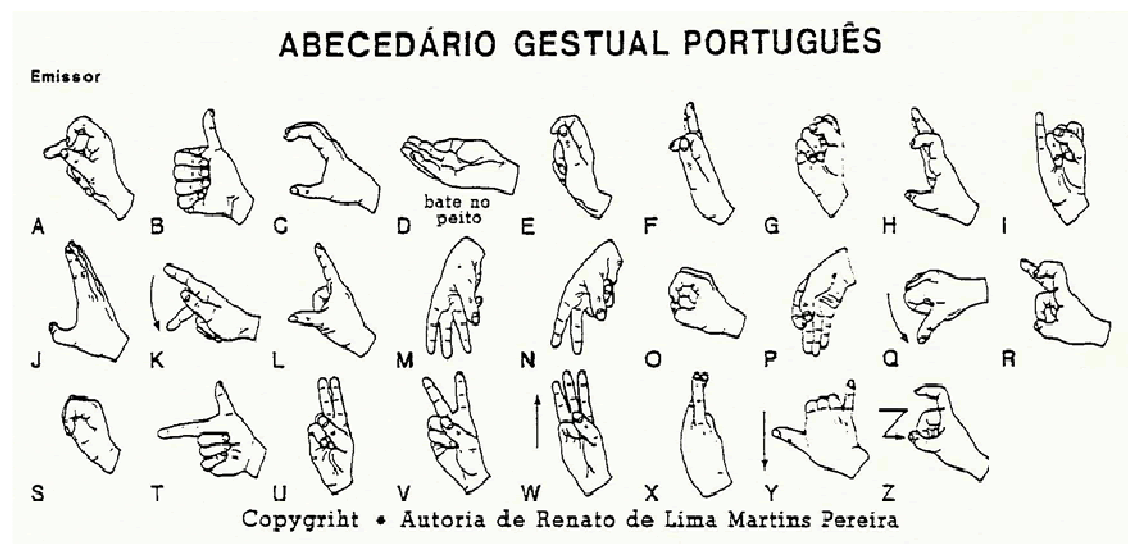


Figura 3.2: Alfabeto gestual da LGP. [77]

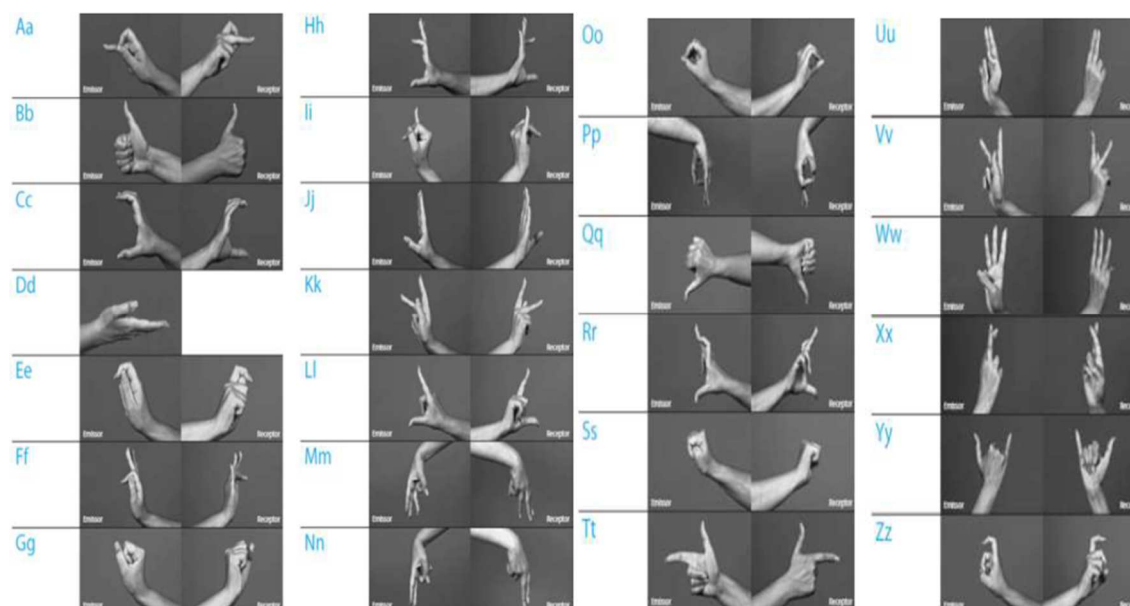


Figura 3.3: Alfabeto gestual observado da posição do emissor e do recetor. [12]

As imagens das colunas esquerda e direita correspondem às posições de emissor e recetor, respetivamente.

### 3.3.1.2. Orientação Manual

Relaciona-se com a configuração da mão e descreve o posicionamento da palma da mão, facultando as coordenadas que nos indicam para que lado a mão deve estar virada. Tem especial importância tanto na morfologia do gesto como no significado, por exemplo “ir” e “vir” têm configurações manuais similares diferindo apenas na direção do movimento [12].

### 3.3.1.3. Local de articulação

São três grandes espaços articulatorios, o espaço que abrange partes do corpo que servem como ponto de contacto, o espaço mais ou menos próximos do corpo e o espaço de articulação da mão dominante. Os principais locais de articulação são: cimo da cabeça; queixo; testa; pescoço; têmporas; ombro; olho; esterno; nariz; meio do tronco; bochecha; abdómen; orelha; braço; boca; antebraço e lábios [12].

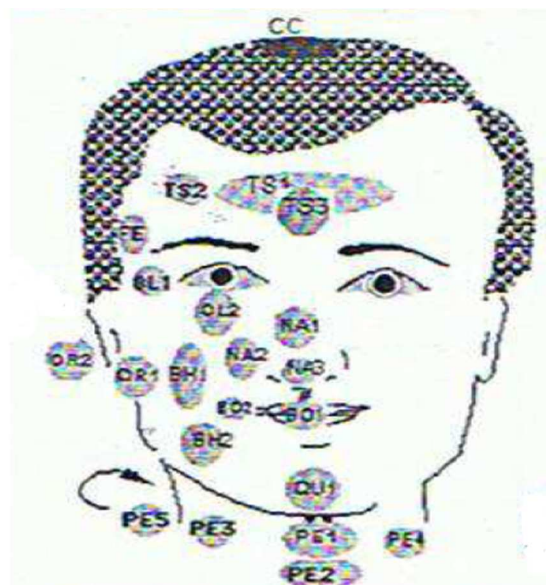


Figura 3.4 Pontos de articulação na face, pescoço e cabeça. [12]

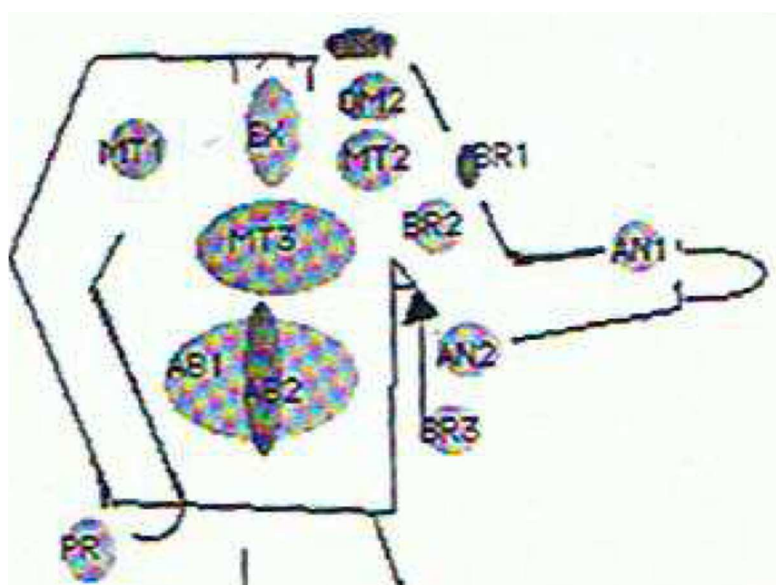


Figura 3.5: Pontos de articulação no corpo. [12]

#### 3.3.1.4. Movimento

O gesto pode ser estático ou dinâmico, pode resultar de um movimento isolado ou da combinação de dois ou mais movimentos contíguos. O dinamismo do gesto distingue-se por movimentos longos ou curtos, lentos ou rápidos e suaves ou tensos. Os movimentos dinâmicos são apontar ou indicar, bater, tocar, curvar, cruzar, dedilhar, enganchar, fletir ou dobrar, friccionar, inserir, ondular, raspar, torcer, em ziguezague, em espiral, entre outros.[12]



### 3.3.1.5. Expressão facial e corporal

A expressão facial e corporal conotam o gesto de significado. Interrogação, exclamação, negação, afirmação, admiração, dúvida, vergonha, medo, tristeza, etc... são algumas das emoções que se transmitem além do gesto. Existem alguns gestos da LGP que sem expressão facial/corporal deixam de ter significado. Assim a expressão corporal pode ser essencial na definição de sujeito. Além das expressões faciais e corporal há que ter em conta a vocalização que acompanha algum dos gestos, como se pode ver na Figura 3.6 [12].









	Bochecha Cheia
	Bochecha Vazia
	Boca Cerrada
	Expressão Facial Alegre E Aberta
	Expressão Facial Triste
	Olhar Cerrado/ Expressão Facial Carregada
	Olhar Cerrado/Expressão Calma
	Olhar Aberto/Expressão Exclamativa De Admiração Ou Surpresa
	Olhar Aberto /Expressão De Certeza
	Expressão Interrogativa Ou De Dúvida

Figura 3.6: Expressões faciais. Adaptado de [12]

### 3.3.1.6. Sintaxe

A estrutura da LGP é distinta da sintaxe/gramática da Língua Portuguesa, sendo geralmente organizada em sujeito-objeto-verbo (SOV). Os Verbos são sempre usados no

infinitivo, utilizando-se advérbios temporais para marcar a cronologia dos fatos e no caso de o sujeito referido estiver implícito no contexto não existe necessidade de o assinalar. Mais ainda, o género é apenas aplicado quando se faz referência a algo feminino usando o gesto de “Mulher” antes do gesto do objeto/sujeito, caso contrário apenas é marcado o objeto/sujeito. Não existem preposições em LGP, “de”, “para”, “em” são alguns exemplos. Existem ainda alguns casos em que o gesto no feminino e no masculino são diferentes (e.g.: mãe/pai). O plural é assinalado de diferentes formas, repetição do gesto, redobro (sinalização com ambas as mãos), incorporação (uso de um número para especificar quantidades reduzidas como sufixo do gesto relevante), determinativo (uso do signo, como sufixo, “muito”, para quantidades não contáveis).

As conotações de interrogativa, exclamativa fazem-se com as expressões faciais e corporais [13].

### 3.4. Reconhecimento de Língua Gestual

Atualmente, o reconhecimento de Língua Gestual é feito por intérpretes. Existem associações como a Associação de Tradutores e Intérpretes de Língua Gestual Portuguesa que tem por objetivo social “*interpretar, traduzir e realizar por todos os meios de comunicação a ligação entre o surdo e sociedade, recorrendo aos seus Intérpretes associados para o realizar*” [14]. No entanto, o acesso aos intérpretes depende do poder financeiro da pessoa surda e da disponibilidade do intérprete.

Por conseguinte, a comunidade científica aproveitou esta oportunidade para desenvolver tradutores de Língua Gestual automáticos, usando a tecnologia existente. Tem sido um processo evolutivo, ainda muito recente, no entanto têm surgido resultados positivos bastante prometedores.

No âmbito desta dissertação, frequentei um curso de verão de Língua Gestual Portuguesa (30 horas), ministrado pela Associação Portuguesa de Surdos em Lisboa. Aqui pude perceber a complexidade da LGP.

No capítulo seguinte serão introduzidas as tecnologias mais relevantes de momento, bem como o estado de arte do reconhecimento gestual.

## 4. Estado de Arte

### 4.1. Reconhecimento do Gesto Humano

A comunicação por gestos é algo natural para o Homem, mas a segmentação da análise de um gesto, para que haja reconhecimento/identificação do mesmo por uma máquina, é algo complexo. Como tal há que segmentar o problema em porções de resolução mais simples. Por ser um movimento que possui 27 graus de liberdade (incluindo os graus de liberdade do punho, ver Figura 4.1) é necessário particularizar os diferentes tipos de gestos (ver Figura 4.2) [15].

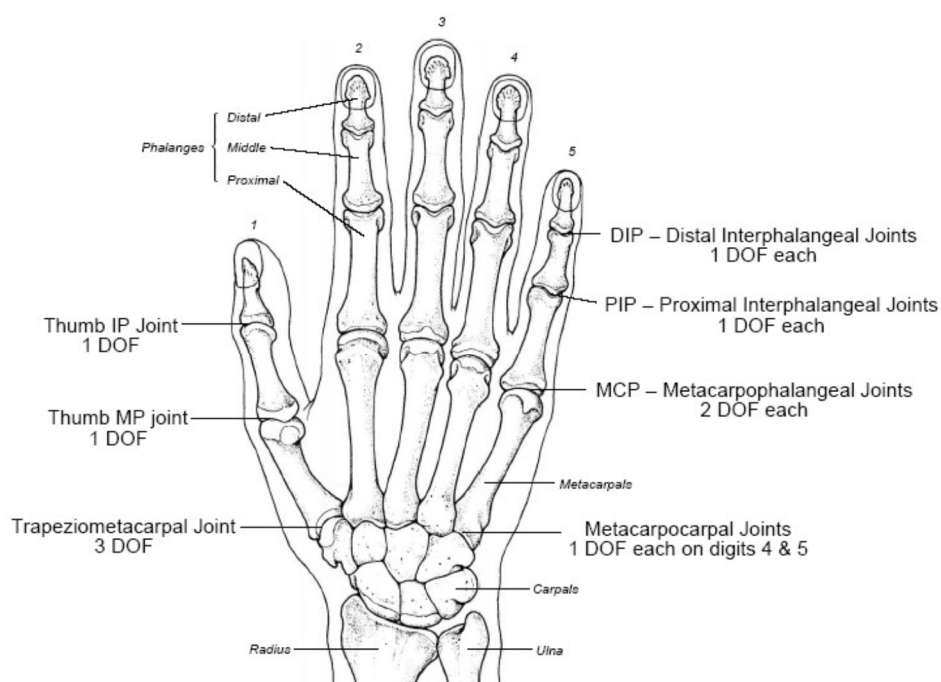


Figura 4.1: Graus de liberdades da mão e punho. [15]

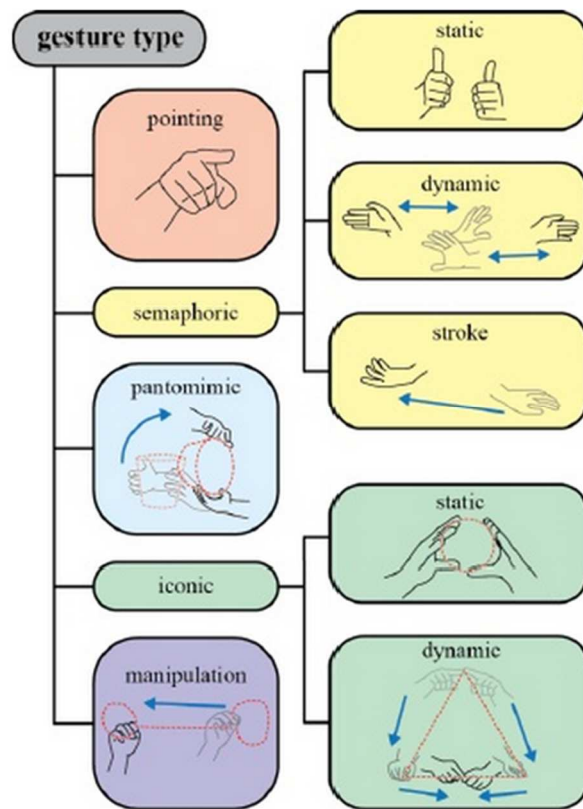


Figura 4.2: Classificação de Gestos por Aigner et al [16]

Aigner et al. repartiu os gestos em 5 tipos, apontadores, sinaléticos, pantomímicos, icônicos e de manipulação. Os gestos apontadores correspondem ao ato de apontar para algum objeto/direção. Os gestos de sinalética correspondem a gestos ou posturas das mãos que têm significado e são usados em situações específicas como símbolos e estão repartidos em gestos sinaléticos estáticos (como por exemplo o “*thumbs up*”), dinâmicos (como por exemplo “*olá*”) e o gesto de afagar (como por exemplo o movimento *swipe*). Os gestos pantomímicos são gestos que imitam características ou a atividade de determinado objeto ou tarefa. São gestos considerados quase arbitrários devido à sua variedade de movimentos e posturas. Os gestos icônicos demonstram características como forma, tamanho ou curvatura de um objeto e o significado, ao contrário dos gestos sinaléticos, não está implícito. Também estão divididos em gestos icônicos dinâmicos (o trajeto efetuado pelas mãos/dedos) e estáticos (a forma que as mãos fazem). Em último, estão os gestos de manipulação que estão relacionados com a interação com objetos/ferramentas. São usados para controlar posições, rotações e escalamento de entidades no espaço [16]. No esquema da Figura 4.3 está representado os diferentes tipos de modelos e abordagens a eles associados.

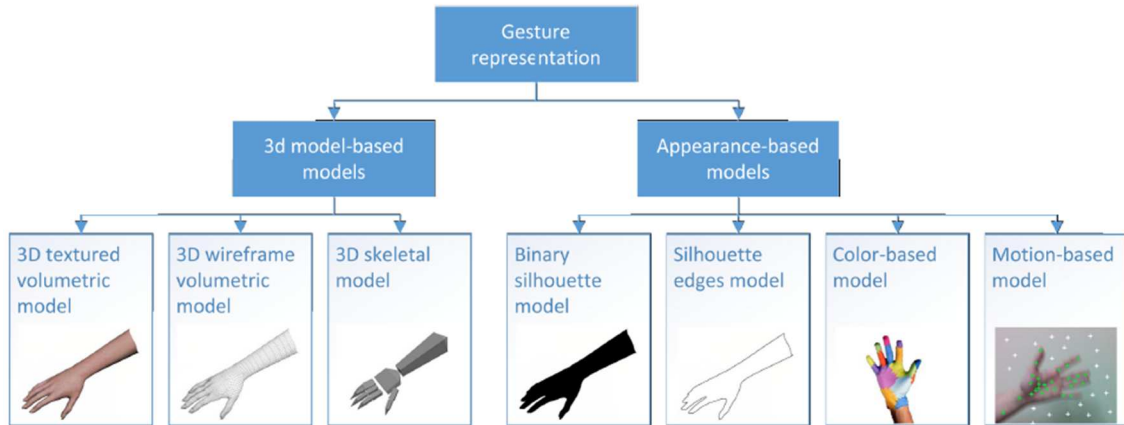


Figura 4.3: Esquema das técnicas de representação de gestos [16]

Na literatura, as abordagens sobre reconhecimento de gestos dividem-se em 2 grandes grupos, as abordagens que resultam em modelos 3D e os modelos que resultam da análise do aspeto da mão. Dentro das abordagens que resultam em modelos 3D existem abordagens volumétricas e modelos adaptados ao esqueleto, mas todas definem a forma da mão/braço e é esse fator que as caracteriza.[3]

Os modelos volumétricos reproduzem com alta precisão a forma da mão/braço, e são usadas malhas de vértices ou NURBS (*non-uniform rational B-spline*) ordenadas no espaço, animadas. Apesar da exatidão com que é definida a forma da mão/braço estas técnicas exigem *hardware* com capacidade de processamento razoável e raramente se conseguem aplicações em tempo real com estas abordagens (ver Figura 4.3) [16].

As abordagens cujos modelos resultam da análise da aparência da mão não exigem coordenadas espaciais rigorosas do objeto, baseiam-se apenas na forma/contorno do mesmo objeto retiradas das imagens capturadas. Normalmente são comparados padrões 2D entre as imagens. Nesta categoria encontram-se a modelação por binarização de imagens, modelação por contorno, modelo *color-based* e *motion-based* [16].

## 4.2. Tecnologias de Reconhecimento Gestual

Não é tarefa fácil fazer máquinas reconhecer e classificar gestos humanos, no entanto tem havido variadas abordagens a este desafio.

É fundamental dividir as abordagens, na tentativa de encontrar um protocolo de reconhecimento de gestos, em duas grandes áreas – as que usam sensores de captação de imagem e as que não usam [17].

◊ Sensores Não Baseados em Visão – São exemplo as braçadeiras/braceletes, Luvas com instrumentação e Dispositivos com *Tracking*. Estes dividem-se em 5 grupos segundo Yang et al. [17]:

- *Wearable* – dispositivos instrumentados que podem ser usados como adereço. Contêm sensores que analisam o movimento de uma determinada área do corpo ou do corpo todo. Temos por exemplo a CyberGlove R [18], Myo [19], IGS-190 [20], Mi.mu Gloves [21], Mister Gloves [22], Power Glove [23] e Enable Talk [24].



Figura 4.4: 1 – Myo [19], 2 - Enable Talk glove [24], 3 – CyberGlove [18], 4 – Power Glove [23]

- Biomecânico - tipo de dispositivo, que utilizam técnicas biomecânicas como eletromiografia, para medirem parâmetros do gesto. Exemplos deste tipo de dispositivos são os projetos desenvolvidos por Kim et al. em “*EMG-based hand gesture recognition for realtime biosignal interfacing*” [25] e Kainz et al. em “*Approach to Hand Tracking and Gesture Recognition Based on Depth-Sensing Cameras and EMG Monitoring*” [26].
- Inercial - estes dispositivos medem variações recorrendo a dispositivos como acelerómetros, giroscópios, magnetómetros para as medições, como por exemplo o comando da Wii [27].



Figura 4.5: Comando da Wii [27]

- Háptico – Os exemplos mais flagrantes deste tipo de aparelho são os *Smartphones* e *Tablets* com *Screens Multi-Touch*, que captam e analisam os movimentos neles executados.
- Eletromagnético – através da medição da variação de um campo eletromagnético (por exemplo, Wi-Fi) permite a detecção e reconhecimento de gestos humanos. Um exemplo de tais dispositivos é o WiSee, que aproveita as transmissões sem fio na atmosfera para permitir a detecção em toda a casa [28]. Outro exemplo é o smartphone *touchless* com 4 antenas de radiofrequência, desenvolvido na Universidade de Washington [29].

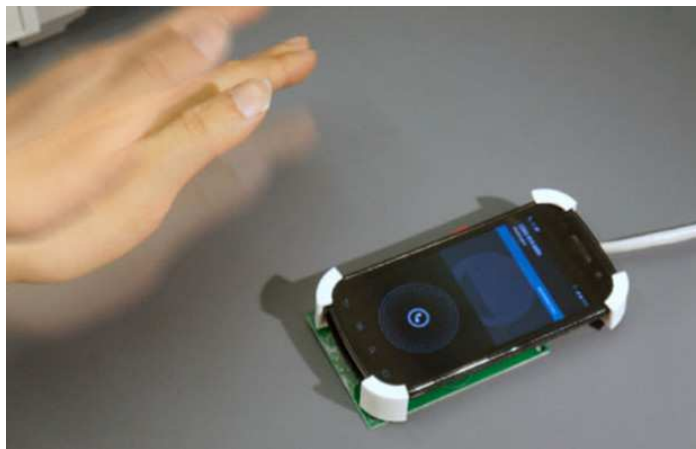


Figura 4.6: *Smartphone touchless* com 4 antenas de radiofrequência [29]

- ◊ Sensores Baseados em Visão – Implica o uso de Câmaras e/ou sensores óticos. Estes dispositivos dependem do uso de câmaras ou sensores óticos. Estão divididos segundo Yang et al. [16], [17], [30] em 4 grandes grupos:
  - Câmaras de Vídeo - técnicas de reconhecimento de gestos com base em dados derivados de apenas uma câmara empregando métodos de detecção,

tais como cor ou forma técnicas baseadas em forma ou contorno, detetores de aprendizagem a partir de valores de *pixel* ou detecção baseada em modelo 3D. É exemplo disto o Flutter App [31], [32].

- Estereoscopia - técnicas baseadas em imagens capturadas a partir de duas câmaras que fornecem uma aproximação dos dados gravados para a representação do modelo 3D.
- Técnicas Ativas - exigem a projeção de alguma forma de luz estruturada. Exemplos deste modelo de captura são a Kinect [33], Lazeeeye [34], Intel RealSense [35] e o Leap Motion [36],



Figura 4.7: 1 – Kinect [33], 2 – Lazeeeye [34], 3 – Leap Motion [36], 4 - Intel RealSense [78]

- Técnicas Invasivas - técnicas que exigem o uso de marcadores no corpo, como luvas de cor. Exemplo disso são os LED's no comando Move da Play Station [37].



Figura 4.8: Comando Move da PlayStation [37]

Como se verifica existe um grande conjunto de tecnologias com as quais o reconhecimento gestual é possível. Para melhor descrição e aprofundamento escolheu-se o Enable Talk, a Microsoft Kinect, a Wii Remote e o Leap Motion. Em seguida será feita



uma descrição dos produtos e estado-de-arte do desenvolvimento de aplicações com estas tecnologias.

#### 4.2.1.1. Enable Talk

O Enable Talk foi um projeto de estudantes que foi apresentado no concurso Microsoft Imagine Cup em 2012. A ideia do projeto consistia numas luvas capazes de traduzir Língua Gestual em som. Tendo como lema “*We’re giving a voice to movements*”, o projeto conseguiu o primeiro prémio, ganhando destaque nas tecnologias inovadoras capazes de ajudar a comunidade surda.

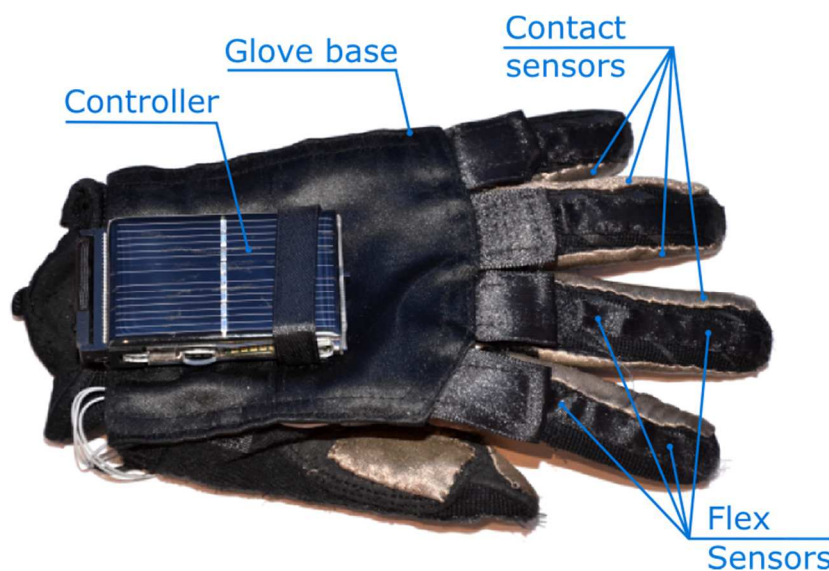


Figura 4.9: Enable Talk - Especificações das luvas. [24]

Como se observa pela Figura 4.9, a ideia consistiu na criação de uma luva com dois tipos de sensores e um controlador móvel. Cada luva é revestida com sensores flexíveis e sensíveis ao toque e inclui ainda uma bússola, giroscópio, e altímetro (ver Figura 4.10). Com isto, o utilizador com a luva consegue detetar movimento, direção e posição no espaço da sua mão, fazendo a transmissão da informação recolhida a um controlador na parte de trás de cada luva. O controlador converte os sinais e através da sua interface de comunicação *bluetooth* transmite o sinal final para um *smartphone* onde este é convertido em discurso para o utilizador [38]. Para reproduzir o sinal para som, a aplicação do *smartphone* faz uso do Microsoft Speech [39] e do Bing API [40].

A inspiração para a criação deste produto veio da observação de colegas de faculdade que eram surdos, observando que os mesmos tinham dificuldade para comunicar com os restantes alunos, o que resultava numa exclusão dos alunos surdos de certas atividades. Inicialmente, a equipa pensou em usar luvas disponíveis no mercado que pudessem ser adaptadas para interpretar sinais dos sensores, mas no final, eles optaram por desenvolver o seu próprio produto [41].

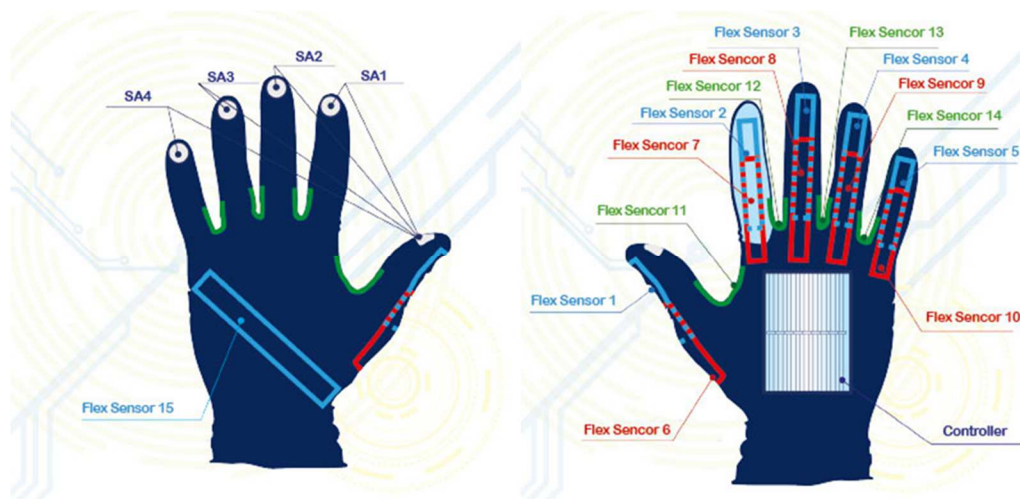


Figura 4.10: Detalhes Técnicos da Luva Enable Talk. [41]

De momento este produto encontra-se ainda em investigação não estando disponível para o público. O custo de produção do protótipo é de 50€. No entanto é esperado que com a produção em massa, o valor de venda ao público seja menor. De acordo com os estudantes, esta tecnologia poderia significar uma forma para cerca de 70 milhões de pessoas com deficiências auditivas e se fazerem entender e comunicar com a comunidade em seu redor sem o fator de exclusão social [24].

#### 4.2.1.2. Microsoft Kinect

A Microsoft Kinect é uma ferramenta com sensores de movimento desenvolvido inicialmente para a Xbox 360. Esta tecnologia foi desenvolvida principalmente para a indústria dos jogos de vídeo, no entanto, as suas fortes ferramentas de *software* fizeram deste dispositivo um dos mais usados na investigação de aplicações por controlo gestual. Pode ser adquirida por 150€, sendo o *software* de desenvolvimento gratuito. O sensor Kinect é composto no seu interior por uma câmara, dois sensores de infravermelho, um microfone e um acelerómetro, como se verifica na Figura 4.11. A nível técnico temos:

- ◊ A câmara RGB que captura imagem em três canais com uma resolução de 1280x960. A resolução da Kinect é de 1 cm para a profundidade, e para a largura e altura é de cerca de 3 mm, fazendo um registo de cerca de 30 *frames* por segundo;
- ◊ Um emissor de infravermelho e um sensor de infravermelho em profundidade. O emissor emite feixes de luz infravermelha padronizada e o sensor de profundidade lê os feixes infravermelhos refletidos de volta para o sensor. Os feixes refletidos são convertidos em informação de profundidade medindo a distância entre um objeto e o sensor. A leitura e conversão da informação obtida com esta luz são feitas com auxílio de um sensor *Complementary metal-oxide-semiconductor* (CMOS) e do PS1080 SoC que é um sistema multissensorial que sincroniza a informação recolhida;
- ◊ Um microfone multi-matriz, que contém quatro microfones de 24-bits para captar o som. No processamento inclui cancelamento de eco acústico e supressão de ruído;
- ◊ Um acelerómetro de 3 eixos configurados para uma gama até 2g, usado para detetar a posição da Kinect.

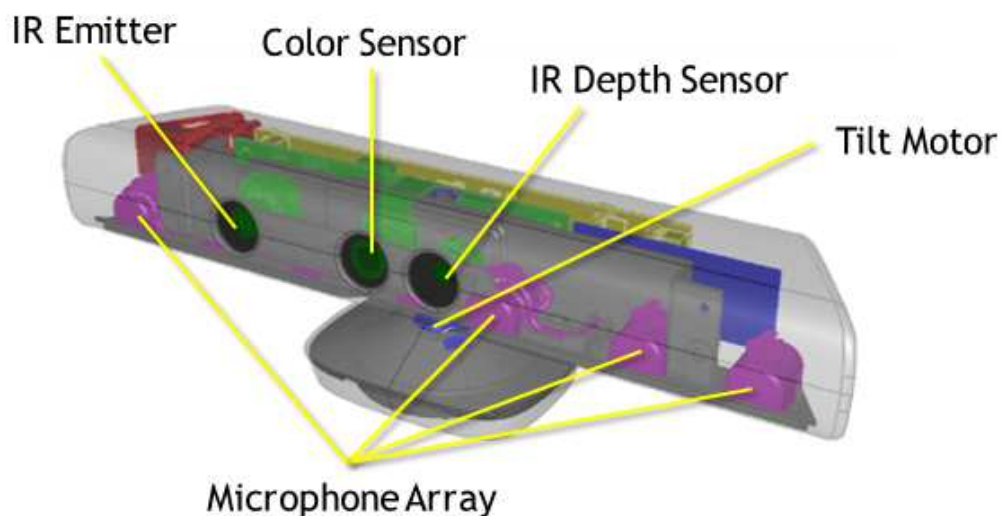


Figura 4.11: Componentes Técnicos da Kinect [33]

O principal objetivo da Kinect é fazer o processamento de informação em 3D. Uma das grandes vantagens deste sensor é o facto de ser bastante aberto aos utilizadores. A Kinect permite ao utilizador fazer um processamento da *Raw Data*, a imagem que vem diretamente dos sensores e das câmaras. Esta liberdade permite aos utilizadores fazer uma melhor programação e ajuste dos dados para as aplicações pretendidas [42].

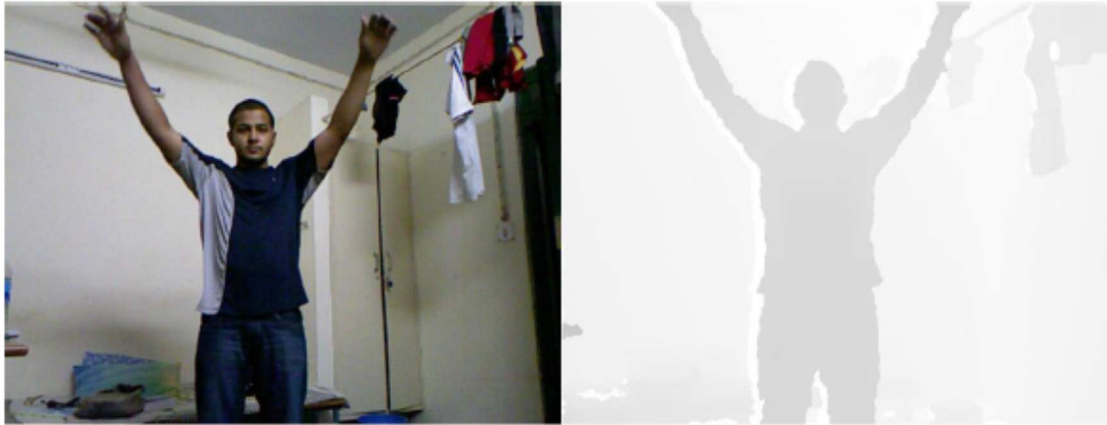


Figura 4.12: *Raw Data* fornecido pela Kinect. [42]

Para mapeamento de gestos, a Kinect desenvolveu um algoritmo de *Skeleton Tracking*. Este algoritmo permite identificar 20 juntas de esqueleto, obtendo a informação referente ao valor X, Y e Z de cada uma dessas juntas, como se observa na Figura 4.13.

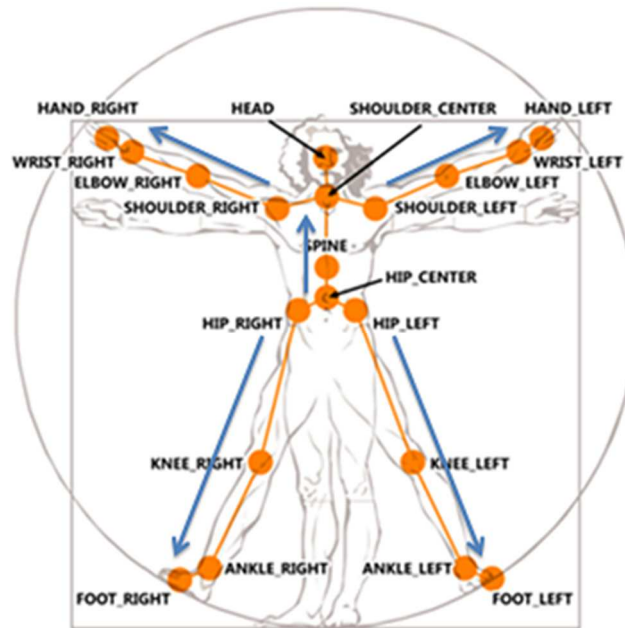


Figura 4.13: *Skeleton Tracking* – informação sobre articulações. [79]

Com o *Skeleton Tracking* é dada ao Kinect a funcionalidade de reconhecer pessoas no seu campo de visão e seguir o seu movimento. Em particular, a Kinect pode reconhecer até seis utilizadores, sendo que desses dois deles podem ser seguidos em detalhe com um ajuste ao modelo de *Skeleton Tracking*.

Com o lançamento das *Software Development Kit* (SDK) sobre *Face Tracking*, a Kinect abriu caminho para uma melhor classificação da expressão facial. Juntamente com o SDK do Kinect para Windows consegue-se fazer classificação das expressões faciais em tempo real, cativando o interesse da comunidade científica. Começa a ser visto como uma

oportunidade para criar aplicações de reconhecimento de língua gestual com base na configuração das mãos, expressão facial, localização e pontos de articulação, fatores que são importantes para uma boa classificação da língua gestual.

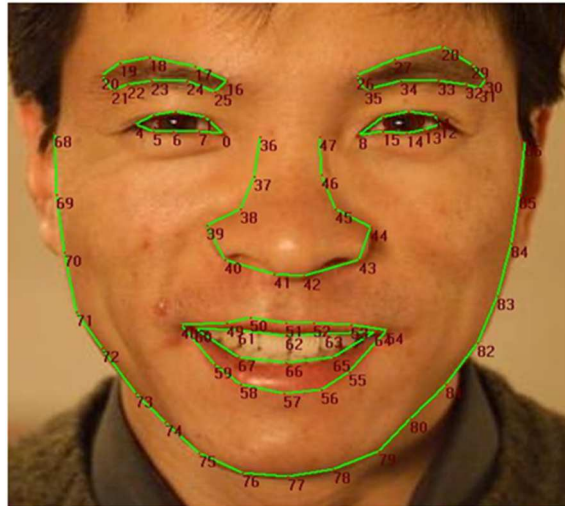


Figura 4.14: Face Tracking executado pela Kinect. [80]

Em suma, com o sensor Kinect o utilizador pode ter acesso a um vasto conjunto de dados (ver Figura 4.15), desde o *Raw Data* para o utilizador adaptar a seu gosto, como dados provenientes do *Skeleton Tracking* e *Face Tracking* já pré-processados:

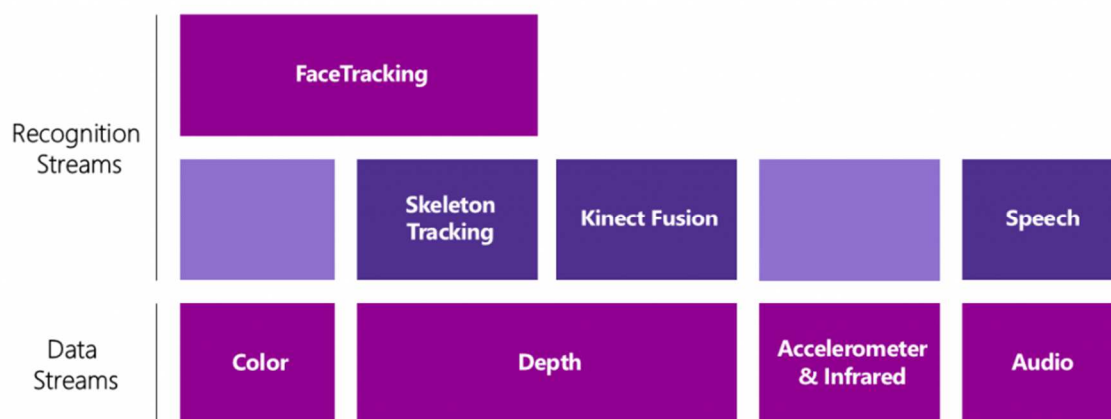


Figura 4.15: Informação disponibilizada pela API da Kinect. [81]

Vários projetos têm vindo a ser desenvolvidos com o reconhecimento gestual da Kinect. Em particular, nos últimos anos têm surgido vários trabalhos na área do reconhecimento gestual e tradução de Língua Gestual. Em seguida será feita uma descrição do estado de arte de reconhecimento de Língua Gestual com base no Kinect.

#### 4.2.1.2.1. Reconhecimento de Língua Gestual com Kinect

O reconhecimento de língua gestual tem sido um tema bastante estudado ao longo dos últimos anos, sendo que no entanto está longe de ser finalizado. Fatores como a iluminação e as condições em que o gesto é feito, influenciam a classificação [43]. No entanto, a Kinect é capaz de fornecer profundidade e cor dos dados adquiridos em simultâneo, facilitando assim o rastreamento da mão.

Com base nisto, uma equipa do *Key Lab of Intelligent Information Processing of Chinese Academy of Sciences*, desenvolveu uma aplicação para testar o reconhecimento da Língua Gestual [43]. O projeto foi o resultado de colaboração, facilitada pela *Microsoft Research Connections*, entre a Academia de Ciências da China, *Beijing Union University*, e a *Microsoft Research Asia* [44].

O protótipo apelidado de *Kinect Sign Language Translator* consegue interpretar os gestos da língua gestual e os converte em fala e escrita ou vice-versa. O sistema regista a conversa escrita e gestual. O sistema mostra ao utilizador uma tradução da Língua Gestual para escrita. Se o utilizador oralizar, o sistema com recurso a um avatar, gesticula o contexto oralizado para Língua Gestual. Na Figura 4.16 encontra-se representada a interface do protótipo da aplicação.



Figura 4.16: Interface do protótipo Kinect Sign Language Translator. [82]



A equipa para desenvolvimento do protótipo analisou a imagem que a Kinect gera como se verifica na Figura 4.17. Com o uso do *Skeleton Tracking*, os investigadores obtiveram as trajetórias 3D da mão, fazendo uma normalização das mesmas. Comparando com uma base de dados previamente construída, fazem uma classificação do gesto. Posteriormente fazem a oralização do gesto mais provável da base de dados com base em distâncias euclidianas [43].

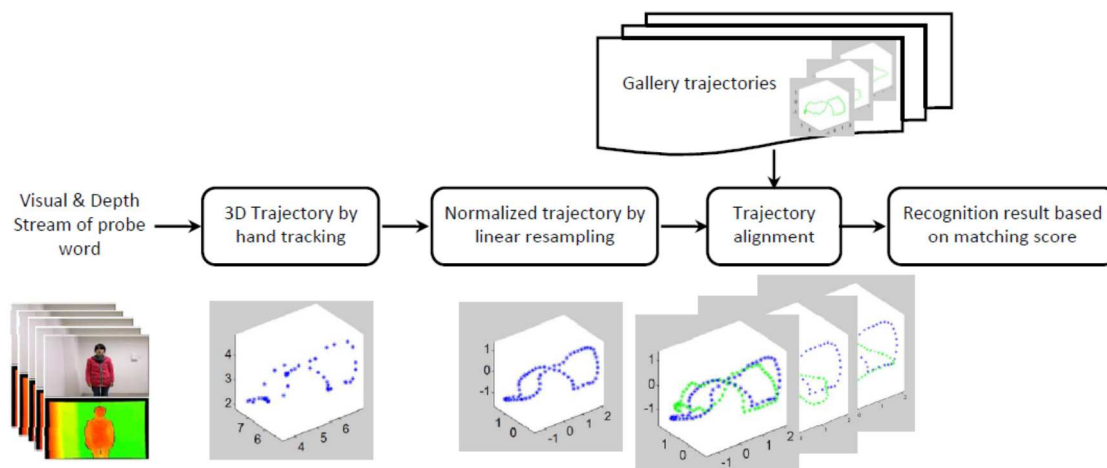


Figura 4.17: Diagrama de blocos do tratamento de dados para reconhecimento gestual. [43]

Os investigadores deixam claro que, de momento, o *Kinect Sign Language Translator* é um protótipo. No entanto acreditam que no futuro este sistema pode vir a melhorar a comunicação entre ouvintes e surdos, acabado com o fosso existente entre estas duas comunidades.

Em Portugal também tem sido feita investigação neste sentido. Exemplo disso é o projeto Virtual Sign desenvolvido pelo Instituto Superior de Engenharia do Porto em colaboração com investigadores da Faculdade de Engenharia da Universidade do Porto, da Universidade Aberta e de uma profissional em Língua Gestual Portuguesa [45]. O projeto Virtual Sign pretende desenvolver e avaliar um modelo que facilite o acesso dos surdos e deficientes auditivos aos conteúdos digitais, em particular aos conteúdos educativos e aos objetos de aprendizagem criando as condições para uma maior inclusão social dos surdos e deficientes auditivos.

Este projeto tem sido a ser desenvolvido ao longo de três anos, com financiamento de 100 mil euros pela Fundação Ciência e Tecnologia (FCT) [45]. O Virtual Sign usa dois dispositivos externos, uma luva com sensores e uma câmara Kinect que permitem identificar os gestos e os movimentos corporais e faciais. Os sinais são então transmitidos

para um computador onde são interpretados como texto. Numa outra vertente, este sistema pode ser usado para escrever texto que é traduzido em gestos por um avatar [46].

### 4.2.1.3. Leap Motion

O Leap Motion é um dispositivo desenvolvido pela empresa Leap Motion Inc. Este começou a ser comercializado em 2013 no *website* da empresa por um custo de 89.99€. O Leap Motion foi desenvolvido inicialmente na vertente dos jogos de vídeo, estando agora a sofrer uma expansão na sua aplicabilidade.

O Leap Motion é um dispositivo constituído por 2 sensores de imagem digitais (CMOS) e três emissores de infravermelhos posicionados de tal forma, que é possível com este aparelho ter noção das coordenadas espaciais dos dedos e da palma da mão. O Leap Motion reconhece e monitoriza mãos, dedos e ferramentas com forma parecida aos dedos. O dispositivo opera em proximidade com alta precisão (na ordem dos 0,2 mm em deteção de gestos estáticos [47]) e taxa de amostragem (na ordem das 300 *frames per second* usando USB 3.0 ou 50 *frames per second* com USB 2.0 [16]) obtendo as posições discretas, gestos e movimentos.

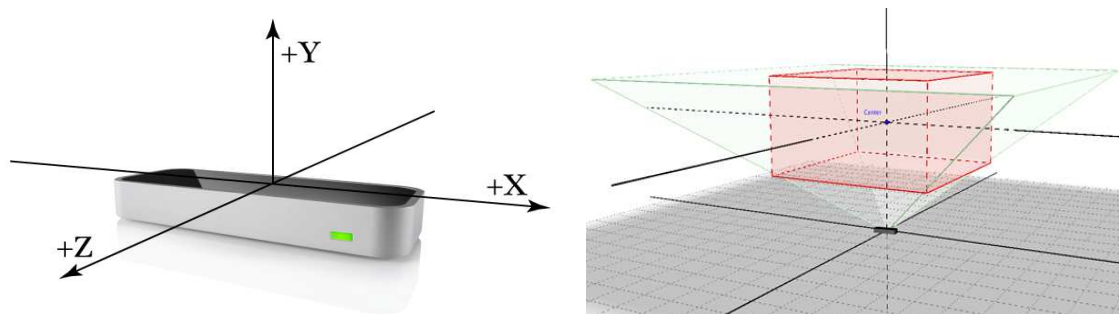


Figura 4.18: Disposição das coordenadas espaciais e campo de visão do dispositivo Leap Motion. [36]

Os seus sensores estão direcionados ao longo do eixo dos yy positivos e possui um campo de visão de 150 graus, na sua posição de funcionamento normal, e o intervalo de alcance do mesmo está compreendido entre os 25 e 600 mm acima do dispositivo [47], [48].

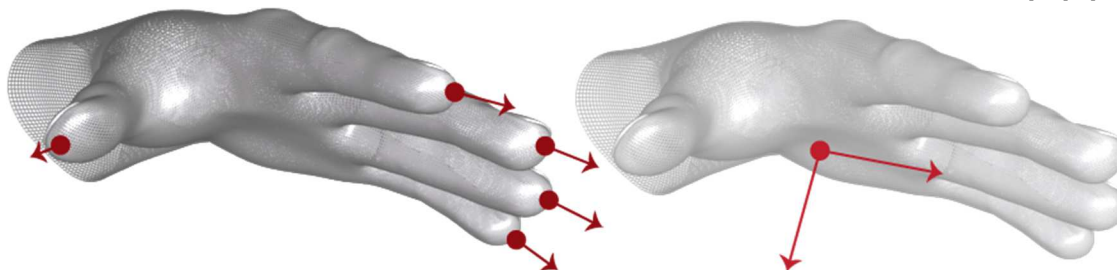


Figura 4.19: Informação extraída pelo dispositivo. [36]



Disponibilizado ao público em Julho de 2013, os *developers* tem à disposição o SDK para as linguagens C++, Python, C#/Unity, Objective-C, Java e JavaScript.

Quando o dispositivo deteta uma mão, dedo ou objeto é atribuída uma identificação na *frame*. Para cada identificação feita, do Leap fornece *update* da informação em cada *frame*, dando informação referente à posição, tempo da *frame*, velocidade e ângulo de cada um dos dedos e palma da mão. Existe, já implementada, a deteção de 4 gestos:

- *Circle*: Com o indicador em riste executa-se a trajetória circular no ar como mostra a Figura 4.20.



Figura 4.20: Gesto *Circle* detetado pelo dispositivo. [83]

- *Swipe*: movimento longo e linear da mão e dos dedos, como mostra a Figura 4.21.



Figura 4.21: Gesto *Swipe* implementado na API do Leap Motion. [83]

- *Key Tap*: movimento similar ao toque numa tecla, como se mostra na Figura 4.22.



Figura 4.22: Gesto *Key Tap* implementado na API do dispositivo. [83]

- *Screen Tap*: movimento similar ao toque num ecrã, como se mostra na Figura 4.23.



Figura 4.23: Gestos *Screen Tap* implementado na API do Leap Motion [83]

O principal objetivo da empresa para o produto é a interface pessoa-máquina. Ao contrário do que a Kinect e a Intel RealSense fazem, a Leap Motion, Inc., não forneceu a *cloud point of raw data* até Agosto de 2014 (está disponível desde então no *site* da empresa), sendo que toda a informação obtida com o aparelho era pré-processada, inacessível a alteração e o acesso pelo utilizador à informação era feito pela estrutura *Frame* da API.

Neste momento, a empresa tem ainda disponível no seu *website* duas versões do software, uma versão beta que usa *Skeletal Tracking* disponível para *developers* e uma versão oficial disponível ao público em geral. Uma das últimas versões do SDK (15-08-2014, V2.1.0+18736) abriu o caminho para exploração das imagens captadas diretamente pelo dispositivo, uma vez que agora a *Frame* contém essa informação disponível. Dado que quando a presente dissertação se iniciou ainda não existiam estas SDK's (*Skeletal Tracking* (ver Figura 4.24) e acesso a *Raw Data*), usaram-se as versões mais antigas das mesmas.

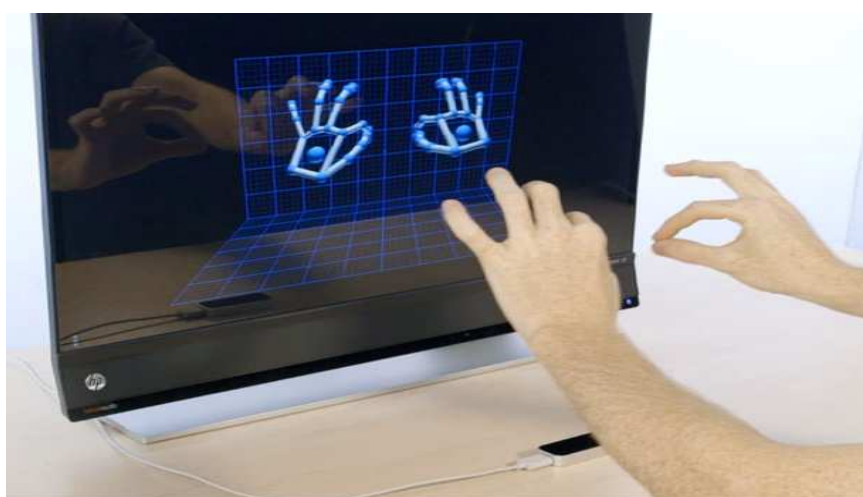


Figura 4.24: Leap Motion – Normal funcionamento do aparelho da V2 com *skeletal tracking* [84]

Dadas as potencialidades deste dispositivo tem havido um vasto interesse por parte da indústria e da investigação na sua utilização.

#### 4.2.1.3.1. Reconhecimento de Língua Gestual com Leap Motion

Com o aparecimento do Leap Motion o interesse em criar um dispositivo de tradução de Língua Gestual foi imediato por várias entidades. Um dos principais fatores que movem essa iniciativa é a dimensão do dispositivo. Este apresenta-se como sendo um dos mais pequenos dispositivos de reconhecimento gestual no mercado tornando-se bastante atrativo para o desenvolvimento de aplicações móveis. Uma das primeiras ideias de criação de um sistema tradutor de língua gestual com recurso a Leap Motion surgiu de duas *designers* portuguesas em Sidney. Idealizaram um dispositivo móvel que interpretasse os movimentos das mãos e o traduzisse para texto, como o que se mostra na Figura 4.25. Essas traduções apareceriam num pequeno ecrã, levando a que a interação entre ouvintes e surdos fosse mais fácil [49].



Figura 4.25: Sistema idealizado pelas Designers. [49]

Em Janeiro de 2014, um grupo de norte-americanos foi aceite num programa de aceleração patrocinado pela empresa Leap Motion Inc, o LEAP.AXLR8R [50]. Este grupo era composto por seis pessoas surdas e tinham em mente a criação de um Tablet ligado ao Leap Motion para traduzir Língua Gestual Americana (ASL) em texto e vice-versa. Assim sendo, foi criado o projeto/empresa MotionSavvy. Esta colaboração foi uma mais-valia uma vez que os mentores do MotionSavvy tiveram acesso a recursos que não estavam disponíveis para os utilizadores regulares do Leap Motion, como por exemplo, a equipa de engenheiros criadores do Leap Motion e a ligação Leap Motion/Android, ainda não disponível de momento para os *developers*, o que favoreceu o desenvolvimento do aplicativo [51], [52].

Segundo o fundador da empresa, o sistema reconhece até 100 palavras. Mas visto que a forma de como é feito o gesto, varia de pessoa para pessoa, o reconhecimento pode nem sempre funcionar. No entanto, o protótipo da empresa revela um potencial enorme no campo de reconhecimento gestual [53].

O sistema MotionSavvy é composto por um tablet, uma capa personalizada e um dispositivo Leap Motion, como se pode ver na Figura 4.26 [51].

O produto, quando partir para a venda ao público, está previsto ser lançado a um preço de 600\$, que inclui o tablet, o Leap Motion e o suporte. O utilizador terá de fazer uma subscrição mensal de 20\$ para aceder ao *software*, o que poderá ser uma desvantagem para esta aplicação.

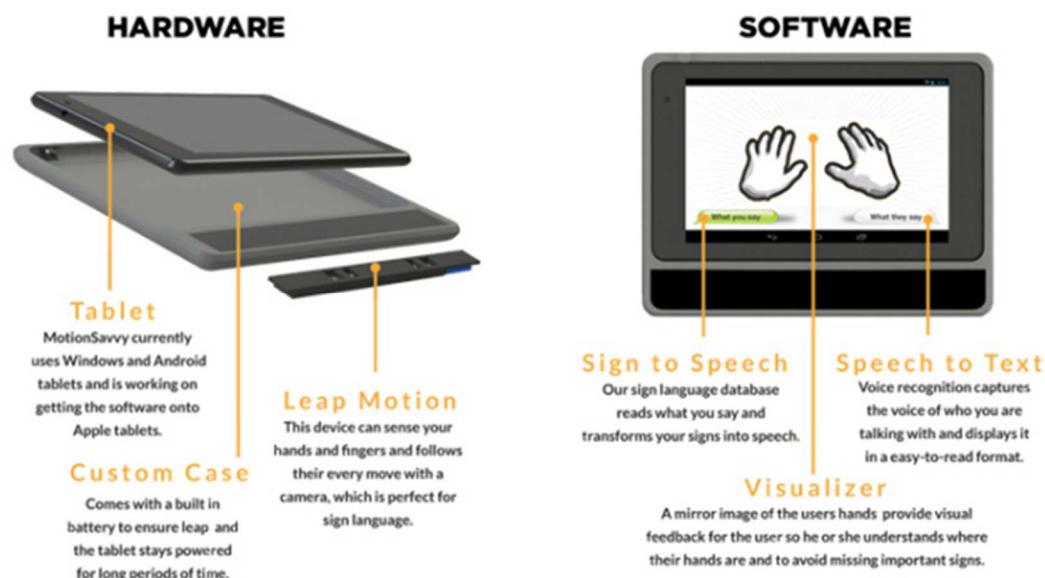


Figura 4.26: MotionSavvy – Componentes de *hardware* e *software*. [51]

Por ser um dispositivo que quase passa despercebido ao ser usado, a forma de utilização ser intuitiva e fácil e é acessível financeiramente, optou-se por explorar esta ferramenta. A capacidade de capturar e identificar no espaço as mãos e dedos com precisão milimétrica foi um fator decisivo no desenvolvimento das aplicações expostas nesta dissertação.

Na próxima seção será feita uma descrição de metodologias de identificação gestual que se espera vir a ser o primeiro tradutor *wearable* de LGP com base em Leap Motion, apto para funcionamento em qualquer smartphone/tablet Android ou iOS, sem qualquer tipo de subscrições.

# 5. Metodologia

Neste capítulo explica-se qual o caminho seguido para a concretização do objetivo da dissertação, o início do desenvolvimento de um tradutor de LGP. Dado ser um trabalho com um dispositivo recente, pouca literatura existe sobre o mesmo. Assim foram experimentadas várias abordagens de modo a identificar qual a abordagem mais adequada.

Foram feitas 5 abordagens diferentes, sendo elas:

- Acesso ao *Raw Data* do dispositivo, para processamento com auxílio do OpenCV;
- Avaliação da performance do dispositivo em gestos estáticos/poses e gestos dinâmicos/gestos;
- Classificação de gestos dinâmicos (numeração cardinal em LGP até 5), tendo como base em output do Leap Motion;
- Classificação de gestos dinâmicos (numeração cardinal em LGP até 5), com recurso a processamento dos dados recolhidos e um classificador de Hidden Markov Model;
- Classificação de um gesto dinâmico à escolha do utilizador com base na adaptação do classificador do *software wiigee* [54].

## 5.1. Raw Data

A primeira abordagem que se tentou foi adquirir as imagens captadas pelos sensores/câmaras do Leap, para serem analisados em tempo real por algoritmos de *Computer Vision* como o Open-CV, assim como é permitido fazer com a Kinect [13], [44]. No entanto, por não haver disponível a data, a informação do *Raw Data*, desistiu-se desta metodologia. A API do Leap fornecia apenas a informação limitada à localização das mãos e movimentos dos dedos. Este *output* do Leap foi a informação utilizada nas restantes metodologias.

### 5.1.1.1. LeapTrainer UI

Para melhor entendimento do que se podia concretizar com o SDK do dispositivo, utilizou-se esta *framework* disponível na plataforma GitHub [55] designada por LeapTrainer.js cujo autor, Rob O’Leary, disponibilizou com a licença do MIT.

Esta *framework*, que permite classificar vários gestos, é executado no *browser* principal do computador. Usando o output do Leap, esta *framework* expande a capacidade de reconhecimento. Fornece uma API que grava e mais tarde classifica os movimentos da mão no espaço.

Nesta aplicação, um *gesture* ou gesto dinâmico é considerado como um movimento com início e final reconhecível. Exemplo de gestos são o *swipe* para a direita ou para a esquerda, ou o *wave* e o reconhecimento é despoletado por um movimento rápido da mão seguido por uma interrupção rápida do movimento. *Pose* é designação dada para gestos estáticos da mão, em que existe movimento da mão no começo do gesto mas esta permanece parada nos instantes seguintes.

#### 5.1.1.1.1. Interface de Utilização

Ao abrir a aplicação é apresentada a interface da Figura 5.1.

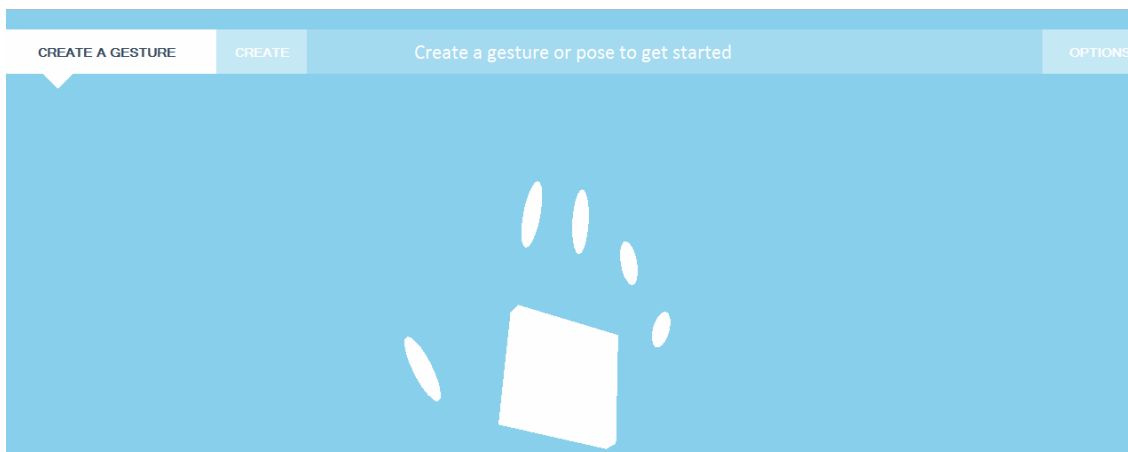


Figura 5.1: Interface Inicial do LeapTrainer.js

Para se iniciar o treino de um gesto ou pose, é necessário começar por escrever o nome do mesmo. Feito isto, a aplicação inicia uma contagem decrescente de 3 segundos, na qual se dá tempo ao utilizador de se preparar para treinar o gesto/pose. Este processo pode ser repetido para N gestos/poses.

Neste caso, foram efetuados 5 treinos diferentes, que consistem na numeração cardinal até 5 em LGP (por simplicidade na execução no gesto). Estes foram treinados como poses, isto é, gestos estáticos.

Para classificação de gestos dinâmicos, foram executados os gestos do alfabeto manual da LGP “a”, “b”, e “c”, dada a sua complexidade maior que a numeração.

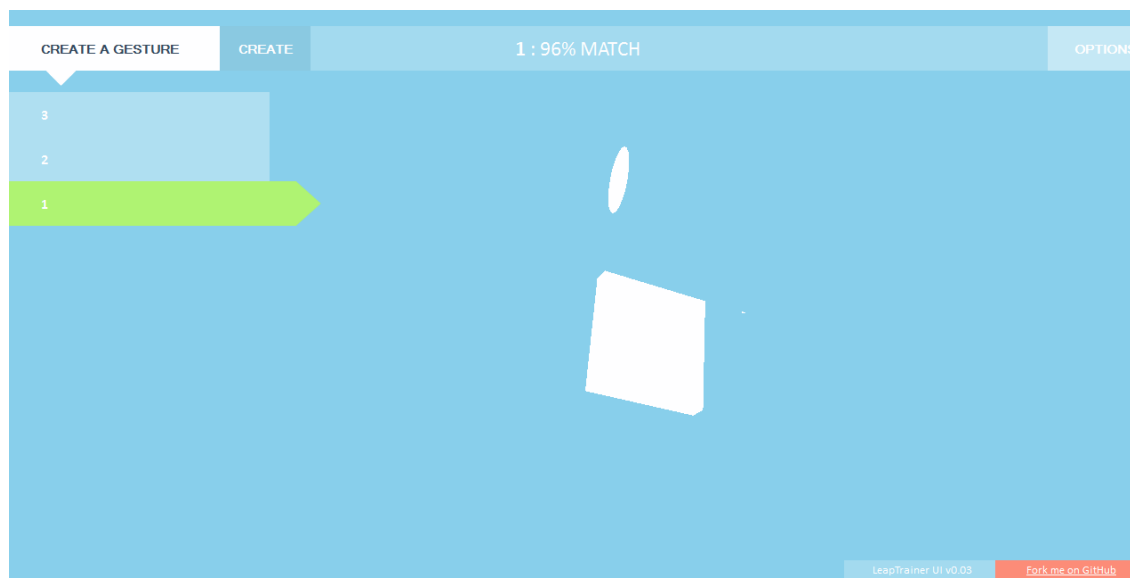


Figura 5.2: Treino Concluído.

Esta *framework* está desenhada para que só quando o gesto/pose é reconhecido envie mensagem de sucesso com a percentagem da correspondência. Funciona com vários gestos

e calcula também a probabilidade de correspondência da sequência adquirida ser alguma das que foram treinadas.

#### 5.1.1.1.2. Opções Configuráveis

Clicando em ‘Options’, o menu opções é exibido.

Neste menu existe a possibilidade de configurar alguns parâmetros tais como:

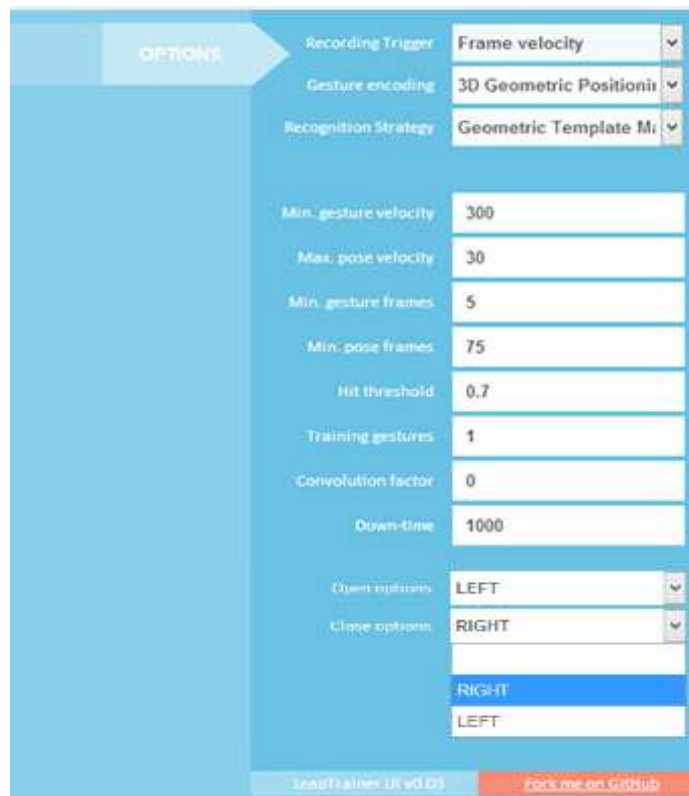


Figura 5.3: Menu de Opções. Adaptado de [55]

- Recording Trigger: só tem uma opção, Frame velocity.
- Gesture Encoding: Tem por 3 opções, (default) *3D Geometric Positioning* (sobreposição da aquisição e do treino e retiram se as diferenças entre os 2, como distâncias euclidianas), *High Resolution* (são usadas coordenadas das pontas dos dedos e da palma da mão no treino), *Low Resolution* (são usadas médias nos 3 eixos, xyz).
- Recognition Strategy: Corresponde ao método que será usado para fazer o reconhecimento. *Geometric Match Template* (default), *Artificial Neural Networks* e *Cross-Correlation* são as opções neste menu.



- Min. Gesture Velocity: Velocidade mínima contida na *frame* à qual o *trigger* de gravação de gesto dinâmico responde, *frames* com velocidade inferior à estipulada para a gravação. A velocidade na *frame* é igual à máxima velocidade da mão ou ponta do dedo detetada. (default: 300)
- Max. Pose Velocity: Velocidade máxima contida na *frame* à qual o *trigger* de gravação de pose responde. Velocidades superiores para a gravação. (default: 30)
- Min. Gesture *Frames*: Número de *frames* mínimo que contém um gesto. (default: 5)
- Min. Pose *Frames*: Menor número de *frames* que ocorre antes de gravar uma pose. Quanto maior este valor, maior o tempo de espera enquanto se executa a pose que se pretende treinar. (default: 75)
- Hit Threshold: Limite em que o gesto é ou não reconhecido, quanto maior, mais exato é o reconhecimento do gesto. (default: 0.65)
- Training Gestures: Número de treinos exigidos na fase de Treino.
- Convolution factor: Fator pelo qual as aquisições de treino serão convolvidas com uma distribuição gaussiana com o objetivo de expandir a informação dos dados. (default: 0 = disabled)
- Down-time: Tempo em milissegundos entre um gesto ser reconhecido e a ocorrência de um novo gesto. (default: 1000) (adaptado de [55])

A *framework* permite gravar treinos de gestos e carregar gestos em formato de código .js, voltar a treinar um mesmo gesto e exportar/importar o código-fonte do gesto treinado.

### 5.1.1.1.3. Funcionamento da Aplicação

É relevante escrever sobre as opções de Recognition Strategy, uma vez que são o cerne da solução para o reconhecimento gestual. Os dados analisados pelo *Recognition Strategy* variam consoante a opção *Gesture Encoding*. Se a opção selecionada no *Gesture Encoding* for *3D Geometric Positioning* será só analisado a posição das pontas dos dedos,

em cada *frame*. Se a opção for *Low Resolution* for escolhida, será analisado as médias das posições das pontas dos dedos em cada eixo espacial. Estes valores são adquiridos e analisados a cada 5 *frames*. Na opção *High Resolution*, os dados adquiridos são a velocidade, direção e o vetor normal à palma da mão e também a velocidade e a direção de cada ponta dos dedos. Este processo é feito todas as *frames*, o que faz deste processo o mais lento, mas é mais preciso que os anteriores.

Uma vez que esta *framework* foi desenvolvida com base em algoritmos já existentes, estes são vistos como uma “caixa negra” na descrição da aplicação por parte do autor, não se conseguindo grande compreensão desse código. Será apenas feita uma breve descrição teórica dos métodos usados sem grande detalhe.

#### 5.1.1.1.4. Método Geometric Match Template

Este método compara a sequência de dados treinada e a sequência de dados adquirida. Normalizando, primeiro, tanto a sequência de dados treinados como a sequência de dados adquiridos, sobrepõem-se e analisam-se as diferenças entre elas. O output será o correspondente ao gesto que tiver menor diferença na comparação.

Este método é similar ao que é usado pela *Kinect Sign Language Translator* descrito na seção 4.2.1.2.1.

#### 5.1.1.1.5. Método Cross - Correlation

Método de reconhecimento de gestos por defeito. Usa correlação cruzada entre dados treinados e dados de teste para calcular semelhanças entre as sequências.

A correlação cruzada é definida matematicamente por:

$$r = \frac{\sum_i [(x(i) - m_x) \times (y(i - d) - m_y)]}{\sqrt{\sum_i (x(i) - m_x)^2} \sqrt{\sum_i (y(i - d) - m_y)^2}}, d = 0, 1, 2, \dots, N - 1$$

Em que  $m_x$  – média da sequência  $x()$ ,  $m_y$  – média da sequência  $y()$  e  $d$  – atraso (*delay*). Este método é usado para estimar até que ponto duas sequências estão correlacionadas [56].

#### 5.1.1.1.6. Método Rede Neuronal

Aqui foi implementada uma rede artificial neuronal com o intuito de fazer o reconhecimento gestual.

Neste método requer-se um *input* de dados uniformes, portanto é feito uma normalização dos dados antes de se efetuar o treino da rede neuronal.

#### 5.1.1.1.7. Sumário

Esta *framework* serviu para perceber como obter, transformar e analisar os dados que o Leap permite manipular. Escrita na linguagem *JavaScript* (com extensão *JSON*), comentada quase ao detalhe, usa como base para a detecção por parametrização 3D geométrica, o artigo de Wobbrock et al. “*Gestures without libraries, toolkits or training: a \$1 recognizer for user interface prototypes*” [57]. Segue-se uma lista de vantagens e desvantagens desta *framework*:

Prós:

- Personalizável;
- Documentação das funções bem realizada;
- *Graphical User Interface* (GUI) atrativa e funcional;
- Criação/Treino de gestos/poses intuitivos e simples;
- Separação de Gestos estáticos e dinâmicos - pose e gesto, respetivamente.

Contras:

- Efetua demasiadas interpolações dos dados;
- Não mantém uma coerência na localização dos dedos e palma da mão;
- Consoante a complexidade dos gestos/poses aumenta, a localização e o reconhecimento correto destes baixa drasticamente;
- Processo de reconhecimento com baixa performance nas funções de High Resolution e Neural Networks.

## 5.2. Reconhecimento de Numeração

Decidiu-se iniciar um novo projeto em C# com integração na *Game Engine* Unity3D [58]. Usou-se a *framework* Unity3D para obter a visualização e a interação em tempo real com o utilizador.

O Unity3D faz parte de um conjunto de *softwares* conhecidos por *frameworks* usados para auxiliar na criação de jogos de vídeo, visto que integram em si um conjunto de ferramentas modulares que facilitam a vida dos *developers*. As principais ferramentas que tornam estas *frameworks* tão poderosas são as de *rendering* gráfico 2D e 3D, sistemas de simulação de eventos físicos e *scripting* de animação e som. Este facto foi visto como uma mais-valia para este projeto, uma vez que facilitou a renderização virtual das mãos vistas pelo Leap Motion, permitindo o utilizador ter uma noção de como o dispositivo está a ver a orientação manual em tempo real, como dito anteriormente. Outra mais valia é a nível do *scripting* de som, que é facilitada com esta *framework*, tornando-se mais fácil de associar um som ao gesto correspondente (som gravado previamente com um microfone) [59].

A programação em Unity é uma programação orientada a objetos, o que faz com que a programação do projeto desenvolvido possa ser feita de forma segmentada. Existe uma clara distinção entre os scripts de modelação e controlo da mão e os de processamento de dados. Tornou-se mais fácil, com o uso do Unity3D, focar os pontos mais importantes da dissertação, nomeadamente na classificação dos dados captados pelo Leap, tendo ao mesmo tempo uma interface simples e visual para o utilizador.

A possibilidade de exportar para várias plataformas como Windows, MacOS, Android, iOS, foram outras particularidades que no futuro serão relevantes e inovadoras com o decorrer deste projeto, pois permitirá exportar o projeto desenvolvido para outras plataformas com facilidade.

A empresa Leap Motion, Inc. disponibiliza as bibliotecas nativas para C# e alguns exemplos de como se integra o Leap com os projetos de Unity3D. A documentação mais atualizada sobre a API está disponível *online* no site da empresa para *developers*.

Esta aplicação foi desenvolvida toda de raiz com recurso ao SDK fornecido pela Leap Motion, Inc. O objetivo desta aplicação é classificar gestos dinâmicos correspondentes à numeração cardinal até cinco. Foi focada a dinâmica do gesto uma vez que este foi um ponto fraco identificado no LeapTrainerUI.

Nesta abordagem é feita uma análise direta dos dados fornecidos pelo Leap Motion, nomeadamente a posição de cada dedo e palma da mão. Esta abordagem foi seguida visto que cada gesto pode ser analisado como contagem de dedos esticados numa mão, para facilitar o reconhecimento e não tem qualquer característica facial associada ao gesto, para que tenha significado. Esta é a forma mais simples de começar a desenvolver um classificador de LGP com base em Leap Motion.

O modelo da mão foi criado com base em exemplos fornecidos pela documentação do Leap Motion. Neste modelo foi implementado um algoritmo de captura da posição da ponta dos dedos e da palma da mão. Posteriormente é feito o processamento da informação capturada, calculado a distância euclidiana entre a palma da mão e os cinco dedos. Com as distâncias calculadas, calcula-se o máximo e o mínimo da variação das distâncias para cada um dos dedos, estabelecendo um limiar para definir quando essa variação é quantificada como um número. A escolha do limiar será analisada na secção 7. Internamente a aplicação tem associado um som aos números de 1 a 5, reproduzindo o som respetivo quando o número é reconhecido pelo processamento ditado anteriormente.

#### 5.2.1.1. Interface de Utilização

A simplicidade na interação com o utilizador era um dos requisitos na GUI desta aplicação.

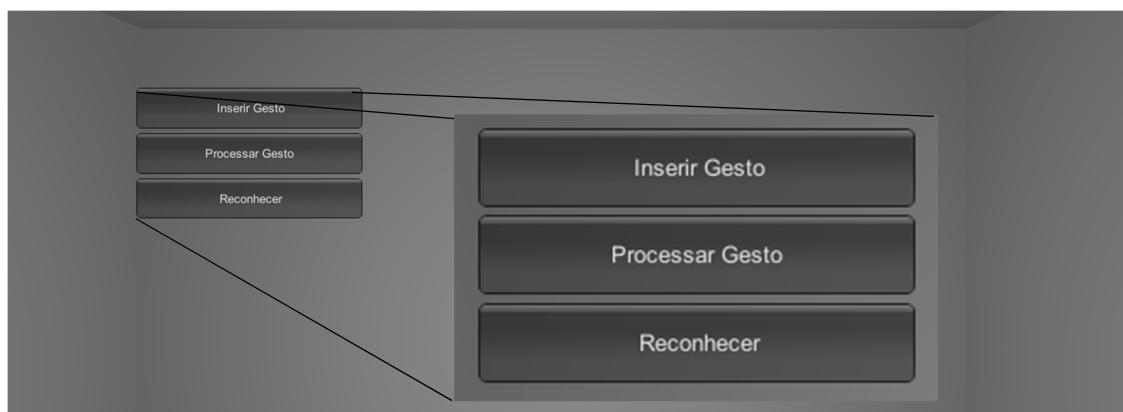


Figura 5.4: GUI usada para a aplicação Reconhecimento de numeração até 5.

Em baixo, na Figura 5.5, é apresentada uma imagem com o modelo da mão que reproduz os movimentos capturados pelo aparelho.

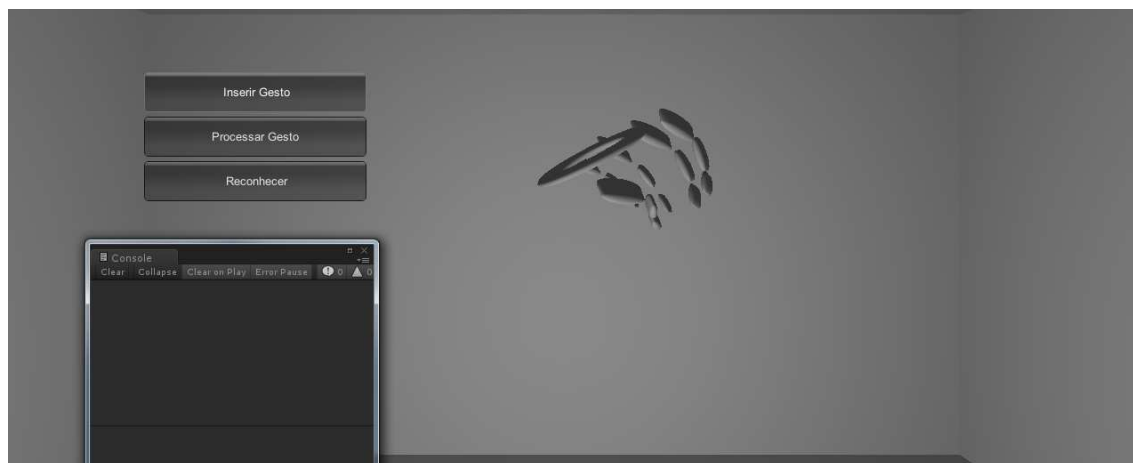


Figura 5.5: Interface com a mão direita representada. Esta interface mostra em tempo real os movimentos capturados pelo dispositivo.

Como este modelo funciona por comparação de distâncias, o modelo da mão que se vê no Unity deve ter a direção correta e as pontas dos dedos devem estar o mais próximo umas das outras antes de iniciar o processo de reconhecimento. No processamento do gesto são extraídas distâncias euclidianas entre a palma da mão e os dedos. Caso a distância ao início não seja a mínima possível, no processamento dos dados a aplicação pode gerar resultados errados e gerar falsos positivos ou falsos negativos. Após clicar no botão “Inserir Gesto”, o utilizador deve abrir a mão, dispondo um número e clicar “Processar Gesto”, mantendo a pose (ver Figura 5.6 e Figura 5.7). Resta então ao utilizador clicar no botão “Reconhecer”, que faz o processamento descrito anteriormente, para que na consola mostre o valor correspondente ao número gesticulado, e o utilizador ouvir o som associado a esse mesmo número.



Figura 5.6: Exemplo de postura da mão no início da aplicação.

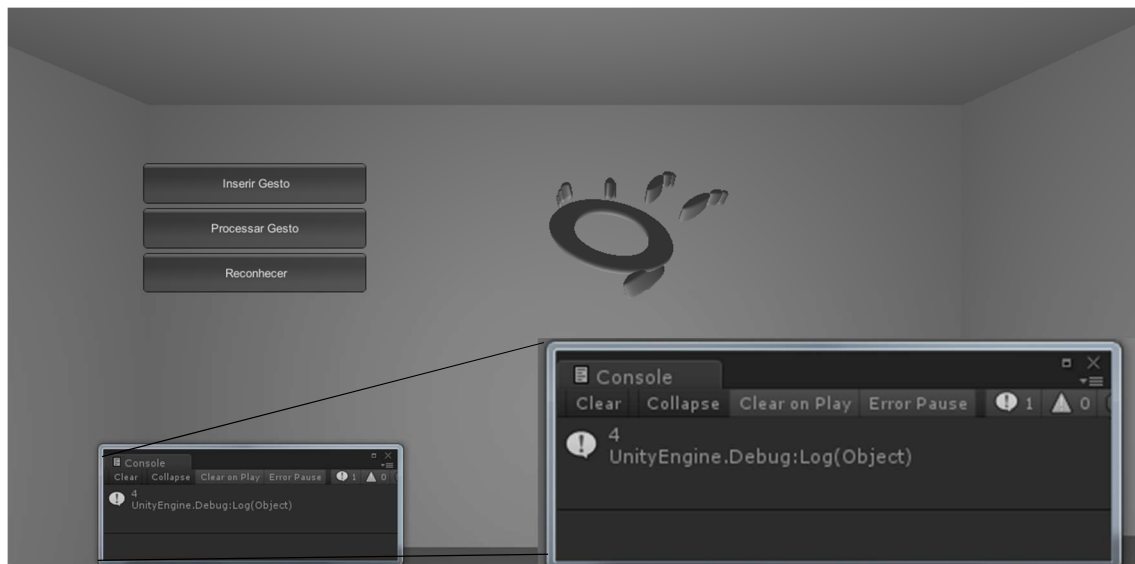


Figura 5.7: Exemplo da conclusão da aplicação. Neste caso, o número 4 foi exibido com a mão esquerda.

No caso de o utilizador não efetuar nenhum número é exibida a mensagem de “Punho Exibido” na consola (ver Figura 5.8).

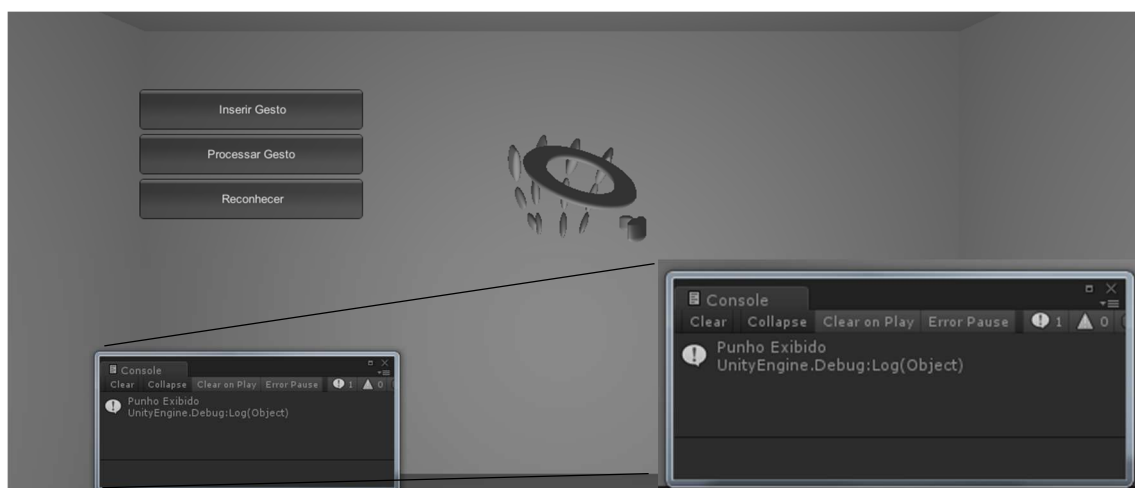


Figura 5.8: Resultado da aplicação quando um punho é exibido.

No caso de não haver captura de mãos pelo Leap, a aplicação corre, mas o resultado mostra na consola uma mensagem de erro (ver Figura 5.9).

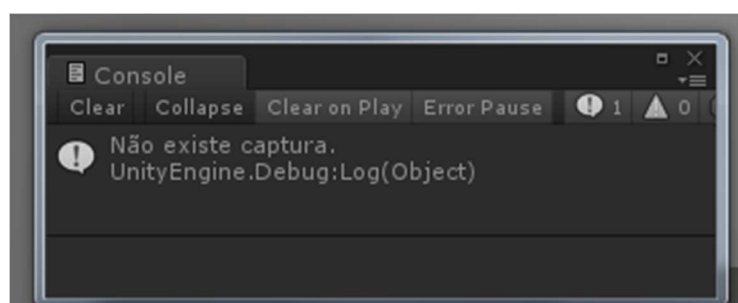


Figura 5.9: Resultado da aplicação quando não existe uma mão no campo de visão do aparelho.

### 5.2.1.2. Funcionamento da Aplicação

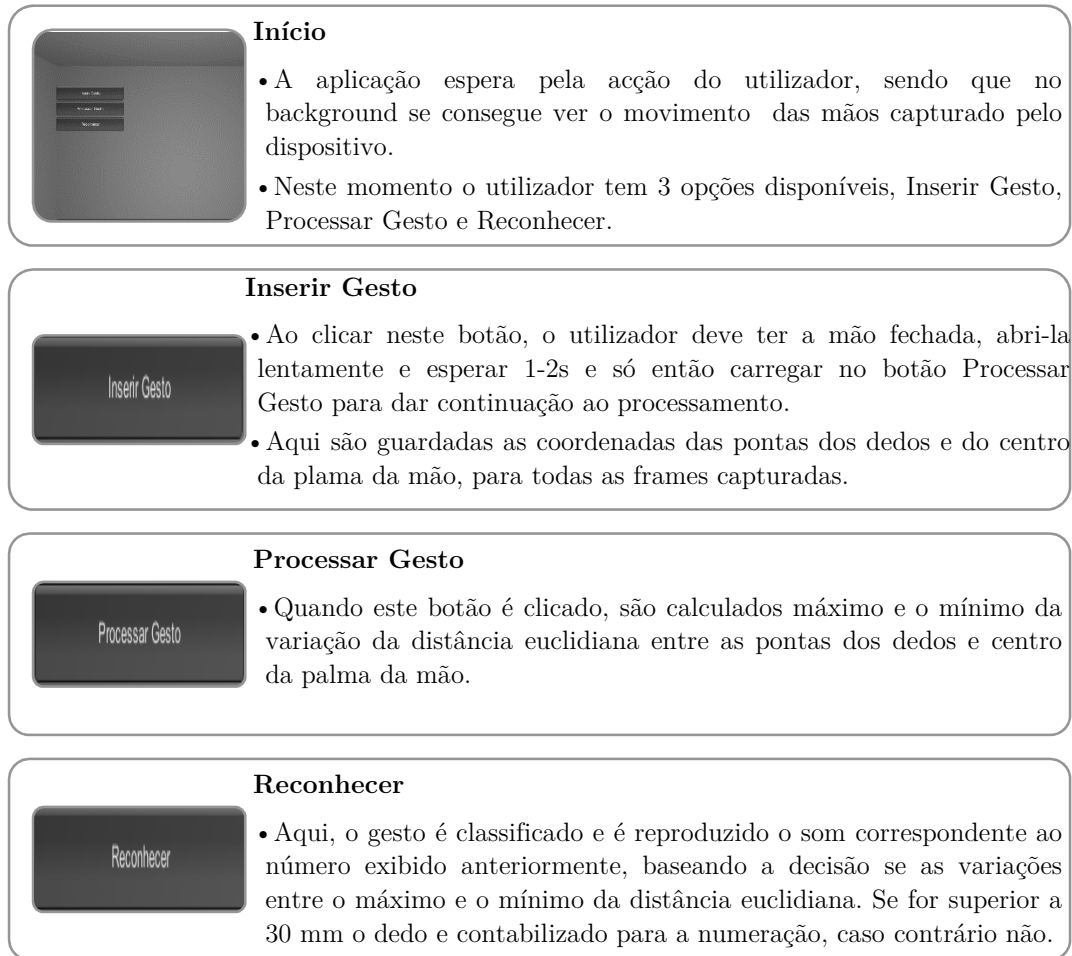


Tabela 5.1: Esquema demonstrativo do funcionamento da aplicação.

Esta aplicação tem a particularidade de associar ao número reconhecido, o som respetivo. Esta é uma aplicação simples mas funcional.



## 5.3. Reconhecimento de Numeração com HMM

Nesta metodologia foi implementado um algoritmo de classificação com ajuda da *framework* Accord. O objetivo desta aplicação continua a ser a classificação, com sucesso, da numeração de 1 a 5 usando apenas a posição. O motivo pelo qual se utilizou a *framework* Accord.Net é explicado na seção 5.3.1.2. Espera-se melhorias na classificação da numeração cardinal até 5 em LGP. Seguidamente será explicado no que consiste o Accord.NET.

### 5.3.1.1. Accord.NET *Framework*

A *framework* Accord.NET tem o objetivo de tornar mais acessível programar para *machine learning*, sendo uma plataforma de programação científica. Tendo por base outra *framework*, a AForge.NET (outra *framework* de computação científica que proporciona algoritmos de processamento de imagem, entre outros), a Accord.NET compila uma vasta biblioteca de aplicações, como processamento de dados estatísticos, *machine learning* e reconhecimento de padrões. Tem uso principalmente em aplicações que impliquem *Computer Vision* e *Computer Audition*.

Esta *framework* foi escolhida por compatibilidade com C# (apesar de se ter usado uma adaptação das *Native Libraries* para Unity3D, que podem ser encontradas aqui [60]) e pela facilidade de implementação dos modelos estatísticos.

### 5.3.1.2. Modelo para Reconhecimento de Gestos

Um gesto possui um atributo dinâmico, por definição. A compreensão desse movimento tem-se revelado uma tarefa árdua e complexa. Um gesto pode ser analisado como um padrão, e a identificação de um padrão é um procedimento estatístico. Como tal, o reconhecimento de um gesto trata-se de um problema que pode ser dividido em 4 partes: a localização, a trajetória que executa, a análise desses dados e a identificação do padrão [61].

A localização está assegurada pelo Leap Motion, uma vez que a API nos garante funções que podem ser implementadas na aplicação que fazem o *output* de coordenadas espaciais e velocidades das pontas dos dedos e da palma da mão. Com estes dados conseguimos ordenar a trajetória, gravando a cada *frame* o conjunto dos *outputs* das funções da API do aparelho.

A análise dos dados corresponde à solução do objetivo. Na literatura é possível encontrar vários artigos que apontam para a utilização de modelos estatísticos para o reconhecimento de gestos:

<b>Autores - Ano</b>	<b>Objetivo</b>	<b>Modelo: Método</b>
<b>Nowicki <i>et al.</i> 2014</b>	Classificação de gestos estáticos e dinâmicos arbitrários.	Estatístico: Hidden Markov
<b>McInerney, J. 2013</b>	Classificação de gestos personalizáveis pelo utilizador	Estatístico: Hidden Markov
<b>Poppinga <i>et al.</i> 2008</b>	Classificação de trajetórias executadas com Wii Remote	Estatístico: Hidden Markov
<b>Barczak <i>et al.</i> 2005</b>	Simular movimentos da mão em tempo real	Estatístico: Viola-Jones <i>Framework</i>
<b>Lin <i>et al.</i> 2004</b>	Seguir o movimento de uma mão articulada através de imagens de vídeo	Estatístico: Nelder-Mead Method
<b>Ye <i>et al.</i> 2004</b>	Classificar gestos manipulação de objetos	Estatístico: Visão Estéreo
<b>Bowden <i>et al.</i> 2004</b>	Classificação de Língua Gestual usando classificação linguística e cadeias de Hidden Markov para as transições temporais	Sintática: Uso de mangas compridas e fundo branco
<b>Tomasi <i>et al.</i> 2003</b>	Classificação de ortografia gestual em 3D	Estatístico: Combinação de classificação 2D e interpolação em 3D
<b>Schlenzig <i>et al.</i> 1994</b>	Classificação de Movimento 2D das mãos	Estatístico: Hidden Markov

Tabela 5.2 Resumo da literatura sobre Identificação de Gestos. Adaptado de [15], [62], [63].

Assim, seguindo a literatura encontrada resolveu-se utilizar o modelo de Hidden Markov, usando os recursos do Accord *Framework*. Na tentativa de classificar movimentos

capturados pelo dispositivo, o modelo de Hidden Markov é a melhor aproximação visto que se trata de modelo estatístico genérico representado por conjuntos de vetores-chave. Após o treino, as novas sequências observadas serão classificadas como se fossem geradas a partir do modelo que apresenta maior verosimilhança.

### 5.3.1.3. Modelo de Hidden Markov

O modelo de Hidden Markov é um modelo estocástico com a particularidade de ser dinâmico, de forma a depender apenas do estado em que se encontra para prever o próximo estado, e ignorando o caminho percorrido até chegar ao estado corrente. Os modelos markovianos podem ser considerados sistemas de estados, onde os estados são conjuntos de vetores probabilísticos e as transições entre estados dependem apenas do presente estado em que se encontra. Nos modelos de Markov, as transições podem variar no espaço temporal discreto, sendo designado, então, por cadeias de Markov ou no espaço temporal contínuo, designados então por processo de Markov. Os modelos de Markov têm especial importância na modelação onde nem todos os estados são conhecidos, neste caso são chamados de modelos escondidos de Markov (HMM), e é, portanto, um modelo estatístico flexível [64]–[66].

Um modelo escondido de Markov pode ser definido por 5 componentes:

1. Um conjunto de  $O_k$  símbolos, que resultam de  $k$  observações,
2. Um conjunto  $N$  de estados  $n_i$ , onde em cada estado  $n_i$  é possível observar-se  $b_i(k)$  símbolos, com  $i=0, \dots, N$  e  $k \in O_k$ ,
3. A matriz de probabilidade de transição de estados dada pela matriz

$$A = [p_{ij}], \quad \text{onde } p_{ij} = P(n_j(t+1) | n_i(t))$$

4. Uma distribuição de probabilidade para as observações em cada estado dada pela matriz

$$B = [b_j(k)], \quad \text{onde } b_j(k) = P(O_k | n_j)$$

5. Um vetor de distribuição inicial

$$\Pi = [\Pi_i], \quad \text{com } \Pi_i = P(n_i(0)).$$

O vetor distribuição é representado como  $\lambda = (A, B, \Pi)$  [67], [68].

Para ajustar a modelação do HMM é necessário responder a 3 questões impostas por Rabiner em [64]:

◊ 1º Problema

Dada a sequência de observáveis  $O = O_1, O_2, \dots, O_k$  e o modelo  $\lambda = (A, B, \Pi)$ . Como calcular de forma eficiente a probabilidade dessa sequência ser gerada pelo modelo,  $P(O|\lambda)$ ?

◊ 2º Problema

Dada a sequência de observáveis  $O = O_1, O_2, \dots, O_k$  e o modelo  $\lambda = (A, B, \Pi)$ . Qual a sequência de estados ótima,  $N = N_1, N_2, \dots, N_k$ , mais provável?

◊ 3º Problema

Como ajustamos os parâmetros do modelo  $\lambda = (A, B, \Pi)$  para maximizar  $P(O|\lambda)$ ?

A resolução destes 3 problemas resulta na adequação do modelo aos dados. O 1º problema trata-se da avaliação do modelo, isto é, dada uma sequência de observações como se calcula a probabilidade daquele modelo originar a sequência. O 2º problema é um problema de otimização em que se tenta “espreitar” para os estados escondidos do HMM. Calcula-se a sequência “mais correta”, estimando o critério de desempenho, infelizmente este critério pode ser mais do que um pelo que se terá de estimar, também, qual o critério que melhor ajusta o modelo. O 3º problema está relacionado com o treino do modelo, isto é, qual o modelo  $\lambda = (A, B, \Pi)$  que melhor se adapta à sequência de observação [64], [69].

Na realidade, a forma de implementar HMM, faz-se numa ordem diferente. Treinamos o modelo com uma sequência de dados específica e obtém-se os parâmetros ótimos do modelo (3º Problema), refina-se o modelo usando o 2º Problema obtendo sequências ótimas para o modelo de HMM obtido no passo anterior, otimizando os parâmetros por iteração (usando o Algoritmo de Viterbi ou o Algoritmo de Baum-Welch). No final, executa-se o 1º Problema, testando as novas sequências adquiridas para perceber a que modelo melhor se relacionam, com base na maior verosimilhança (*maximum likelihood*) [64].

Para obter os melhores resultados na resolução do 3º Problema de HMM usou-se o Algoritmo de Baum-Welch, por na literatura apresentar melhores resultados em comparação com outros algoritmos [15], [16], [65], [70].

O algoritmo de Baum-Welch, também chamado de *Forward-Backward*, é um processo iterativo que converge para máximos locais, em apenas 2 passos recursivos. Se

tivermos  $\lambda = (A, B, \Pi_1)$  e se calcular um novo modelo  $\gamma = (C, D, \Pi_2)$  com base nos parâmetros de  $\lambda$  existem 2 opções:

1. Ou  $\gamma = \lambda$  e então  $\gamma$  é um ponto crítico da função de probabilidade, significando então que  $\lambda$  é o modelo que otimiza a sequência de observação.
2. Ou  $\gamma$  é mais provável que  $\lambda$ , sendo  $P(O_k | \gamma) > P(O_k | \lambda)$ , sendo o modelo  $\gamma$  o mais provável de ter originado a sequência de observação.

Até que  $\gamma = \lambda$  [65], [71].

Nesta aplicação implementou-se o HMM recorrendo à *framework* Accord.NET e o esquema resultante do algoritmo resume-se à Figura 5.10.

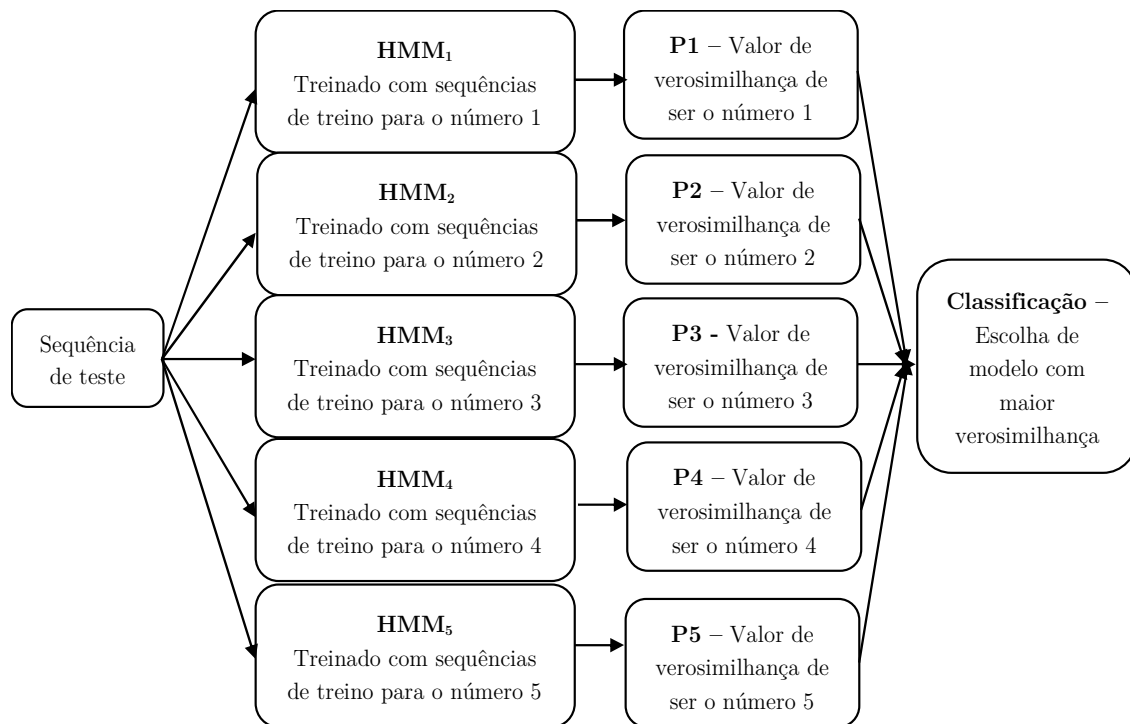


Figura 5.10: Esquema representativo do modelo da decisão usado na aplicação.

#### 5.3.1.4. Interface de Utilização

Neste caso, o *input* para o treino Hidden Markov será repetido quantas vezes o utilizador quiser. Terá que iniciar o treino (clicando no botão “Treinar”) de punho fechado e lentamente realizar o número que pretende, clicando de seguida no botão “Parar treino” (ver Figura 5.11).

Após ter executado o set de Treino, clicando no botão “Gravar Gestos” é executado o algoritmo de treino do modelo de Hidden Markov para aquele gesto. Será atribuída uma etiqueta ao gesto que corresponde à ordem com que foram treinados, que será apresentada na consola quando se executar o reconhecimento. O reconhecimento deve-se proceder da mesma forma que ao realizar o treino, dar início ao algoritmo com a mão fechada e lentamente abrir, clicando em seguida “Parar Reconhecimento”. No final obtém-se o valor da etiqueta atribuída na consola e o som correspondente.



Figura 5.11: Interface da aplicação Reconhecer Numeração até 5.

```
// Creates a sequence classifier containing 2 hidden Markov Models with 2
// states and an underlying multivariate mixture of Normal distributions as
// density.
classifier = new HiddenMarkovClassifier<MultivariateNormalDistribution>(
    classes: 2, topology: new Forward(2), initial:
initialDensity);

// Configure the learning algorithms to train the sequence classifier
var teacher = new
HiddenMarkovClassifierLearning<MultivariateNormalDistribution>(
    classifier,

    // Train each model until the log-likelihood changes less than
0.0001
    modelIndex => new BaumWelchLearning<MultivariateNormalDistribution>(
        classifier.Models[modelIndex]){

        Tolerance = 0.0001,
        Iterations = 0, //default

        FittingOptions = new NormalOptions ()
        {
            Diagonal = true, // only diagonal covariance matrices
            Regularization = 1e-5 // avoid non-positive definite errors
        }
    }
);

// Train the sequence classifier using the algorithm
double logLikelihood = teacher.Run(seq_treino, labels_to_use);
```

Tabela 5.3: Excerto da implementação do Classificador HMM usando a API da framework Accord.NET

### 5.3.1.5. Funcionamento da Aplicação


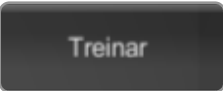
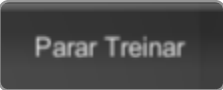
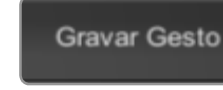
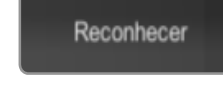
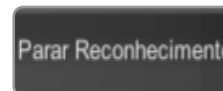
	<p><b>Início</b></p> <ul style="list-style-type: none"> <li>• A aplicação espera pela acção do utilizador, sendo que em background se consegue ver o movimento das mãos capturado pelo dispositivo.</li> <li>• Neste momento o utilizador tem 5 opções disponíveis, Treinar, Parar Treinar, Gravar Gesto, Reconhecer e Parar Reconhecimento.</li> </ul>
	<p><b>Treinar</b></p> <ul style="list-style-type: none"> <li>• Ao clicar neste botão, o utilizador deve ter a mão fechada, abri-la lentamente e esperar 1-2s e só então carregar no botão "Parar Treinar" para dar continuação ao processamento.</li> <li>• Aqui são guardadas as coordenadas espaciais das pontas dos dedos e do centro da palma da mão, para todas as frames capturadas.</li> </ul>
	<p><b>Parar Treinar</b></p> <ul style="list-style-type: none"> <li>• Quando este botão é clicado, são calculados o máximo e o mínimo da variação da distância euclidiana entre as pontas dos dedos e centro da palma da mão.</li> </ul>
	<p><b>Gravar Gesto</b></p> <ul style="list-style-type: none"> <li>• Clicando neste botão, assume-se que o utilizador já realizou o conjunto de treino, querendo treinar o modelo HMM para posterior classificação.</li> <li>• É guardada a informação relevante, matriz de transição, matriz de observações, vetor de distribuição inicial.</li> </ul>
	<p><b>Reconhecer</b></p> <ul style="list-style-type: none"> <li>• Aqui o utilizador tem que executar um gesto que estiver na sequência de treino e então quando concluir o gesto clicar no botão "Parar Reconhecimento".</li> </ul>
	<p><b>Parar Reconhecimento</b></p> <ul style="list-style-type: none"> <li>• Quando este botão é clicado, é aplicado o mesmo processamento à sequência adquirida durante o Reconhecer.</li> <li>• É computado o valor de Maximum Likelihood dentre os modelos treinados e pode ser observado na consola o número que foi atribuído ao gesto quando "Gravar Gesto".</li> </ul>

Tabela 5.4: Esquema demonstrativo do funcionamento da aplicação.

A fase de Treino-Parar Treino-Gravar Gesto pode ser repetida  $n$  vezes, sendo atribuída a cada entrada, após clicar no botão “Gravar Gesto”, uma etiqueta que corresponde à ordem com que foram efetuados os diferentes gestos.

Para conseguir uma melhor performance do classificador é aconselhável executar um conjunto de 15 treinos, na secção seguinte será explicado o motivo.

## 5.4. Classificador de 1 Gesto

O desenvolvimento do Reconhecimento de Numeração até 5 levou à tentativa de classificar um gesto personalizado pelo utilizador. Neste subcapítulo descreve-se esta abordagem.

Tal como a aplicação anterior, esta também usa o modelo de HMM para classificar, mas este classifica um gesto com maior complexidade. É utilizada para a classificação, apenas a velocidade da ponta do dedo com coordenada absoluta zz maior, com intenção de reduzir a complexidade associada ao uso de HMM e do algoritmo Baum-Welch. Esta característica figura como ponto forte e fraco deste método, como será abordado na análise de resultados.

Esta aplicação foi adaptada do trabalho de Benjamin Poppinga, *wiigee* [54]. Apresentado ao público em 2008, este trabalho consistiu na classificação de movimentos realizados com o comando da Wii, usando a linguagem Java e classificação baseada em HMM, tendo-se obtido resultados bastantes bons na classificação de gestos dinâmicos personalizados [62], [72].

### 5.4.1.1. Interface de Utilização

Nesta aplicação, o utilizador tem completo poder sobre como realizar o gesto durante o treino, não havendo restrições, bastando clicar no botão “Treinar”. Após o treino do gesto clicando em “Parar Treinar” o treino fica gravado (ver Figura 5.12). Quanto maior a quantidade de treinos executados, maior a probabilidade do gesto ser reconhecido com sucesso. Poppinga aconselha no *site* oficial a executar um conjunto de 15 treinos para melhor performance do algoritmo [54]. O código foi adaptado para correr em Unity3D tendo-se usado os algoritmos de filtragem de informação e o algoritmo de classificação da *wiigee*, também adaptados.



Figura 5.12: Interface de apresentação da aplicação.



Ao clicar “Gravar Gestor”, é apresentada uma janela em que é esperado que o utilizador clique em “Ok” para gravar o gesto e dar início ao treino de um modelo HMM com tantas sequências de observação como o conjunto de treino realizado pelo utilizador (ver Figura 5.13). Nesta fase são guardados num ficheiro .txt as informações do gesto personalizado com o nome dado pelo utilizador a cada um dos gestos. Se der um nome diferente a cada um dos gestos é guardada a informação em ficheiros diferentes, para mais tarde se incluir em uma base de dados.

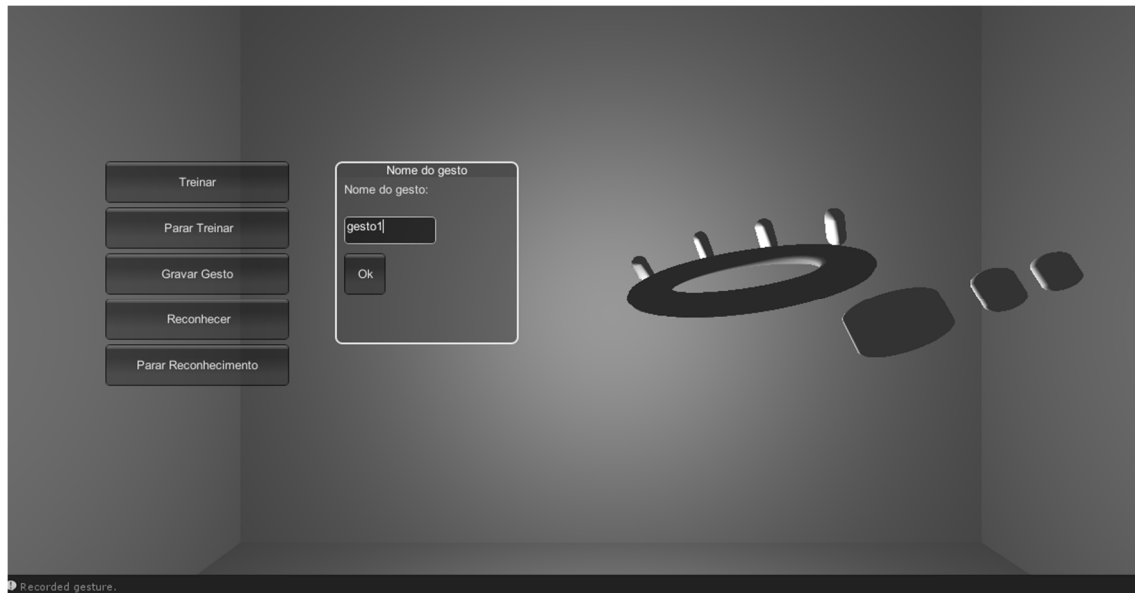


Figura 5.13: Interface apresentada após clicar “Gravar Gestor”.

Após guardar o gesto existe a possibilidade de fazer o reconhecimento de gestos. Clicando no botão “Reconhecer”, o gesto é capturado pelo Leap Motion e guardado para quando se clicar em “Parar Reconhecimento” se executar o algoritmo de classificação e enviar para a consola a identificação (ou não) do gesto (ver Figura 5.14 e Figura 5.15).

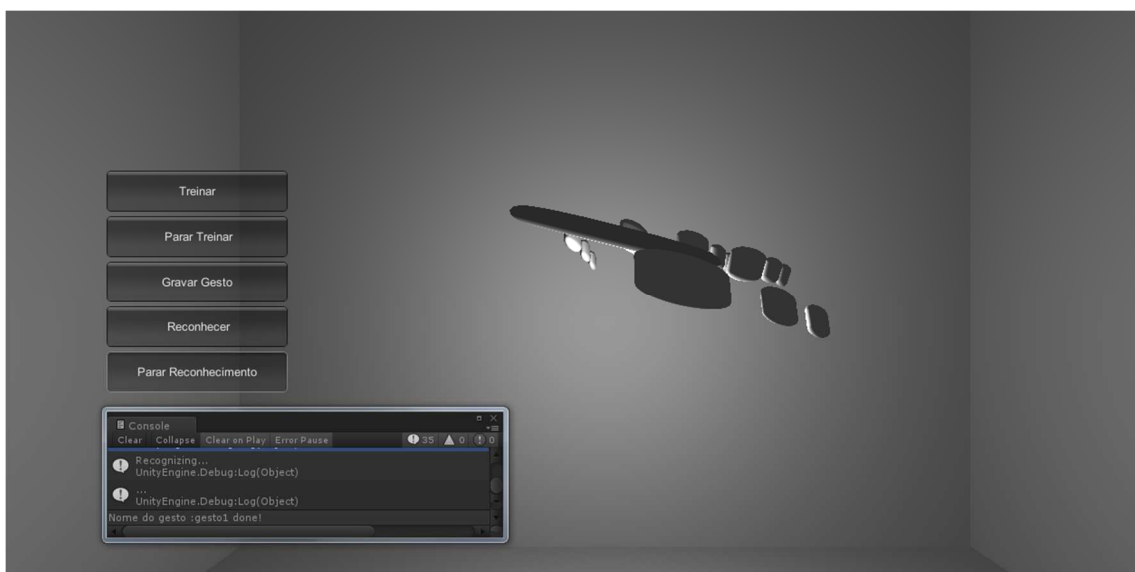


Figura 5.14: Reconhecimento de gestos (início).

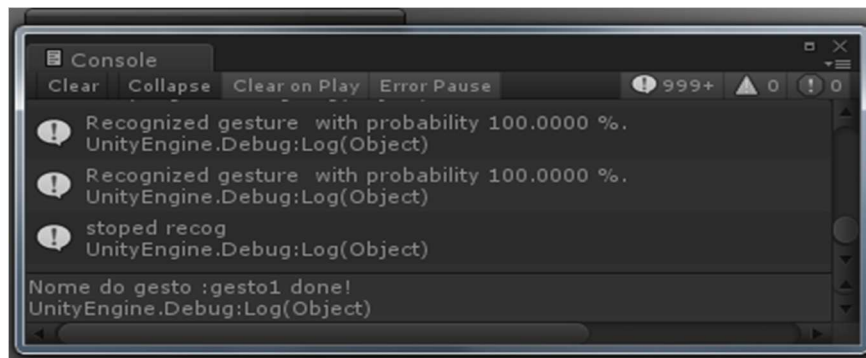


Figura 5.15: Exemplo do reconhecimento positivo de um gesto.

### 5.4.1.2. Funcionamento da Aplicação

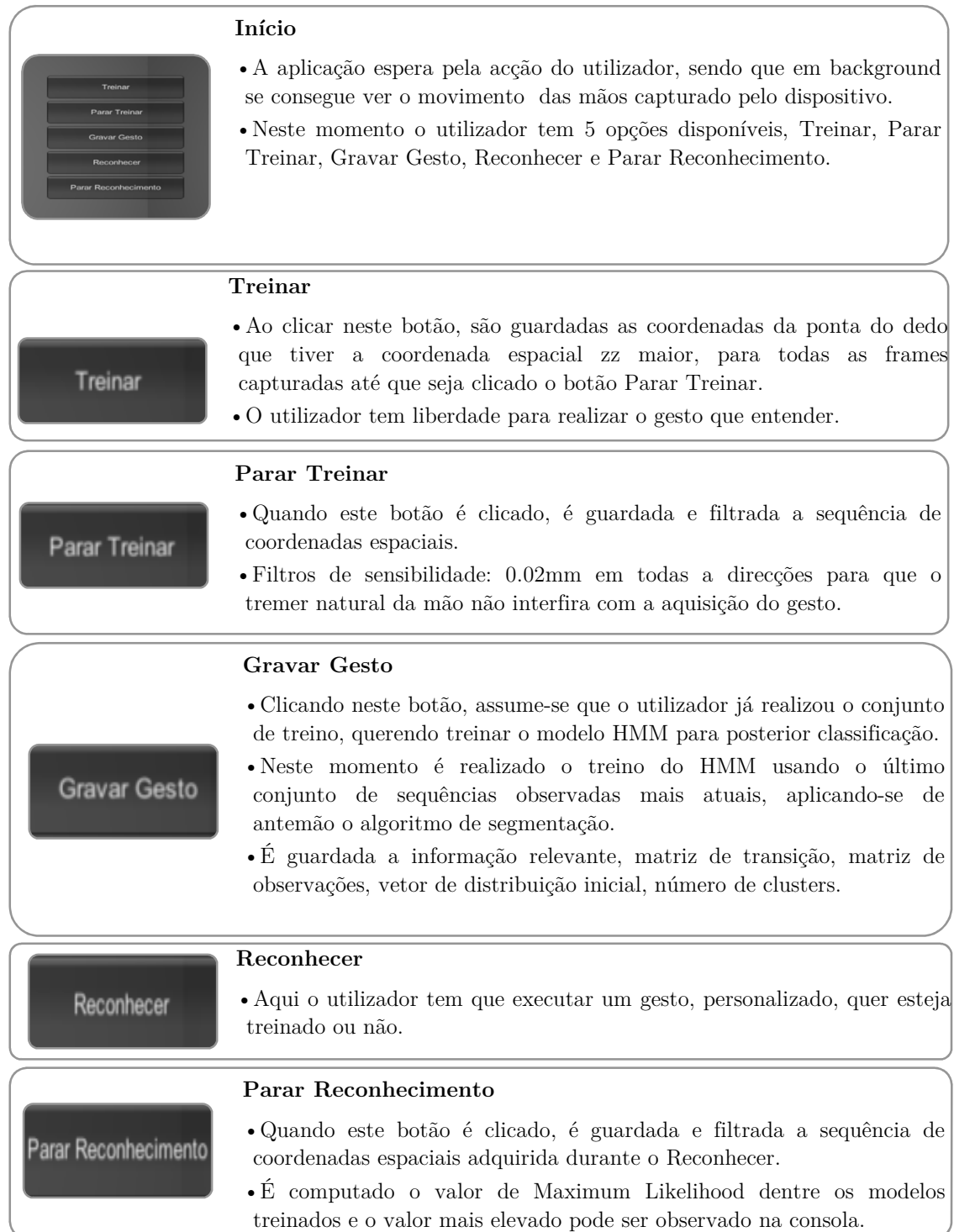


Tabela 5.5: Esquema demonstrativo do funcionamento da aplicação Classificador de 1 Gesto.

De cada vez que é adquirida uma sequência é feita uma filtragem de sensibilidade para que a movimentação natural da mão não interfira com os dados, visto que o dispositivo

tem uma precisão média de detecção de movimento de 1.2mm e desvio padrão é aproximadamente de 0.5mm [47], [48].

Uma motivação extra para testar a hipótese de reconhecimento e classificação de gestos com base na *wii* foi o fato de ser possível encontrar neste repositório [73] uma ideia semelhante adaptada para C# usando também o dispositivo da Leap Motion, Inc..

## 6. Resultados

Para testar as metodologias do capítulo anterior foi usado, para testes um conjunto de 50 gestos de 5 tipos em cada uma das aplicações explicadas no capítulo anterior. Os gestos nas primeiras três aplicações são baseados na numeração contada em LGP, por simplicidade do gesto. A aplicação Classifica 1 Gesto será testada com as letras “a”, “b” e “c” em LGP.

Os testes foram baseados em performance, robustez, fiabilidade dos dados obtidos com base nas fórmulas de precisão, sensibilidade e valor preditivo positivo de M. Sokolova et al. “A systematic analysis of performance measures for classification tasks.” [74] que se seguem:

$$\text{Valor Preditivo Positivo} = \frac{TP}{TP + FP}$$

$$\text{Sensibilidade} = \frac{TP}{TP + FN}$$

$$\text{Precisão} = \frac{TP + TN}{TP + TN + FP + FN}$$

- TP – Verdadeiros Positivos, é o número de vezes que o classificador reconhece o gesto corretamente à sua classe;
- TN – Verdadeiros Negativos, é o numero de vezes que o classificador identifica corretamente o gesto como pertencente a outra classe que não aquela que se está a analisar;
- FP – Falsos Positivos, é o número de vezes que o classificador identifica o gesto incorretamente, sendo o correto pertencente a outra classe;

- FN – Falsos Negativos, é o número de vezes que o classificador não reconhece o gesto como pertencente a uma das classes definidas.

Os dados obtidos correspondem a uma classificação multi-classe, e a performance do classificador será avaliada pelo valor médio da Precisão [65], [74]. Cada aplicação foi testada em 2 sujeitos.

## 6.1. LeapTrainer UI

Para teste, optou-se por fazer cinco gestos estáticos ou poses que representavam a numeração de um a cinco. Para gestos dinâmicos testou-se a fluidez dos gestos para representar as letras “a”, “b” e “c” e a numeração de um a cinco. Nos gestos estáticos, repetiu-se cada gesto 10 vezes, num total de cinquenta gestos. Registrou-se o sucesso com que o gesto foi bem identificado na matriz confusão que se apresenta em seguida:











		Gesto reconhecido					
							Total
Gesto realizado		7	2	1	0	0	10
		1	8	1	0	0	10
		0	1	9	0	0	10
		0	0	0	7	3	10
		0	0	0	0	10	10
Total		8	10	10	7	13	50

Tabela 6.1: Matriz – Confusão para a aplicação LeapTrainerUI

Calculou-se o Valor Preditivo Positivo, a Sensibilidade para cada classe e a Precisão do classificador, obtendo-se:


					
<b>Valor Preditivo Positivo</b>	0,88	0,80	0,90	1,00	0,77
<b>Sensibilidade</b>	0,70	0,80	0,90	0,70	1,00
<b>Precisão</b>	0,82				

Tabela 6.2: Análise do classificador do LeapTrainerUI

Como se pode observar na tabela 6.1, os gestos “5” e “3” foram os que mais identificações positivas obtiveram, com 10 e 9 identificações verdadeiras, respetivamente

Analisando a tabela 6.2 pode-se aferir que o classificador tem um Valor Preditivo Positivo bastante razoável, uma vez que prevê cada gesto, corretamente, com probabilidades acima dos 77%. Em termos de sensibilidade é sempre superior a 70%. Na perspetiva global, o classificador tem uma percentagem de precisão de 82%, que é um valor bastante razoável.

Esta análise avalia apenas a capacidade do classificador identificar corretamente poses, isto é gestos estáticos, uma vez que é incapaz de analisar gestos dinâmicos, para além do *Swipe*.

## 6.2. Reconhecimento de Numeração

Esta aplicação foi implementada de forma a conseguir classificação de gestos dinâmicos. Optou-se novamente pelo processamento da numeração cardinal de 1 a 5 em LGP. Com esta abordagem foi possível classificar os cinco gestos dinâmicos em tempo real. Contudo, não foi testado qualquer tipo de gestos estáticos uma vez que esta aplicação está direcionada para o dinamismo do gesto.

Foi realizado um conjunto total de 50 gestos divididos por 5 classes (5 números) e observou-se a resposta do classificador, que foi anotada na tabela 6.3, em baixo:











		Gesto reconhecido					
							Total
Gesto realizado		3	2	5	2	1	10
		1	6	2	1	0	10
		0	0	7	3	0	10
		0	0	2	8	0	10
		0	0	0	0	10	10
Total		4	8	16	14	11	50

Tabela 6.3: Matriz – Confusão para a aplicação Reconhecimento de Numeração

Para este caso, os melhores resultados foram obtidos para o gesto “5” e “4”, com 10 e 8 contagens positivas, respetivamente.

Calculou-se para estes dados o Valor Preditivo Positivo, a Sensibilidade para cada uma das classes, e a Precisão do classificador (ver Tabela 6.4).






					
Valor Preditivo Positivo	0,75	0,75	0,44	0,57	0,91
Sensibilidade	0,30	0,60	0,70	0,80	1,00
Precisão	0,68				

Tabela 6.4: Análise do classificador da aplicação Reconhecimento de Numeração



Analisando a Tabela 6.4 conclui-se que também este classificador tem um Valor Preditivo Positivo também bastante razoável para os gestos “1”, “2” e “5”. Em contrapartida, nos gestos “3” e “4” deixam muito a desejar, com valores preditivos positivos menores que 57%. A sensibilidade é bastante variável, uma vez que o gesto “1” tem uma probabilidade de ser classificado erradamente em 70% dos casos, mas para os outros 4 gestos a probabilidade máxima de errar a classificação é inferior ou igual a 40%.

Na perspectiva global, o classificador tem uma precisão de 68%.

### **6.3. Reconhecimento de Numeração com HMM**

Nesta aplicação testou-se a capacidade do classificador em reconhecer a numeração cardinal até 5 em LGP, havendo uma tentativa de melhorar os resultados da classificação anterior com base em um classificador de HMM.

Foi realizado um conjunto total de 50 gestos divididos por 5 classes (5 números) e observou-se a resposta do classificador, que foi anotada na Tabela 6.5, em baixo:











		Gesto reconhecido					
							Total
Gesto realizado		5	2	2	1	0	10
		1	6	3	0	0	10
		0	0	7	3	0	10
		0	0	1	8	1	10
		0	0	0	0	10	10
Total		6	8	13	12	11	50

Tabela 6.5: Matriz – Confusão para a aplicação Reconhecimento de Numeração – Accord Framework

Também para este caso, os melhores resultados foram obtidos para os gestos “5” e “4”, com 10 e 8 contagens corretas, respetivamente.

Calculou-se para estes dados, o Valor Preditivo Positivo, a Sensibilidade para cada uma das classes e a Precisão do classificador (ver Tabela 6.6).






					
<b>Valor Preditivo Positivo</b>	0,83	0,75	0,54	0,67	0,91
<b>Sensibilidade</b>	0,50	0,60	0,70	0,80	1,00
<b>Precisão</b>	0,72				

Tabela 6.6: Análise do classificador da aplicação Reconhecimento de Numeração – Accord Framework

Analisando a Tabela 6.6 conclui-se que houve melhoras na classificação em relação ao anterior classificador. Tem um Valor Preditivo Positivo acima dos 75% nos gestos “1”, “2”, “5”, melhorando a precisão nos gestos “3” e “4”, com probabilidade de errar a previsão de 46% e 33%.

Quanto à sensibilidade, no gesto “1” falha a classificação metade das vezes tentadas, mas acerta todas quando se trata do gesto “5”.

Na perspectiva global, o classificador tem uma precisão de 72%, valor bastante razoável para a classificação de gestos dinâmicos.

## 6.4. Classificador de 1 Gesto

Esta aplicação foi testada com as letras do alfabeto manual “a”, “b” e “c” em LGP, executando um conjunto de 15 treinos para cada gesto. Como esta aplicação ainda está em desenvolvimento, só classifica 1 vez por cada execução do programa. Portanto os resultados obtidos serão baseados em Sucesso/Erro. Foram executados 5 testes para cada gesto e registada a resposta do classificador na Tabela 6.7:




	Gesto Testado		
			
Acertos	1	2	0

Tabela 6.7: Resultados obtidos da aplicação Classificador 1 Gesto

Como se pode observar pela Tabela 6.7, para o gesto “a” apenas reconheceu 1 em 5 tentativas, classificou o gesto “b” 2 das 5 vezes e não conseguiu reconhecer o gesto “c” nas 5 tentativas que foram feitas. No total, este classificador teve um sucesso de 3 tentativas em 15 o que perfaz 20% de precisão na classificação, valor bastante baixo.

Estes resultados serão discutidos no capítulo que se segue, justificando estes resultados.



# 7. Discussão dos Resultados

Neste capítulo serão discutidos os resultados apresentados no capítulo 6. Começar-se-á por discutir os resultados da aplicação LeapTrainerUI, seguindo para a discussão dos resultados das aplicações Reconhecimento de Numeração com e sem HMM, finalizando com a aplicação Classifica 1 Gesto.

## 7.1. LeapTrainerUI

Idealmente desejar-se-ia ter usado, como referido no início das abordagens, o *Raw Data*, visto que analisando a imagem em bruto seria possível fazer um processamento muito mais adequado às necessidades do estudo. Como tal não era possível na altura em que se estava a realizar a dissertação, enveredou-se por outros caminhos, onde se faz a análise de dados fornecidos pelo Leap Motion, nomeadamente coordenadas espaciais e velocidades dos dedos e palma da mão.

Esta abordagem, LeapTrainerUI, foi o passo inicial para se compreender como se comportava o dispositivo no reconhecimento de gestos para além dos que já vêm com o SDK do dispositivo. Esta abordagem como usava as coordenadas da ponta dos dedos e da palma da mão, conseguiu-se tirar conclusões a nível da performance do dispositivo na aquisição desses dados.

Verificou-se nesta fase que o reconhecimento de gestos dinâmicos era difícil com o dispositivo. Para ter o reconhecimento de um gesto dinâmico esta plataforma só reconhece o swipe que tem de ser executado rapidamente mas sem se perder a localização que o Leap faz. A nível da classificação de gestos estáticos/poses, percebeu-se que o Leap apresenta uma boa consistência quando a mão se mantém parada. Isto é derivado de quando a mão

está parada é feita uma análise da posição e número de dedos no campo de visão. No entanto, é possível verificar pelos resultados que muitas vezes o dispositivo confunde dedos entre *frames*, o que levou a que no reconhecimento de numeração por vezes confundisse números. A nível geral para gestos estáticos/poses o dispositivo apresenta uma boa sensibilidade com valores superiores a 70%.

Apesar do dispositivo apresentar uma precisão nominal elevada de 82%, este tem dificuldade em manter esta precisão conforme a mão se vai movimentando ao longo dos eixos. Por conseguinte, a análise direta feita com coordenadas sejam elas de posição ou de velocidade é lenta e nem sempre realista. Este facto foi constatado por uma equipa de investigadores da Universidade de Griffith, que analisaram a viabilidade do Leap Motion para um sistema de tradução de Língua Gestual Australiana [75].

O facto de se ter verificado que o dispositivo apresentava boa performance em gestos estáticos em vez dos dinâmicos podia ser um problema para a realização do nosso objetivo. No entanto foi visto como mais um desafio a superar e criar uma abordagem que ultrapassasse esta limitação, visto que este dispositivo apresenta bastantes potencialidades, apenas necessitam de ser desenvolvidas. Assim sendo, partiram-se para abordagens de análise de gestos dinâmicos.

## 7.2. Reconhecimento de Numeração com e sem HMM

Seguindo os resultados da abordagem anterior, nesta fase começou-se pelo simples, desenvolvendo duas aplicações para reconhecimento e classificação de numeração cardinal em LGP.

Na primeira aplicação experimentada foi feita uma classificação direta com base na variação de coordenadas da ponta dos dedos e palma da mão. A nível de resultados verificou-se que com a análise direta da variação das coordenadas sem aplicação de limiar é difícil manter a consistência dos dados, uma vez que mesmo com a mão parada existe uma variação de coordenadas em todos os dedos, mesmo que essa variação seja mínima. Este facto é suportado pela literatura, encontrando-se no artigo “*An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking*” [48] dados que fundamentam esta conclusão. Investigadores afirmam que em

gestos dinâmicos, quando o utilizador movimenta a mão existe um erro associado de 25 mm em cada coordenada. Esta falta de consistência é sentida principalmente no eixo zz, o principal eixo onde existe variação do gesto dinâmico.

Assim sendo, com base nestes resultados, foi implementado um limiar na variação das coordenadas em 30 mm. Com isto conseguiu-se precaver o erro associado de 25 mm inerentes à aquisição de gestos dinâmicos pelo dispositivo. Como foi possível observar na Tabela 6.4, foi assim possível fazer a classificação de numeração cardinal. No entanto, verifica-se que o dispositivo ainda confunde a posição de cada dedo e entre dedos levando muitas vezes a uma interpolação de dados incorreta. Isto é mais acentuado quando se classifica numeração entre 1 e 2, em que o dispositivo força a interpolação dos restantes dedos levando a apresentar uma precisão e uma sensibilidade menores nestes casos.

Com isto, verifica-se que não basta usar os dados de saída do dispositivo, mas sim é necessário implementar um modelo em *machine-learning*, como foi feito em “*Gesture Recognition Library for Leap Motion Controller*” [16]. Esta equipa conseguiu uma boa classificação nos gestos estáticos, havendo uma redução acentuada nos resultados quando passam para os gestos dinâmicos.

Para melhorar os resultados, optou-se por fazer a classificação com base em HMM de forma a tentar contornar a interpolação feita pelo Leap Motion. Com o treino de 15 vezes em cada gesto, e fazendo a classificação nas 10 sequências seguintes ao treino, é possível reduzir o erro associado a cada um. Com isto conseguiu-se obter uma melhor classificação da numeração mesmo nos números mais baixos, levando a um aumento ligeiro na sensibilidade da classificação dos gestos.

Visto que se conseguiu um ligeiro aprimoramento dos resultados da classificação com base em HMM, quis-se continuar o estudo deste caminho. Nesta perspetiva avançou-se nas abordagens para uma classificação mais complexa que a numeração cardinal.

Como dito anteriormente, esta nova abordagem foi baseada no trabalho “*Gesture Recognition with a Wii Controller*” [62]. Neste trabalho, foi desenvolvida uma ferramenta, a *wiigee* [54], onde o autor faz um processamento de dados que nos pareceu adequado para a situação, nomeadamente o processamento para eliminar o tremer natural da mão.

## 7.3. Classificador de 1 Gesto

Nesta abordagem, foi experimentada uma classificação do gesto com base na variação da velocidade da ponta dos dedos. Optou-se por testar esta abordagem e observar se o dispositivo encontrava concordâncias entre as velocidades do gesto.

Os resultados obtidos com este método não se revelaram tão bons como o seria de esperar. Quando se iniciou este método pensou-se que o pré-processamento de dados feitos iria de certa forma levar a um melhoramento dos dados fornecidos ao classificador. No entanto, tal não se verificou. O HMM adaptado do trabalho *wiigee* apresentou dificuldades em encontrar concordâncias entre os gestos. Quando gravávamos dois ficheiros .txt com o mesmo tipo de gesto, fosse ele a letra “a”, “b” e “c”, verificou-se que existiam fortes discrepâncias entre os dados. Isto revela que os dados dinâmicos captados pelo dispositivo, mesmo pré-processados, apresentam bastante ruído de fundo e distorção. Ora, sendo complicado encontrar concordância entre os dados, torna-se difícil o algoritmo de HMM estabelecer um padrão entre os 15 treinos de cada gesto, sendo assim difícil de reconhecer o gesto.

Este método de todos foi o que mais desapontou, sendo que no início era considerado como um dos mais prometedores para conseguir uma classificação gestual, dado que no trabalho *wiigee* tinham conseguido obter resultados bastante razoáveis com uma taxa de reconhecimento superior a 84%.

Isto leva-nos a crer que outro caminho para análise de dados tem que ser seguido com o dispositivo. O Leap Motion é um dispositivo bastante prometedore mas que ainda se encontra em desenvolvimento. O constante lançamento de novas versões do SDK é a prova disso. Como dito anteriormente, em Agosto foi libertado o acesso ao Raw Data para os developers, indo de encontro à primeira abordagem proposta. Isto irá ser certamente uma mudança na forma de como os dados adquiridos pelo Leap Motion são processados. No entanto nesta dissertação não houve tempo para testar essa hipótese com grande pena, mas será abordada no trabalho futuro, uma vez que se acredita que este dispositivo tem potencialidades, apenas têm que ser melhor exploradas.



## 8. Conclusão

Serão expostas neste capítulo as limitações, perspectivas futuras e conclusões deste trabalho.

### 8.1. Conclusões

Este projeto foi um desafio devido a usar tecnologia tão recente como é a do Leap Motion mas ao mesmo tempo motivador uma vez que pode vir a melhorar a vida da comunidade surda, sendo um desenvolvimento social inovador.

Espera-se, com este trabalho, abrir o caminho para a criação do primeiro tradutor pessoal de LGP, eliminando o fator de exclusão social da comunidade surda.

De um modo geral verificou-se que a tecnologia Leap Motion tem potencialidades para ser usado como base de tradutor. Com o desenvolvimento certo e a aplicação das novas SDK, será certamente aberto um novo caminho na compreensão da Língua Gestual.

Este trabalho enriqueceu-me como pessoa, permitiu-me ter contacto com uma realidade diferente, a realidade da surdez. As pessoas surdas não são diferentes, apenas necessitam de pequenos ajustes ao seu meio envolvente, como por exemplo em vez de terem campainhas, estas são substituídas por lâmpadas de cor forte, chamativas. No entanto, continua a existir um fosso ao nível da comunicação entre surdos e ouvintes não especialistas em LGP. É esse fosso que pretendo ajudar a eliminar.

## 8.2. Limitações

A maior limitação deste trabalho foi a falta de robustez dos dados adquiridos pelo dispositivo. Poder-se-ia ter contornado esta limitação se fosse disponibilizado o acesso ao *Raw Data* do aparelho dentro do tempo limite de trabalho de dissertação. Sendo assim o tempo e privação do acesso ao *Raw Data* foram também limitações deste trabalho, no entanto uma vez que o trabalho vai continuar ao abrigo do apoio do “Passaporte para o Empreendedorismo”, haverá hipótese de a curto prazo ultrapassar esta limitação.

O modelo estatístico Hidden Markov revelou ser inadequado para o tipo de dados que o dispositivo adquire. O escasso número de treinos e a complexidade da análise dos dados obtidos pelo Leap Motion leva a uma baixa eficácia na classificação, sendo portanto limitação.

A falta de coerência, isto é o mesmo gesto executado várias vezes seguidas da mesma forma gera outputs diferentes, limita a padronização de gestos mais complexos.

A constante evolução do SDK do dispositivo, também foi uma limitação, uma vez que o SDK mais recente revelava-se incompatível com o anterior.

## 8.3. Perspetivas Futuras

Este projeto será continuado no futuro, uma vez que a temática abordada é bastante recente com uma constante evolução da tecnologia usada e porque foi vencedor do concurso “Passaporte para o Empreendedorismo” e passou também à 2<sup>a</sup> fase de avaliação do concurso “Fraunhofer Portugal Challenge 2014”.

Como tal, perspetiva-se desenvolver o projeto com a análise do *Raw Data* com recurso ao OpenCV, por ser o próximo passo a executar de forma a contornar as limitações encontradas. Modificar o modelo estatístico de Hidden Markov para Redes neuronais e o uso de bases de dados para guardar os dados do utilizador serão os pontos-chave que futuramente terão de ser desenvolvidos. A criação de bibliotecas de Línguas Gestuais variadas também direcionadas para a perspetiva da pessoa surda como turista noutros países será outra perspetiva a considerar.

Está também perspectivado o desenvolvimento de uma plataforma lúdica de forma a auxiliar na aprendizagem da LGP, uma vez que terá especial relevância no casos de famílias com filhos e pais ouvintes, facilitando a aprendizagem tanto dos pais como dos filhos.



## 9. Bibliografia

- [1] J. Paço, C. Branco, I. Moreira, C. Carocha, and M. M. Henriques, *Introdução à Surdez*. Lisboa: Universidade Católica Editora, Unipessoal, 2010.
- [2] “Perda da audição e surdez.” [Online]. Available: <http://www.manualmerck.net/?id=236>. [Accessed: 06-Aug-2014].
- [3] A. Kochhar, M. Hildebrand, and R. Smith, “Clinical aspects of hereditary hearing loss,” *Genet. Med.*, vol. 9, no. 7, pp. 393–408, Jul. 2007.
- [4] WHO, “Deafness and hearing loss,” *Fact sheet N°300*, 2013. [Online]. Available: <http://www.who.int/mediacentre/factsheets/fs300/en/>. [Accessed: 17-Sep-2014].
- [5] SIL International Publications, “Portuguese Sign Language.” [Online]. Available: <http://www.ethnologue.com/language/psr>. [Accessed: 07-Aug-2014].
- [6] “DBpedia.” [Online]. Available: [http://dbpedia.org/page/Portuguese\\_Sign\\_Language](http://dbpedia.org/page/Portuguese_Sign_Language). [Accessed: 17-Sep-2014].
- [7] “Constituição Portuguesa.” [Online]. Available: <http://www.parlamento.pt/Legislacao/Paginas/ConstituicaoRepublicaPortuguesa.aspx>. [Accessed: 17-Sep-2014].
- [8] “Associação Portuguesa de Surdos.” [Online]. Available: [http://www.apsurdos.org.pt/index.php?option=com\\_content&view=article&id=41&Itemid=8](http://www.apsurdos.org.pt/index.php?option=com_content&view=article&id=41&Itemid=8). [Accessed: 17-Sep-2014].
- [9] C. Gomes, “A Reconfiguração Política da Surdez e da Educação de Surdos em Portugal: Entre os Discursos Identitários e os Discursos de Regulação,” Tese de Doutoramento, Faculdade de Psicologia e Ciências da Educação, Universidade do Porto, 2012.

- [10] “Cronologia e História da Surdez e da Língua Gestual no Mundo - porsinal, consegues ouvir o Mundo ?” [Online]. Available: <http://www.porsinal.pt/index.php?ps=historia>. [Accessed: 07-Sep-2014].
- [11] R. da Silva, “SignWriting: um sistema de escrita das línguas gestuais. Aplicação à Língua Gestual Portuguesa,” *Exedra Rev. Científica*, pp. 387–399, 2012.
- [12] O. Oliveira, “Tradutor da Língua Gestual Portuguesa - Modelo de Tradução Bidirecional,” Tese de Mestrado, Instituto Superior de Engenharia do Porto, 2014.
- [13] M. Correia, “Reconhecimento de elementos da língua gestual portuguesa com Kinect,” Tese de Mestrado, Faculdade de Engenharia, Universidade do Porto, 2012.
- [14] “Site Oficial ATILGP - Associação de Tradutores e Intérpretes de Língua Gestual Portuguesa.” [Online]. Available: <http://www.atilgp.com/home.html>. [Accessed: 15-Sep-2014].
- [15] Q. Chen, “Real-time vision-based hand tracking and gesture recognition,” Tese de Doutoramento, Faculty of Engineering, University of Ottawa, 2008.
- [16] M. Nowicki, O. Pilarczyk, J. Wąsikowski, and K. Zjawin, “Gesture Recognition Library for Leap Motion Controller,” Bachelor’s Thesis, Faculty of Computing, Poznan University of Technology, 2014.
- [17] M.-H. Yang and N. Ahuja, *Face Detection and Gesture Recognition for Human-Computer Interaction*, vol. 1. Boston, MA: Springer US, 2001.
- [18] “Homepage | cyberglovesystems.com.” [Online]. Available: <http://www.cyberglovesystems.com/>. [Accessed: 24-Aug-2014].
- [19] “Myo - Gesture control armband by Thalmic Labs.” [Online]. Available: <https://www.thalmic.com/en/myo/>. [Accessed: 24-Aug-2014].
- [20] “IGS-190 (Gypsy Gyro) Inertial Motion Capture System.” [Online]. Available: <http://www.metamotion.com/gypsy/gypsy-gyro.htm>. [Accessed: 24-Aug-2014].
- [21] “Mi.mu Gloves.” [Online]. Available: <http://mimu.org.uk/>. [Accessed: 24-Aug-2014].
- [22] “Mister Gloves - A Wireless USB Gesture Input System.” [Online]. Available: [https://courses.cit.cornell.edu/ee476/FinalProjects/s2010/ssc88\\_egl27/](https://courses.cit.cornell.edu/ee476/FinalProjects/s2010/ssc88_egl27/). [Accessed: 24-Aug-2014].
- [23] “The Power of Glove.” [Online]. Available: <http://thepowerofglove.com/>. [Accessed: 24-Aug-2014].

- [24] “Enable Talk project.” [Online]. Available: <http://enabletalk.com/>. [Accessed: 24-Aug-2014].
- [25] J. Kim, S. Mastnik, and E. André, “EMG-based hand gesture recognition for realtime biosignal interfacing,” *Proc. 13th Int. Conf. Intell. user interfaces - IUI '08*, p. 30, 2008.
- [26] O. Kainz and F. Jakab, “Approach to Hand Tracking and Gesture Recognition Based on Depth-Sensing Cameras and EMG Monitoring,” *Acta Inform. Pragensia*, vol. 3, no. 1, pp. 104–112, 2014.
- [27] “Wii | Nintendo.” [Online]. Available: <http://www.nintendo.pt/Wii/Wii-94559.html>. [Accessed: 09-Sep-2014].
- [28] Q. Pu, S. Gupta, S. Gollakota, and S. Patel, “Whole-home gesture recognition using wireless signals,” *Proc. 19th Annu. Int. Conf. Mob. Comput. Netw.*, pp. 27–38, 2013.
- [29] “Exame Informática - Notícias.” [Online]. Available: [http://exameinformatica.sapo.pt/noticias/ciencia/2014-09-04-Investigadores-criam-interface-touchless-para-smartphone?utm\\_source=newsletter&utm\\_medium=mail&utm\\_campaign=newsletter&utm\\_content=2014-09-04](http://exameinformatica.sapo.pt/noticias/ciencia/2014-09-04-Investigadores-criam-interface-touchless-para-smartphone?utm_source=newsletter&utm_medium=mail&utm_campaign=newsletter&utm_content=2014-09-04). [Accessed: 08-Sep-2014].
- [30] Y. Wu and T. Huang, “Vision-based gesture recognition: A review,” in *Gesture-Based Communication in Human-Computer Interaction*, S.-V. B. Heidelberg, Ed. France: Springer Berlin Heidelberg, 1999, pp. 103–115.
- [31] “Hold Your Hand Up to Play Some Music | MIT Technology Review.” [Online]. Available: <http://www.technologyreview.com/news/427400/hold-your-hand-up-to-play-some-music/>. [Accessed: 14-Sep-2014].
- [32] “Flutter | Gesture Recognition on Your Webcam.” [Online]. Available: <https://flutterapp.com/about/>. [Accessed: 14-Sep-2014].
- [33] “Xbox Kinect | Full Body Gaming and Voice Control.” [Online]. Available: <http://www.xbox.com/en-US/kinect>. [Accessed: 24-Aug-2014].
- [34] “LazeeEye: Turn Your Smartphone Into a 3D Camera :: Kicktraq.” [Online]. Available: <http://www.kicktraq.com/projects/332648737/lazeeeye-turn-your-smartphone-into-a-3d-camera/>. [Accessed: 09-Sep-2014].
- [35] “Intel® RealSense™ Technology.” [Online]. Available: <http://www.intel.com/content/www/us/en/architecture-and-technology/realsense-overview.html>. [Accessed: 24-Aug-2014].

- [36] “Leap Motion | Mac & PC Motion Controller for Games, Design, & More.” [Online]. Available: <https://www.leapmotion.com/>. [Accessed: 09-Sep-2014].
- [37] “PlayStation®Move Motion Controller - PlayStation®3 Move Info, Games & Updates.” [Online]. Available: <http://us.playstation.com/ps3/playstation-move/>. [Accessed: 09-Sep-2014].
- [38] “How the Deaf Will Communicate - Signing Comes to the World Wide Web.” [Online]. Available: <http://www.21stcentech.com/communications-21st-century-part-4-deafening-speech-capability-deaf/>. [Accessed: 12-Sep-2014].
- [39] “Microsoft Speech Platform.” [Online]. Available: [http://msdn.microsoft.com/en-us/library/office/hh361572\(v=office.14\).aspx](http://msdn.microsoft.com/en-us/library/office/hh361572(v=office.14).aspx). [Accessed: 12-Sep-2014].
- [40] “Bing Developer Center.” [Online]. Available: <http://www.bing.com/dev/en-us/dev-center>. [Accessed: 12-Sep-2014].
- [41] “Smart Gloves Turn Sign Language Gestures Into Vocalized Speech | Singularity Hub.” [Online]. Available: <http://singularityhub.com/2012/09/16/smart-gloves-turn-sign-language-gestures-into-vocalized-speech/>. [Accessed: 12-Sep-2014].
- [42] A. Kar, “Skeletal tracking using microsoft kinect,” *Methodology*, pp. 1–11, 2010.
- [43] X. Chai, G. Li, Y. Lin, Z. Xu, Y. Tang, X. Chen, and M. Zhou, “Sign Language Recognition and Translation with Kinect,” *Vipl.Ict.Ac.Cn*, 2013.
- [44] X. Chen, T. Pan, and S. Tansley, “Kinect Sign Language Translator.”
- [45] “FCT — Projectos de I&D — Consultas à base de dados.” [Online]. Available: [http://www.fct.pt/apoios/projectos/consulta/vglobal\\_projecto?idProjecto=121878&idElemConcurso=4231](http://www.fct.pt/apoios/projectos/consulta/vglobal_projecto?idProjecto=121878&idElemConcurso=4231). [Accessed: 12-Sep-2014].
- [46] “Virtual Sign vai revolucionar a inclusão digital de pessoas surdas - porsinal, consegue ouvir o Mundo ?” [Online]. Available: <http://www.porsinal.pt/index.php?ps=destaques&idt=not&iddest=224>. [Accessed: 12-Sep-2014].
- [47] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler, “Analysis of the accuracy and robustness of the leap motion controller,” *Sensors (Basel)*, vol. 13, no. 5, pp. 6380–93, Jan. 2013.
- [48] J. Guna, G. Jakus, M. Pogačnik, S. Tomažič, and J. Sodnik, “An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking,” *Sensors (Basel)*, vol. 14, no. 2, pp. 3702–20, Jan. 2014.



- [49] “Leap Motion Technology Translates Sign Language.” [Online]. Available: <http://www.crunchwear.com/leap-motion-technology-translates-sign-language/>. [Accessed: 12-Sep-2014].
- [50] “Home - LEAP.AXLR8R.” [Online]. Available: <http://www.leapaxlr8r.com/>. [Accessed: 12-Sep-2014].
- [51] “MotionSavvy Is A Tablet App That Understands Sign Language | TechCrunch.” [Online]. Available: <http://techcrunch.com/2014/06/06/motionsavvy-is-a-tablet-app-that-understands-sign-language/>. [Accessed: 12-Sep-2014].
- [52] “MotionSavvy.” [Online]. Available: <http://www.motionsavvy.com/>. [Accessed: 12-Sep-2014].
- [53] “MotionSavvy uses gesture recognition to give sign language a voice.” [Online]. Available: <http://www.engadget.com/2014/05/17/motionsavvy-uses-gesture-recognition-to-give-sign-language-a-vo/>. [Accessed: 12-Sep-2014].
- [54] “wiigee - a java-based gesture recognition library for the Wii remote.” [Online]. Available: <http://www.wiigee.org/>. [Accessed: 01-Sep-2014].
- [55] “Leap Trainer UI v.03.” [Online]. Available: <https://rawgit.com/robleary/LeapTrainer.js/master/trainer-ui.html>. [Accessed: 26-Aug-2014].
- [56] “Cross Correlation.” [Online]. Available: <http://paulbourke.net/miscellaneous/correlate/>. [Accessed: 14-Sep-2014].
- [57] J. O. J. Wobbrock, M. G. Hall, A. A. D. Wilson, and Y. Li, “Gestures without libraries, toolkits or training: a \$1 recognizer for user interface prototypes,” in *Visual Analysis of Humans*, London: Springer-Verlag, 2011, pp. 539–562.
- [58] “Unity - Game Engine.” [Online]. Available: <http://unity3d.com/>. [Accessed: 14-Sep-2014].
- [59] E. B. Passos and J. Ricardo, “Tutorial : Desenvolvimento de Jogos com Unity 3D,” pp. 1–30, 2009.
- [60] “Download - AltSoftLab.” [Online]. Available: <http://www.altsoftlab.com/download.aspx>. [Accessed: 30-Aug-2014].
- [61] M. Viblis and K. Kyriakopoulos, “Gesture Recognition: The Gesture Segmentation Problem,” *J. Intell. Robot. Syst.*, pp. 151–158, 2000.
- [62] T. Schlömer, B. Poppinga, N. Henze, and S. Boll, “Gesture recognition with a Wii controller,” *Proc. 2nd Int. Conf. Tangible Embed. Interact.*, pp. 11–14, 2008.

- [63] “Learning Custom Gestures in Leap Motion | Analysing Location Data.” [Online]. Available: <http://jamesmc.com/blog/testing/>. [Accessed: 14-Sep-2014].
- [64] L. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” *Proc. IEEE*, 1989.
- [65] F. Fernandes, “Sit-To-Stand movement analysis using the Kinect platform,” Tese de Mestrado, Faculdade de Engenharia, Universidade do Porto, 2012.
- [66] V. Mantyla, “Discrete hidden Markov models with application to isolated user-dependent hand gesture recognition,” *VTT Publ.*, 2001.
- [67] J. H. Kim, “An HMM-based threshold model approach for gesture recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 10, pp. 961–973, 1999.
- [68] D. Shen, “Some Mathematics for HMM Specification of an HMM model,” 2008.
- [69] G. Rigoll, A. Kosmala, and S. Eickeler, *High performance real-time gesture recognition using hidden markov models*. Springer Berlin Heidelberg, 1998, pp. 69–80.
- [70] J. Yang and Y. Xu, “Hidden markov model for gesture recognition,” 1994.
- [71] L. da S. Espindola, “Um Estudo sobre Modelos Ocultos de Markov HMM-Hidden Markov Model,” Pós-Graduação em Ciências da Computação, Faculdade de Informática, Pontifícia Universidade Católica do Rio Grande do Sul, 2010.
- [72] B. Poppinga, “Beschleunigungs-basierte 3D-Gestenerkennung mit dem Wii-Controller,” Individuelles Projekt, Department für Informatik, Universität Oldenburg, 2007.
- [73] Tesseract, “Open Leap Gestures.” [Online]. Available: <https://github.com/openleap/LeapGestures>. [Accessed: 17-Sep-2014].
- [74] M. Sokolova and G. Lapalme, “A systematic analysis of performance measures for classification tasks,” *Inf. Process. Manag.*, vol. 45, no. 4, pp. 427–437, Jul. 2009.
- [75] L. E. Potter and L. Carter, “The Leap Motion controller : A view on sign language,” pp. 1–4, 2013.
- [76] M. Stumpf, *Escrita das Línguas Gestuais*. Lisboa: Universidade Católica Editora, Unipessoal, 2011.
- [77] “ABC gestual.” [Online]. Available: <http://abcgestual.no.sapo.pt/D2.htm>. [Accessed: 08-Sep-2014].

- [78] “Intel® RealSense™ Technology—Depth Technologies.” [Online]. Available: <http://www.intel.com/content/www/us/en/architecture-and-technology/realsense-depth-technologies.html>. [Accessed: 14-Sep-2014].
- [79] “Developing with Kinect.” [Online]. Available: <http://www.microsoft.com/en-us/kinectforwindows/develop/>. [Accessed: 12-Sep-2014].
- [80] “Face Tracking.” [Online]. Available: <http://msdn.microsoft.com/en-us/library/jj130970.aspx>. [Accessed: 12-Sep-2014].
- [81] “Introduction to Kinect for Windows SDK | Kinecting for Windows.” [Online]. Available: <http://www.kinectingforwindows.com/2013/04/01/introduction-to-kinect-for-windows-sdk/>. [Accessed: 12-Sep-2014].
- [82] “Cool Kinect move: Reading sign language in real time.” [Online]. Available: <http://www.cnet.com/news/cool-kinect-move-reading-sign-language-in-real-time/>. [Accessed: 12-Sep-2014].
- [83] “API Overview — Leap Motion C# and Unity SDK v2.1 documentation.” [Online]. Available: [https://developer.leapmotion.com/documentation/skeletal/csharp/devguide/Leap\\_Overview.html](https://developer.leapmotion.com/documentation/skeletal/csharp/devguide/Leap_Overview.html). [Accessed: 14-Sep-2014].
- [84] “Leap Motion Unveils 3D Sculpting App, Previews New Tracking Software.” [Online]. Available: <http://thenextweb.com/insider/2013/11/20/leap-motion-unveils-freeform-3d-sculpting-app-previews-new-gesture-tracking-software/>. [Accessed: 07-Sep-2014].