



**isec**  
**Engenharia**

MESTRADO EM INFORMÁTICA E  
SISTEMAS

**Reconhecimento facial 2D VS 3D estudo  
comparativo**

DEFINITIVO

Autor

**Hugo Alexandre da Costa Duarte**

Orientador

**Nuno Alexandre Cid Martins**

INSTITUTO POLITÉCNICO  
DE COIMBRA

INSTITUTO SUPERIOR  
DE ENGENHARIA  
DE COIMBRA

Coimbra, novembro 2020





# isec

## Engenharia

DEPARTAMENTO DE INFORMÁTICA E SISTEMAS

### **Reconhecimento facial 2D VS 3D estudo comparativo**

Dissertação para a obtenção do grau de Mestre em Informática e Sistemas

Especialização em Desenvolvimento de Software

Autor

**Hugo Alexandre da Costa Duarte**

Orientador

**Nuno Alexandre Cid Martins**

Co-Orientador

**Pedro Alexandre Dias Martins**



## **AGRADECIMENTOS**

Em primeiro lugar, um agradecimento especial à minha esposa Paula Santos Duarte, pelo apoio, paciência e incentivo irrestrito e fundamental ao longo destes anos.

Agradeço, humildemente, ao Professor Dr. Nuno Cid Martins, pela motivação, exigência e pela sua enorme disponibilidade e empenho.

Agradeço ao Professor Dr. Pedro Martins, por todo o apoio, transmissão de conhecimento e disponibilidade prestada. Agradeço, ainda, o interesse e colaboração concedida durante a realização deste projeto.

Um agradecimento muito especial aos meus amigos por se terem prestado e voluntariado para as várias recolhas de dados e imagens que contribuíram para a criação da base de dados de faces, contribuindo diretamente para a concretização deste trabalho.

Um agradecimento igualmente especial aos meus familiares e amigos por todo o apoio, carinho e amizade com que me brindaram neste percurso e a todas as pessoas que contribuíram direta ou indiretamente para a concretização deste trabalho.



## RESUMO

Reconhecer a identidade de seres humanos a partir de imagens captadas através de câmaras digitais torna-se importante para a evolução dos sistemas de controlo de acesso, na medida em que vem auxiliar os utilizadores que fazem uso deste processo. Para competir com os sistemas atuais de controlo de acesso é necessário que estes sistemas de reconhecimento facial sejam, acima de tudo, fiáveis; caso contrário, será difícil virem a substituir os sistemas atuais de controlo de acesso através do reconhecimento de impressões digitais. Para que este seja um sistema idôneo, é necessário o uso de tecnologia tridimensional de forma a assegurar que se trata, realmente, de um ser humano e não de uma fotografia ou outro tipo de imagem não real. Esta tese apresenta um estudo comparativo de diversos algoritmos 2D e 3D, de forma a obter o melhor método de reconhecimento facial tridimensional, no qual vários métodos são implementados, testados e analisados, com recurso a dados de imagens recolhidas com o Kinect. Esta tese tem como objetivo o estudo da abordagem descrita acima, isto é, identificar os melhores métodos para o reconhecimento facial tridimensional, para uma possível utilização num sistema real. São apresentados resultados experimentais detalhados de 15 indivíduos, dos quais foram recolhidos diversos dados faciais através do sensor Kinect, que, por sua vez, são pré processados e armazenados. Posteriormente, são processados com recurso aos métodos estudados, evidenciando as diferenças de resultados entre estes, com os quais é possível comprovar a eficácia dos métodos 2D e 3D.

**Palavras-Chave:** Reconhecimento facial, 2D, 3D, PCA, LDA, QDA, K-NN, SVN, Kinect



## **ABSTRACT**

Recognizing the human being identity through images captured by a digital camera has become an important factor towards the development of access control systems, as it helps the users of this process. In order to compete with current access control systems, these facial recognition systems must be, above all, a reliable process, otherwise it will be difficult for them to replace the current systems that use fingerprint recognition. To ensure that this is an accurate system, the use of three-dimensional technology is essential to guarantee the identification of a real person and not a photography or any other non-real visual support. This thesis presents a comparative study of several 2D and 3D algorithms in order to get the best method regarding three-dimensional facial recognition, in which several methods are implemented, tested and analyzed, using images from Kinect. This thesis aims to study the approach mentioned above, that is, to identify the best methods for a three-dimensional facial recognition to be used on a real system. It is presented detailed experimental results of 15 individuals, whose facial data is collected using the Kinect sensor, which, in turn, are pre-processed and stored. Afterwards, those results are processed according to the several studied methods, showing the differences in results between them and because of that it becomes possible to prove the accuracy of 2D and 3D methods.

**Palavras-Chave:** Facial recognition, 2D, 3D, PCA, LDA, QDA, K-NN, SVN, Kinect



## ÍNDICE

<b>CAPÍTULO 1</b> .....	1
<b>INTRODUÇÃO</b> .....	1
1.1 – Enquadramento do trabalho e inserção do problema.....	1
1.2 – Objetivos .....	2
1.3 – Trabalhos relacionados .....	3
1.4 – Pipeline.....	5
1.5 – Estrutura da dissertação.....	6
<b>CAPÍTULO 2</b> .....	9
<b>Bases para um reconhecimento tridimensional</b> .....	9
2.1 – Introdução.....	9
2.2 – Formação de imagens .....	11
2.3 – Extração de características.....	14
<b>2.3.1 Histograma da orientação de gradientes (HOG)</b> .....	14
<b>2.3.2 Histograma de Vetores Normais Orientados</b> .....	19
2.4 – Classificação/Reconhecimento .....	21
<b>2.4.1 Principal Component Analysis (PCA)</b> .....	22
<b>2.4.2 Linear Discriminant Analysis (LDA)</b> .....	26
<b>2.4.3 Quadratic Discriminant Analysis (QDA)</b> .....	29
<b>2.4.4 K - Nearest Neighbors (K-NN)</b> .....	30
<b>2.4.5 Support vector machine (SVM)</b> .....	32
<b>CAPÍTULO 3</b> .....	39
<b>Aplicação e desenvolvimento de conhecimentos</b> .....	39
3.1 – Introdução.....	39
3.2 – Descrição do pipeline.....	39
3.3 – Aquisição de dados .....	42
<b>3.3.1 Sensor Kinect V2</b> .....	42
3.4 – Detetor de faces.....	45
3.5 – Registo de base de dados .....	46
3.6 – Extração de características.....	47
3.7 – Classificação .....	49
<b>CAPÍTULO 4</b> .....	51
<b>Resultados</b> .....	51
4.1 – Descrição da base de dados .....	51
4.2 – Tabela de resultados.....	51

<b>4.2.1</b>	<b><i>Extração de características usando GreyLevel</i></b> .....	<b>51</b>
<b>4.2.2</b>	<b><i>Extração de características usando RGB-D</i></b> .....	<b>60</b>
<b>4.2.3</b>	<b><i>Extração de características usando HOG</i></b> .....	<b>65</b>
<b>4.2.4</b>	<b><i>Extração de características usando HONV</i></b> .....	<b>69</b>
<b>4.2.5</b>	<b><i>Extração de características usando HOG com HONV</i></b> .....	<b>73</b>
<b>4.3 –</b>	<b>Tabela de Conclusão de resultados</b> .....	<b>77</b>
<b>Conclusão e trabalho futuro</b> .....		<b>79</b>
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b> .....		<b>81</b>

## ÍNDICE DE FIGURAS

Figura 1 Processo de reconhecimento facial usado na dissertação.....	6
Figura 2 Representação dos três metodos principais para a representação das faces humanas.....	10
Figura 3 Exemplo de formação de uma imagem RGB 2D (Richard Szeliski, 2011).....	12
Figura 4 Exemplo de calculo de distancia através da estereoscopia passive (Jorge Alves da Silva,1994).....	13
Figura 5 Tempo de propagação do sinal de luz (Fu-Pen Chiang,1979). ....	14
Figura 6 Exemplo de conversão de imagem a cores para imagem em tons de cinza.....	15
Figura 7 Exemplo da utilização do filtro de Prewitt, Canny e Sobel respetivamente. ....	16
Figura 8 Exemplo de imagem dividida em células verdes e exemplo de dois blocos possíveis a azul e vermelho. ....	17
Figura 9 (a) Exemplo de agrupamento de orientações por célula. (b) Exemplo da representação das orientações na imagem .....	18
Figura 10 (a) Exemplo de imagem RGB (b) representação em imagem da informação de profundidade .....	20
Figura 11 Representação dos angulos e vetores calculados em HONV .....	21
Figura 12 A linha tracejada no gráfico é um limite de decisão fornecido pela LDA. A linha curva é o limite de decisão resultante do método QDA. ....	30
Figura 13 exemplo de classificação do KNN com duas classes e $k = 7$ .....	31
Figura 14 Caso linear separável.....	34
Figura 15 Caso linear não separável.....	36
Figura 16 Exemplo de normalização das imagens da base de dados .....	40
Figura 17 Representação 2D das características das 80 imagens de treino dos 15 indivíduos .....	41
Figura 18 Média de todas as faces da base de dados usando o PCA .....	41
Figura 19 Representação de uma face do subespaço do PCA.....	42
Figura 20 kinect versão 2 .....	43
Figura 21 Representação das coordenadas da face detetadas com o OpenCV .....	45
Figura 22 Base de dados de faces reconhidas com o Kinect.....	47



## ÍNDICE DE TABELAS

Tabela 1 Matriz de confusão do K-NN com GreyLevel .....	52
---	----



## SIMBOLOGIA E ABREVIATURAS

- 2D – Duas dimensões
- 3D – Três dimensões
- IR – Infra-Red
- Kinect V2 – Segunda versão do sensor Kinect
- RGB – Red, Green, Blue
- RGB-D – Red, Green, Blue and depth
- SDK – Software development kit
- LDA – Linear Discriminant Analysis
- OpenCV – Open Source Computer Vision
- PCA – Principal Component Analysis
- SDK – Software Development Kit
- SVM – Suport Vector Machine
- HOG – Histogram of Oriented Gradients
- HONV – Histogram of Oriented Normal Vectors
- Depth – Profundidade
- Grey – Cinzento
- K-NN – K - Vizinhos mais próximos
- QDA – Análise quadrática discriminante



## **CAPÍTULO 1**

### **INTRODUÇÃO**

O reconhecimento da biometria facial é uma tarefa simples para o ser humano, uma vez que este é capaz de identificar facilmente uma pessoa conhecida, a partir da sua imagem facial, ainda que estejam presentes diversas condições como as variações da iluminação, que permitem a observação de apenas algumas características ou identificar apenas partes da face e, ainda, apresentando desfigurações e/ou deformações. No entanto, torna-se extremamente complexa a sua implementação num processo que envolve uma máquina pois não sabemos como o cérebro humano realiza essa tarefa computacionalmente na totalidade. Ainda assim, a tarefa de reconhecimento facial pode ser realizada automaticamente por algoritmos de visão computacional.

#### **1.1 – Enquadramento do trabalho e inserção do problema**

O reconhecimento facial é muito aplicado em sistemas de segurança, onde o reconhecimento de biometrias mais comum é o das impressões digitais e o da íris. Os sistemas de impressões digitais são sistemas de custo médio, o que é muito positivo, mas que em contrapartida não são adequados para alguns indivíduos. A impressão digital é uma parte do corpo fácil de deteriorar, como, por exemplo, em indivíduos que têm grande contacto com produtos químicos ou que têm, frequentemente, os dedos deteriorados devido a trabalhos pesados. Devemos ainda considerar os indivíduos que exercem funções em trabalho rural ou em oficinas e ainda indivíduos que fazem testes frequentes para verificar a diabetes, que resulta ser um grande número de indivíduos na sociedade. Neste sentido, torna-se necessário utilizar outro meio mais eficaz e abrangente, como por exemplo o uso de cartões RFID, que não oferecem garantias quanto a ser a própria pessoa a utilizar o seu cartão, visto ser um objeto que pode ser facilmente transmitido a outra pessoa e usado pela mesma. Os sistemas de reconhecimento da íris apresentam alta precisão, mas são sistemas com um custo muito elevado e, deste modo, não são rentáveis para a maioria dos sistemas utilizados no quotidiano.

O reconhecimento facial parece ser uma boa solução devido a este poder ser menos dispendioso do que as soluções de reconhecimento da íris e poderá também ser eficaz em indivíduos com impressões digitais danificadas, tendo um custo similar aos outros sistemas. As variações na iluminação podem prejudicar o desempenho de alguns sistemas que só funcionam em ambientes de iluminação controlada. Este tipo de variações geralmente diminuem a precisão do sistema, mas é fácil de instalar o sistema em zonas bem iluminadas. Além disso, diferentes expressões faciais podem levar a alterações no desempenho, embora apenas variações extremas

poderão levar a erros significativos no desempenho do sistema. Para tornar o sistema mais fiável é utilizado o sistema de reconhecimento facial tridimensional, já que é muito mais seguro que o reconhecimento facial simples ou bidimensional pois não será possível enganar o sistema com uma fotografia ou fotocópia. No entanto, isso eleva e origina uma complexidade do sistema.

Ocultações de partes da face, especialmente se esconder a metade superior, leva à falta de informações significativas, resultando na diminuição da taxa de sucesso do respetivo reconhecimento. Estas ocultações podem ocorrer devido a um vasto número de motivos e podem acontecer de forma ocasional ou deliberada, atendendo a acessórios ou outros fatores, tais como a religião, que poderá considerar vestes que ocultem partes da face humana, como o caso do véu islâmico ou hijab. Devido ao envelhecimento a face muda de uma forma imprevisível, tornando difícil a identificação e reconhecimento de um indivíduo, o que poderá obrigar os mesmos a efetuar vários novos registos da sua face no sistema ao longo dos anos, por exemplo. Todos estes fatores são constantes do dia-a-dia e devem ser tidos em conta aquando da construção de sistemas de reconhecimento facial.

O reconhecimento facial tridimensional tem como objetivo aperfeiçoar sistemas de reconhecimento facial bidimensionais. Os sistemas de reconhecimento facial bidimensionais apesar de robustos, apresentam algumas desvantagens que podem comprometer os seus resultados, uma vez que estes sistemas podem ser enganados com uma simples fotografia. Os sistemas faciais tridimensionais surgem como uma vantagem sobre esse aspeto negativo dos sistemas bidimensionais. Os sistemas de reconhecimento facial atuais, com base em imagens bidimensionais, e os sistemas com base em imagens tridimensionais, são ainda um grande desafio devido a variações nas condições de iluminação, à presença de ocultação de parte da face, a variações faciais de expressão, entre outros. Em ambientes reais, o desafio acaba por ser ainda maior, considerando diversos fatores que contribuem para esse aumento, como o próprio local de captura de imagens que poderá ter outros objetos que poderão ajudar a confundir o sistema.

## **1.2 – Objetivos**

Esta dissertação pretende mostrar que, através da utilização das técnicas mais adequadas, é possível implementar um sistema de reconhecimento facial tridimensional fidedigno, capaz de identificar e confirmar se um indivíduo está presente numa base de dados de imagens. Desta forma, apresentaremos um simulador de um sistema de reconhecimento onde iremos analisar quais as técnicas mais eficientes e aplicáveis a este tipo de sistemas que tem como principais objetivos serem robustos, confiáveis e baratos.

A análise efetuada e a solução final proposta irão também demonstrar que estes sistemas atualmente já se começam a tornar extremamente confiáveis e que podem ser usados para melhorar o quotidiano do ser humano em diversas situações.

### **1.3 – Trabalhos relacionados**

Os sistemas de reconhecimento facial tridimensional podem resolver quase todos estes problemas mencionados anteriormente. No entanto, devem ser sistemas que sejam concebíveis, baratos e rigorosos, para que se tornem numa alternativa viável aos sistemas de reconhecimento de impressões digitais.

Segundo Wei-Lun Chao (Wei-Lun Chao, 2010) existem diversas técnicas que podem ser utilizadas para a criação de um sistema de reconhecimento facial. A técnica mais utilizada é aquela que, a partir de uma fotografia ou imagem de vídeo, faz com que o sistema seja capaz de extrair a imagem facial da pessoa e verificar se essa pertence ou não a uma base de dados com imagens faciais previamente processadas para que possam ser comparadas.

Existem muitos métodos para detetar e localizar uma face humana numa imagem. Alguns baseiam-se em técnicas de modelos deformáveis das características da face, que através da deformação da imagem original é possível detetar os olhos e a boca (A.L. Yuille et al., 1992). Wait et al. (Wait et al., 1990) descreve um método em que uma curva contínua (snake) se adapta aos contornos da face a partir de uma posição inicial. Turk e Pentland (M.Turk e A. Pentland, 1991) aplicam a técnica de análise em componentes principais (PCA) para detetar a presença e localização de faces numa imagem. Wang et al. (H. Wang e S. Chang, 1997) usa a informação sobre a cor da pele e faz uso do movimento característico da face de um indivíduo num vídeo.

Para o reconhecimento facial ser possível, é necessária a existência de uma base de dados de faces de indivíduos, uma vez que o processo de reconhecimento compara uma nova face adquirida com um conjunto de faces já armazenadas e conhecidas. A base de dados de faces humanas para reconhecimento facial deve ter em conta alguns aspetos fundamentais. Apenas devem ser guardadas as informações necessárias ao reconhecimento, de forma a que seja utilizado menos espaço possível. Deste modo, irá permitir que seja armazenado um maior número de faces, mas também que a comparação entre faces seja muito mais eficiente e rápida.

Segundo Brunelli e Poggio (R. Brunelli e T. Poggio, 1993), existem três técnicas principais para o armazenamento das faces humanas conhecidas na base de dados. A primeira é a abordagem geométrica, que usa a configuração espacial das características faciais para representar a face. As técnicas baseadas em características geométricas utilizam a dimensão e a distância entre os elementos

constituintes da face humana (olhos, sobrancelhas, nariz, boca, orelhas, etc.) ou quaisquer outros pontos para fazer a identificação da face. A configuração espacial dos elementos faciais é traduzida por uma matriz que contém medidas dos elementos faciais, tais como distâncias, ângulos e curvaturas. A segunda abordagem é baseada na textura da imagem, que usa informação inerente aos pixels que constituem a imagem da face humana. Este tipo de abordagem usa a figura da face como um todo para proceder ao reconhecimento, ou seja, as faces são representadas através de imagens ou através de características não geométricas obtidas a partir da imagem da face. Alguns destes métodos baseiam-se na correspondência de faces que são deformadas por algoritmos. Outros métodos baseiam-se no uso de funções de Gabor, que são filtros propostos por Dennis Gabor (Rolf P. Würtz, 1997) que alteram a imagem de modo a que os padrões e os limites das imagens sejam mais claros e inteligíveis, para se efetuar o reconhecimento. A terceira técnica passa pelos métodos de reconhecimento tridimensional, podendo esta utilizar qualquer uma das abordagens referidas anteriormente. No entanto, nestes métodos apenas é apresentada uma nova variável que possibilita a inclusão da dimensão de profundidade. O objetivo da introdução de mais uma dimensão consiste na hipótese de desenvolver um sistema mais exequível e difícil de enganar e de possibilitar a variação dos ângulos de como são apreendidas as imagens da face humana. Os primeiros estudos foram ao encontro da conceptualização das técnicas de modo a que fosse possível a variação de diferentes ângulos de captura de imagens das faces humanas (B. Moghaddam e A. Pentland, 1997) ou imagens faciais tridimensionais (Joseph J. Atick et al., 1996).

O Kinect é um sensor de deteção de movimentos, criado pela Microsoft para ser usado com os vídeos jogos da XBOX. Devido às suas potencialidades e ao seu baixo custo, várias empresas demonstraram interesse na utilização do sensor para funções distintas daquelas para as quais tinha sido criado, levando a Microsoft a lançar uma versão gratuita do Kinect para Windows SDK, permitindo às empresas externas à Microsoft começarem a explorar o potencial do Kinect. Com o lançamento do SDK várias empresas em todo o mundo começaram a explorar e a desenvolver aplicações mudando drasticamente o paradigma do sensor, que tinha sido criado para a indústria de vídeo jogos, sendo agora utilizado em vários projetos completamente distintos, como a indústria cinematográfica, a saúde, o ensino, sistemas de reconhecimento facial entre outras (Pagliari, 2011). O Kinect sofreu várias atualizações, tendo sido lançada a segunda versão, o Kinect V2, em julho de 2014. Segundo Castaneda (Castaneda, 2014), esta versão é muito mais robusta, tem uma fiabilidade de aquisição dos dados muito maior devido ao facto de possuir uma nova metodologia para a aquisição de valores de profundidade, baseando-se na diferença de tempo entre a emissão e a receção do sinal. Uma vez que a velocidade da luz é constante, é determinado o tempo que o sinal demora a voltar ao sensor. Segundo Fankhauser (Fankhauser et al., 2015), a precisão do Kinect V2

varia consoante a distância a medir até ao objeto, que nunca deverá ser inferior a 50 centímetros e superior a 4.5 metros.

Xiaohu Shao et al. (Xiaohu Shao et al.,2016) apresentam um sistema de reconhecimento facial tridimensional estabelecidas na obtenção de características faciais, a partir da reconstrução tridimensional da face humana através do uso de sensores e de seguida retiram as características de uma imagem normal RGB e apresentam os resultados do sistema.

Seema Mehra e Saurabh Charaya (Seema Mehra e Saurabh Charaya, 2016) realizaram comparações de diversas técnicas de reconhecimento facial de modo a conseguir obter as vantagens e desvantagens de cada uma das técnicas em ambientes controlados de laboratório e não em diversos ambientes do dia-a-dia, chegando à conclusão de que as grandes diferenças entre os métodos estão na deteção dos pontos do nariz e dos contornos da face humana.

Dalal e Triggs (Dalal e Triggs, 2005) através de vários estudos escolheram a combinação de SVM (Support Vector Machine) com o algoritmo HOG (Histogram of Oriented Gradient) para reconhecimento de pessoas. O SVM é um algoritmo de aprendizagem, que foi utilizado para treinar a identificação de uma pessoa. O SVM trata-se, fundamentalmente, de um classificador de duas classes. Enquanto o HOG é um método de extração de características, que tem como objetivo extrair informações referentes à orientação das arestas existentes numa imagem. O SVM combinado com o HOG possibilitou uma melhoria no reconhecimento.

Wang et al. (Wang et al., 2012), compararam cinco técnicas para correção de iluminação. Para o reconhecimento facial é utilizado a combinação de PCA (do Inglês: Principal Component Analysis) e LDA (do Inglês: Linear Discriminant Analysis). Wang et al. (Wang et al., 2012) utilizam uma base de dados de imagens de faces com diferentes níveis de iluminação, a combinação do PCA e LDA tiveram taxas de reconhecimento superiores a 80%.

Masi et al. (Masi et al., 2016) apresentam um estudo onde demonstram que é possível treinar algoritmos de reconhecimento facial com um número reduzido de imagens de faces recolhidas do que as que são geralmente precisas para a calibração do sistema de reconhecimento facial, podendo assim baixar o custo destes sistemas, descrevendo novos procedimentos que utilizaram e ainda apresentarem os seus resultados obtidos.

## **1.4 – Pipeline**

A Figura 1 resume a base do processo de reconhecimento facial usado na dissertação. O primeiro passo consiste na aquisição de imagem, usando o Kinectv2. À medida que as imagens vão sendo adquiridas é assegurada a existência de uma face na imagem. Se esta se verificar, é normalizada, ou seja, é apenas obtida a parte

da imagem que tem interesse e processada, de forma a ficar com as mesmas proporções das faces armazenadas na base de dados. De seguida, são extraídas as características da face, onde são utilizados alguns métodos diferentes. Após a obtenção dessas mesmas características, procede-se à sua redução, para que possam, por sua vez, corresponder ao formato dos dados já existentes na base de dados, uma vez que os dados presentes na base se encontram em formato reduzido. Por fim, são utilizados alguns classificadores diferentes, de modo a comparar a nova face com as que estão presentes na base de dados.

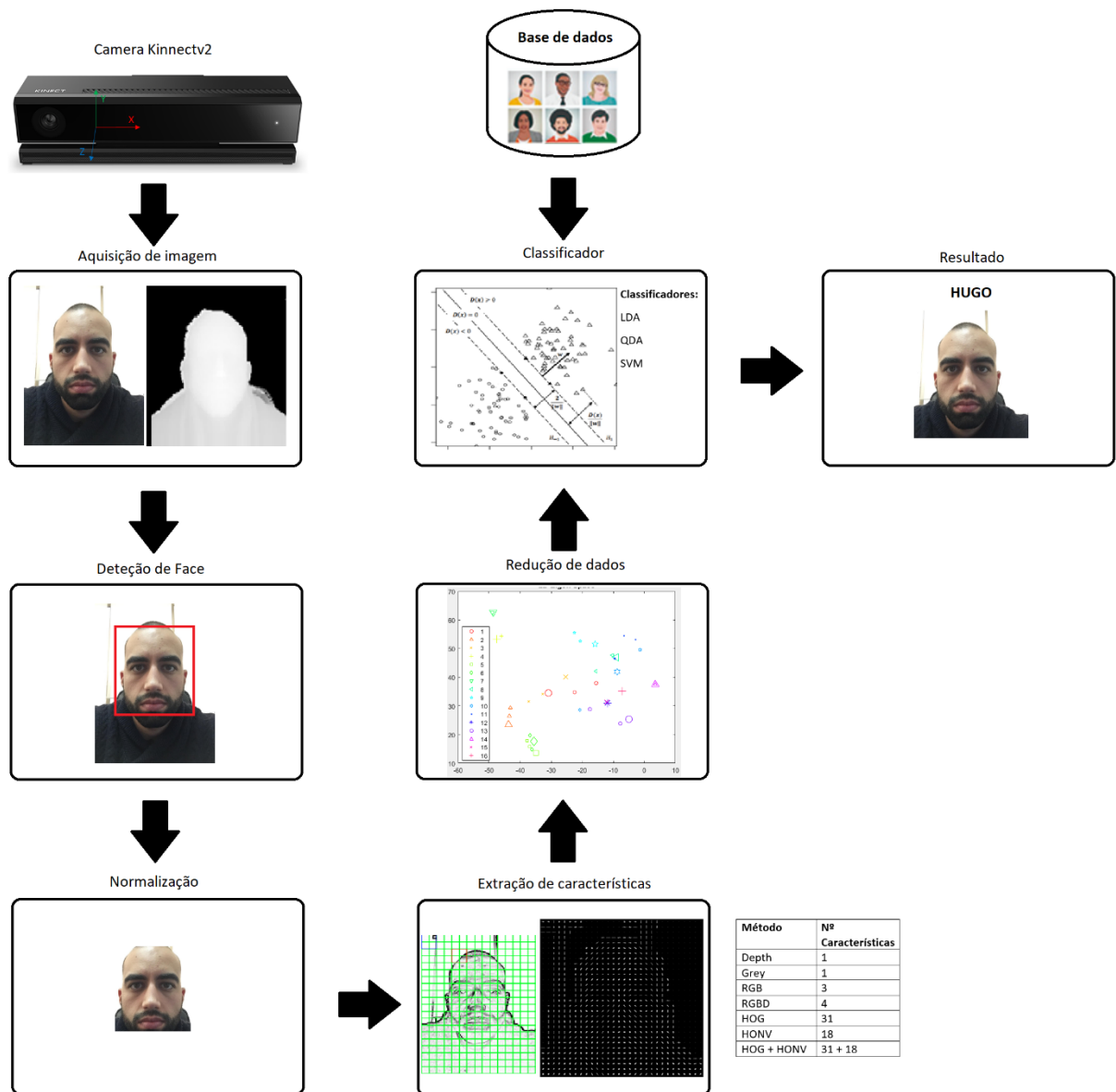


Figura 1 Processo de reconhecimento facial usado na dissertação

### 1.5 – Estrutura da dissertação

A dissertação encontra-se estruturada em cinco capítulos principais, procurando uma sequência e estrutura lógica de pensamento conforme se pode retirar da Figura 1. De modo a enquadrar devidamente o trabalho que é descrito neste documento, é apresentada a estrutura da dissertação:

No Capítulo 1, introdução, referem-se os problemas que motivaram e que conduziram à escolha do tema, onde se apresenta o problema e as questões que guiaram a investigação, bem como os objetivos a alcançar.

No Capítulo 2 apresentamos a Metodologia, no qual explicamos quais os procedimentos metodológicos utilizados ao longo da investigação e da elaboração da dissertação.

No Capítulo 3 são relatados a forma como as técnicas estudadas foram implementadas no projeto, e todo o desenvolvimento do trabalho experimental.

No Capítulo 4 apresenta-se e analisa-se dos resultados obtidos pelo protótipo implementado.

No Capítulo 5 apresentam-se as conclusões e a proposta de trabalhos futuros.



## CAPÍTULO 2

### Bases para um reconhecimento tridimensional

O reconhecimento facial aparenta ser um processo relativamente simples, mas, no entanto, é um processo extremamente complexo. Este processo é organizado em três fases: detectar e localizar a face humana numa imagem; recolher as características principais da face humana; e, por último, reconhecer a face humana tendo em conta todas as faces presentes numa base de dados. A comparação é realizada através de texturas como a cor da pele, a distância entre pontos ou ainda de um modelo composto pelas duas anteriores em simultâneo.

#### 2.1 – Introdução

Uma das grandes dificuldades neste processo são as diferentes posições e ângulos em que as imagens da face humana são captadas. Com o avanço da tecnologia e do processo de reconhecimento facial este obstáculo tem sido ultrapassado, tornando o processo alcançável. Com o avanço dos algoritmos é possível reconhecer uma face humana mesmo sem que esta se apresente na sua totalidade, ao contrário do que acontecia anteriormente, onde os utilizadores tinham que colocar a face seguindo sempre o mesmo modelo das faces registadas na base de dados.

O reconhecimento facial tem por base os pontos fixos da face humana, como os olhos, nariz e boca; não obstante, existem muitos outros que são usados para complementar estes pontos. Este processo está em constante desenvolvimento e é possível verificar um aumento da capacidade dos algoritmos em se adaptar aos vários ângulos e posições como as faces humanas são captadas, mesmo com uma diferença de amplitude que chega aos 90° (C. W. Hsu et al., 2010). O reconhecimento facial é mais preciso em indivíduos do sexo masculino do que do sexo feminino, que, embora as diferenças não sejam significativas, elas existem e têm que ser tidas em consideração (C. W. Hsu et al., 2010). Outra diferença interessante dos algoritmos de reconhecimento facial é que este, na maioria das vezes, são mais precisos em utilizadores mais velhos.

O progresso da fotografia digital tem contribuído substancialmente para a evolução do reconhecimento facial, pois a imagem é uma das matérias-primas do processo de reconhecimento facial. A facilidade de captura de imagens digitais de alta definição e os processadores dos computadores cada vez mais potentes têm levando à evolução do reconhecimento facial de tal forma que os melhores algoritmos quase que superam o desempenho dos seres humano (C. W. Hsu et al., 2010).

A primeira fase do processo de reconhecimento facial tem como desígnio detetar e localizar uma face humana numa imagem. Esta imagem pode ter níveis diferentes de complexidade, dependendo do cenário a imagem pode ter muitos objetos, o que torna a imagem muito complexa, caso contrário, se a imagem não apresentar na sua constituição muitos objetos, esta terá então menos complexidade. No reconhecimento facial apenas é necessária a face humana, tudo o resto que possa aparecer na imagem não será necessário, logo é fundamental obter a face humana da imagem e excluir o restante.

A segunda fase do processo de reconhecimento facial tem como objetivo localizar os pontos essenciais da face humana tais como os olhos, o nariz, a boca, os contornos da cara, as sobrancelhas e outros elementos possíveis de localizar.

A última fase do processo de reconhecimento facial tem o propósito de identificar se a face humana apresentada na imagem corresponde a alguma das faces humanas armazenadas na base de dados. A comparação entre a face humana e as faces presentes na base de dados é feita com base na percentagem de semelhanças, uma vez que é praticamente impossível haver duas imagens da mesma face exatamente iguais. A face humana da base de dados que apresentar maior percentagem de semelhanças é identificada como a face humana a reconhecer, no entanto, caso a percentagem de semelhanças não for satisfatória para o algoritmo a face é dada como desconhecida. Nos algoritmos esta percentagem de semelhança é dada pelo valor de distância entre os valores das características da face apresentada e das faces da base de dados, sendo que o valor de distância menor apresenta a face com maiores semelhanças.

No reconhecimento facial existem três categorias principais para a representação das faces humanas, como mostra a figura 2:

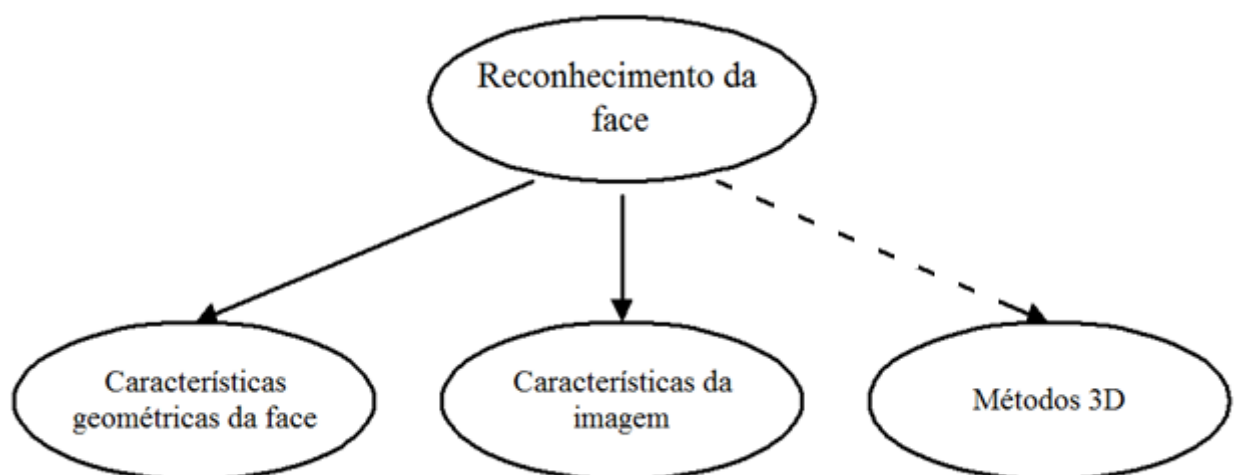


Figura 2 Representação dos três métodos principais para a representação das faces humanas.

O método baseado em características geométricas da face usa a informação das coordenadas das características faciais encontradas na imagem para representar a face. Neste método é usada a dimensão e a distância entre as características faciais como os olhos, as sobrancelhas, o nariz, a boca, ou quaisquer outros pontos encontrados na face humana que contribuam para a sua identificação. Esta informação é guardada num vetor que contém medidas das características faciais tais como distâncias, ângulos e curvaturas.

O método baseado nas características da imagem usa a informação de todos os pixels que representam a face humana na imagem. Esta abordagem usa a imagem da face humana na totalidade de modo a proceder ao reconhecimento facial, ao contrário do método anterior que apenas usa as características estruturais da face.

Os métodos baseados em informação tridimensional podem utilizar qualquer um dos métodos anteriores como base, onde apenas é introduzida mais uma dimensão, a profundidade. Ao introduzir uma nova dimensão o sistema de reconhecimento facial fica mais flexível, uma vez que o sistema tem mais informação, o que permite reconhecer faces humanas mesmo que recolhidas em ângulos um pouco diferentes. O sistema acaba também ficar mais fiável, uma vez que se torna muito mais difícil conseguir enganar o sistema através de imagens e fotografias.

## **2.2 – Formação de imagens**

O processo de criação de uma imagem normal (RGB 2D) descreve uma imagem do mundo real, em 3D, que é projetada em uma imagem 2D onde a profundidade da imagem é representada pela intensidade de cor. A figura 3 ilustra, de forma simples, a base de funcionamento do processo de formação de uma imagem 2D. Para produzir uma imagem 2D é obrigatório que exista um emissor de fonte de luz natural ou artificial, que é emitida e então refletida na superfície de um objeto 3D do mundo real. Essa reflexão de luz é captada pelas lentes das câmaras e é formada a imagem 2D com a profundidade dada pela intensidade de cor (Richard Szeliski, 2011).

As câmaras digitais modernas processam as imagens antes da captura, de modo a remover o ruído existente e melhorar a qualidade das imagens, sendo essas depois guardadas em diversos formatos conhecidos, como, por exemplo, o formato RGB (Gonzalez e Woods, 2006).

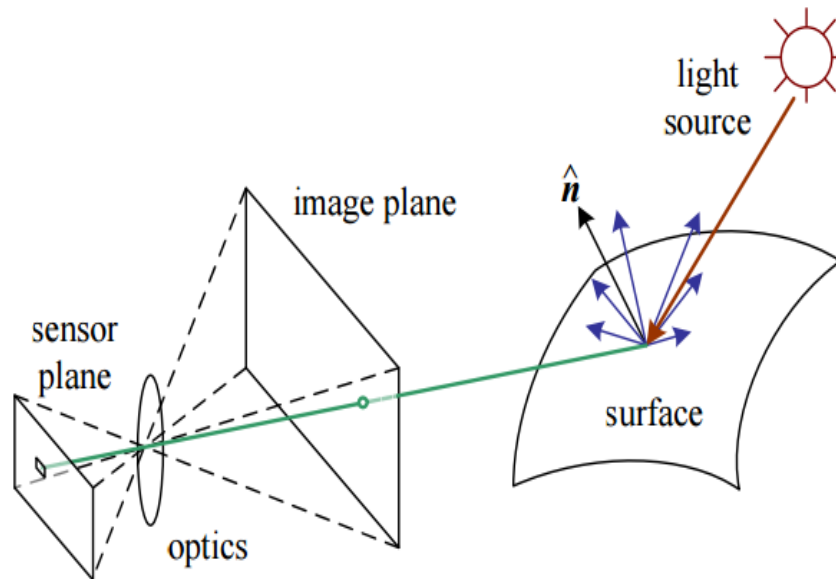


Figura 3 Exemplo de formação de uma imagem RGB 2D (Richard Szeliski, 2011)

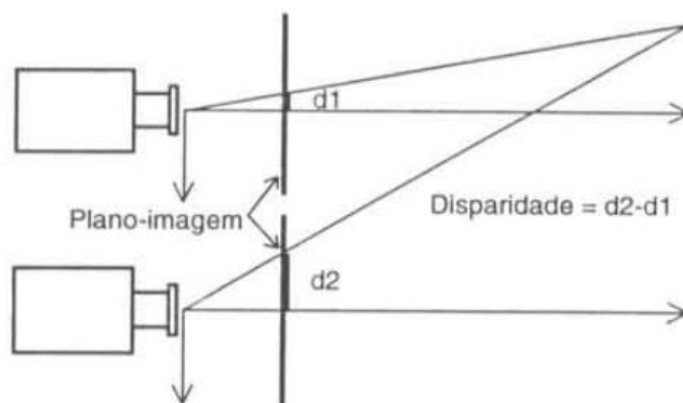
Ao contrário de uma imagem normal (RGB 2D) que designa a intensidade da cor dos objetos na imagem, uma imagem de profundidade informa em cada pixel a distância entre a câmara e os objetos em ação. Existem muitas áreas onde são usadas este tipo de imagens que apresentam informação de profundidade, uma vez que esses mesmos tipos têm maior facilidade de fragmentar objetos do cenário, remover o fundo, calcular o tamanho dos objetos, reconhecer objetos e pessoas, entre outras.

Os seres humanos confiam no conhecimento que foram adquirindo ao longo do tempo e assim conseguem realizar facilmente esta tarefa de estimar a profundidade do que é possível observar no mundo real, apesar de, por vezes, ser possível ocorrerem enganos (Richard Szeliski, 2011). O ser humano, mesmo com a visão de apenas olho, consegue realizar esta tarefa, porque tem à sua disposição outros sentidos e outras informações adquiridas com o tempo, uma vez que já esteve em contacto com os mesmos objetos ou semelhantes várias vezes tendo memória destas informações, enquanto que as câmaras não possuem, o que impossibilita adivinhar a distância apenas através de uma só imagem.

Existem várias técnicas usadas para a obtenção da profundidade de um cenário, as principais e mais usadas são: Visão Estéreo, Fotogrametria; Triangulação laser; Luz estruturada; Tempo de propagação do sinal; e Franjas de Moiré. Estas técnicas são de dois tipos diferentes, denominadas de técnicas passivas e técnicas ativas.

As técnicas passivas utilizam apenas a luz ambiente para iluminar o cenário. A informação relativa à distância é obtida a partir de uma ou mais imagens captadas do cenário. Estas técnicas têm como objetivo conseguir a informação da distância a partir de uma ou mais imagens do mesmo cenário, obtidas com uma ou várias câmaras. Este tipo de técnicas apenas possibilitam obter informação sobre a

orientação das superfícies ou da distância relativa entre os pontos do cenário, não permitindo obter as distâncias. A estereoscopia passiva é a técnica mais usada e conhecida, esta permite obter a informação de distância a partir de duas imagens extraídas do cenário com duas câmaras colocadas lado a lado com algum afastamento entre elas.



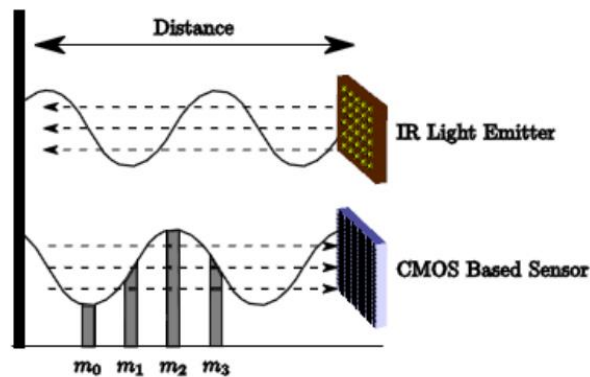
**Figura 4 Exemplo de calculo de distancia através da estereoscopia passiva (Jorge Alves da Silva,1994)**

As técnicas ativas, contrariamente às técnicas passivas que apenas usam a luz natural, usam a projeção direta e de forma controlada de energia no cenário, utilizando vários projetores e sensores, de modo a que seja possível a obtenção de informação tridimensional. A energia projetada pode ser de diversos tipos, onde os mais usados são os de tipos de luz e ultrassons, sendo essa mesma energia refletida pelo cenário e captada por sensores recetores obtendo assim a informação tridimensional do cenário.

Neste projeto é utilizado o Kinect v2 que usa uma técnica ativa denominada de Tempo de propagação do sinal (Time of flight).

Os sistemas baseados em propagação de sinal consistem em enviar um sinal e medir as propriedades do mesmo refletido pelo cenário. As propriedades do sinal são usadas para calcular o tempo de propagação do sinal (TOF), sendo possível obter a distância multiplicando o tempo de propagação do sinal pela velocidade do mesmo no meio físico em questão. O tempo de propagação pode ser calculado modificando o sinal por impulsos ou por um sinal contínuo (Fu-Pen Chiang,1979). Os sensores baseados em impulsos calculam o tempo de trajeto de um impulso de luz, enquanto os sistemas baseados num sinal contínuo calculam a diferença entre o sinal transmitido e o sinal recebido. Na figura 5 está representado um sistema de cálculo da profundidade onde é usado um sinal de luz contínuo. O sistema projeta luz infravermelha através de leds que é refletida pelo cenário e é capturada pelo sensor e cada um dos pixéis do sensor retira uma amostra dessa mesma luz refletida pelo cenário quatro vezes em cada período do sinal ( $m_0,1,2,3$ ). Estas quatro

medições permitem conseguir o sinal refletido, o que permite calcular as distâncias dos objetos e obter assim o mapa tridimensional do cenário.



**Figura 5 Tempo de propagação do sinal de luz (Fu-Pen Chiang,1979).**

Esta técnica tem boa performance, especialmente para aplicações em tempo real, uma vez que os sensores têm uma grande capacidade de processamento de imagens (Fu-Pen Chiang,1979). As câmaras baseadas nesta técnica são muito recentes e permitem obter imagens tridimensionais a uma velocidade excepcional, bem como permitir obter a informação referente à profundidade para todos os pixéis do cenário, resultando numa imagem tridimensional de alta qualidade e precisão. Nesta técnica a sua precisão e qualidade depende da quantidade de luz refletida pelo cenário e que chega ao sensor e vários parâmetros podem afetar a quantidade de luz refletida, até mesmo as propriedades dos objetos e da textura. A luz ambiente e outras fontes de luz existentes no cenário afetam negativamente o desempenho da técnica. Este tipo de técnica ainda requer algum progresso, uma vez que ainda não tem a mesma precisão técnica como a da triangulação a laser (Fu-Pen Chiang,1979).

### **2.3 – Extração de características**

A extração de características tem como objetivo reduzir a dimensionalidade dos dados eliminando informações redundantes. Imagens de face são frequentemente representadas por uma matriz de dimensão de altura por largura, ou seja, a largura e a altura da imagem em pixels, contendo valores entre 0 e 255, referentes à intensidade de cada pixel em escala de cinza. Deste modo, as amostras processadas são frequentemente representadas por vetores contendo apenas as informações mais relevantes ao contexto da aplicação.

#### **2.3.1 Histograma da orientação de gradientes (HOG)**

Histograma da orientação de gradientes, do inglês Histogram of Oriented Gradients, designado de HOG, é um método de extração de informação de imagens que foi introduzido em 2005, pelos investigadores Dalal e Triggs (Dalal e Triggs, 2005). Este método tem como objetivo extrair informações referentes à orientação das arestas existentes numa imagem.

No artigo de Déniz et al. (Déniz et al., 2011) é demonstrado que o HOG pode ser utilizado para auxiliar o reconhecimento facial, compensando erros, na deteção das características, provocados por fatores como oclusão parcial, pose e mudanças de iluminação.

Descrevendo o método de forma resumida e concisa, vamos assumir que se está a trabalhar com imagens em tons-de-cinza, uma vez que este método funciona com este tipo de imagens, que deveram ser convertidas anterior mente para este tipo. Desta forma, todas as imagens a cores terão de ser convertidas para imagens deste tipo, como demonstra a Figura 6 .

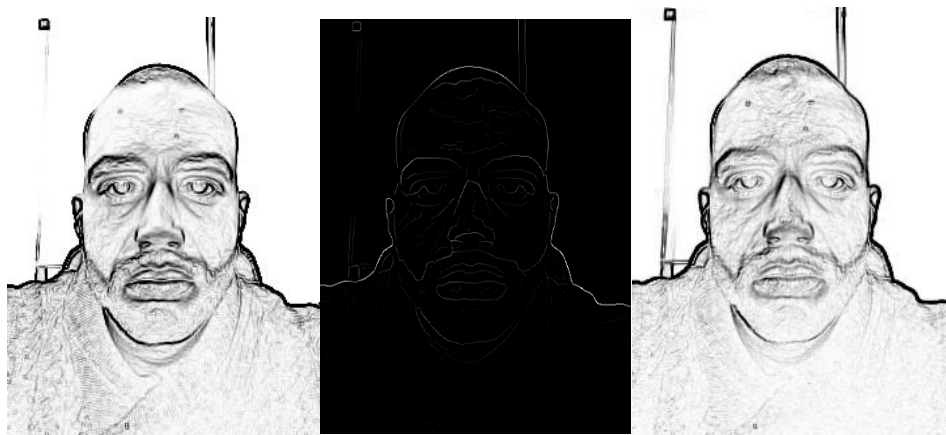
O método está dividido em quatro fases:

- Cálculo da orientação e magnitude dos gradientes de todos os pixéis.
- Divisão da imagem em blocos e células.
- Cálculo do histograma de gradientes por células, agrupados em blocos.
- Ligação destes histogramas. Estes histogramas ligados, formam um único vetor, sendo este vetor o descritor HOG.



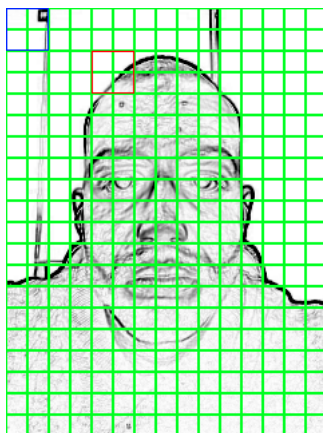
**Figura 6 Exemplo de conversão de imagem a cores para imagem em tons de cinza**

A primeira fase, denominada de cálculo da orientação e magnitude das arestas, tem como foco os contornos dos objetos numa determinada imagem, onde é possível utilizar vários filtros que transformam uma imagem comum numa imagem de contornos. Os métodos de deteção de arestas recorrem à iluminação dos pixéis para detetar contornos. Se  $k$  pixéis adjacentes tiverem uma diferença acentuada de iluminação, poderá indicar que existe um contorno entre eles. Existem vários filtros para transformar a imagem, como por exemplo os filtros Prewitt e Canny, sobel, entre outros. A Figura 7 ilustra a utilização do filtro de Sobel numa imagem.



**Figura 7 Exemplo da utilização do filtro de Prewitt, Canny e Sobel respetivamente.**

Na segunda fase serão definidas duas estruturas, designadas por células e blocos. Estas células são janelas da imagem original, com dimensões de  $v * v$  pixéis. Um bloco é visto como uma junção de  $n * n$  células, perfazendo um total de  $(n * v) * (n * v)$  pixéis. Dalal and Triggs (Dalal e Triggs, 2005) consideram que de bloco para bloco existe sobreposição de uma célula em qualquer uma das direções, horizontal ou vertical, na divisão da imagem em blocos. A divisão da imagem em células e blocos tem como objetivo tentar maximizar o ganho de informação no processamento de uma imagem. Este método executa uma extração de características (vetores descritores) da imagem. Alguns dados associados aos descritores e não apenas à observação direta da magnitude da cor ou aresta num pixel. A Figura 8 ilustra a divisão de uma imagem em células e o agrupamento de células em blocos. Apenas estão representados dois blocos a vermelho e a azul, no entanto, a divisão em blocos será mais densa, havendo um bloco para cada célula, em qualquer direção.



**Figura 8 Exemplo de imagem dividida em células verdes e exemplo de dois blocos possíveis a azul e vermelho.**

A terceira fase visa extrair informação sobre a orientação das arestas existentes na imagem (orientação dos gradientes), e no agrupamento das direções das arestas num histograma com  $n$  divisões. Este agrupamento será feito por células num dos blocos, repetindo-se este processo para os restantes blocos. A orientação dos gradientes de cada pixel da imagem tem os valores entre  $-90^\circ$  e  $90^\circ$ , com intervalo mínimo entre os ângulos igual a  $20^\circ$ . A orientação de cada pixel terá um dos seguintes valores  $[-90^\circ, -70^\circ, -50^\circ, -30^\circ, -10^\circ, 10^\circ, 30^\circ, 50^\circ, 70^\circ, 90^\circ]$ . Antes de ser feito o agrupamento em  $n$  orientações, dar-se-á uma ponderação das magnitudes dos gradientes, em cada pixel do bloco, dando maior importância aos valores centrais do bloco.

Na quarta fase, será calculado o histograma das orientações dos gradientes, por cada bloco, em cada uma das células separadamente. Considera-se que um pixel  $px_{ij}$ , da célula  $c_k$ , de um determinado bloco, seja a magnitude do gradiente deste pixel dado por  $s_{ij}$  e a sua orientação por  $o_{i,j}^l$ , com  $l = 1, 2, \dots, n_b$ . Este pixel contribuirá para a orientação  $l$  do histograma da célula  $c_k$  com o valor da sua magnitude, ou seja,  $s_{ij}$ . Este processo é repetido para os restantes pixels da célula, sendo o resultado um vetor de dimensão  $n_b$ . Finalmente, é executada uma normalização do histograma, onde é obtido o maior valor do histograma e de seguida são divididos todos os elementos por esse maior valor, de forma a este poder ser visto como uma estimativa. Este processamento é repetido para cada uma das  $n_c * n_c$  células do bloco, sendo o resultado de todas estas células agrupadas. Este resultado é designado por descritor HOG. Este processo será aplicado a cada um dos blocos de uma imagem. Estes descritores serão, então, agrupados, formando um único vetor, denominado de vetor descritor de uma determinada imagem. Para uma imagem geral, esta será dividida em janelas com as dimensões necessárias, sendo sobre estas janelas calculados os descritores HOG. Na Figura 9 está representado um exemplo da ilustração dos histogramas por célula da imagem.

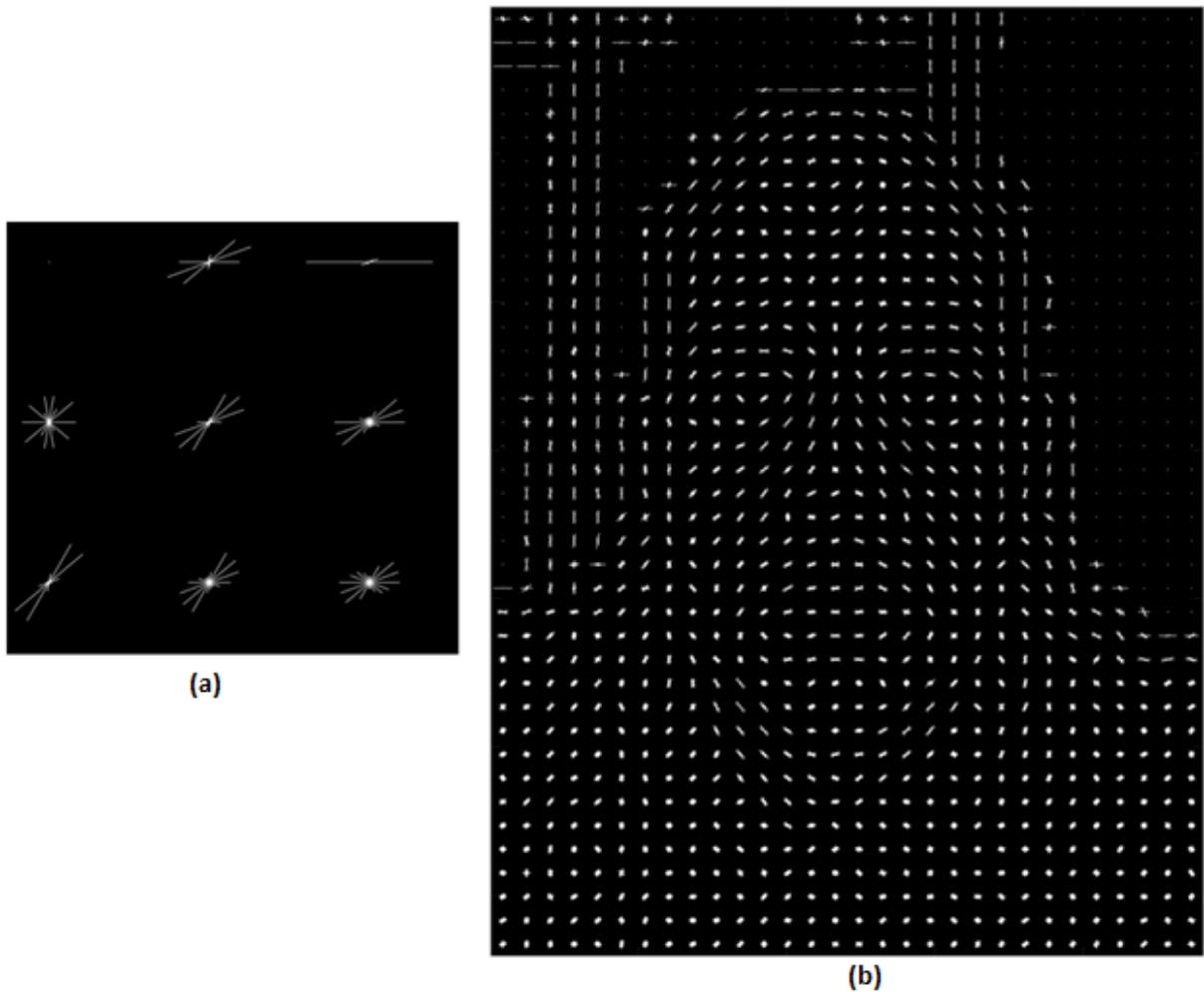


Figura 9 (a) Exemplo de agrupamento de orientações por célula. (b) Exemplo da representação das orientações na imagem

### 2.3.1.1 Pipeline

Passo 1: Fixar os valores de:  $n_c^2$  número de células por bloco;  $v^2$  número de píxeis por célula; e  $n_b$  número de divisões do histograma de orientação. Fixar qual o método de deteção de arestas a utilizar.

Passo 2: Cálculo do gradiente de arestas com orientação vertical e horizontal,  $G_v$  e  $G_h$ , respetivamente.

Passo 3: Cálculo da magnitude do gradiente em cada pixel.

Passo 4: Cálculo do ângulo da fase.

Passo 5: Divisão da matriz de orientação e gradiente,  $O$  e  $G$ , em blocos de  $n_c * n_c$  células, sendo que blocos adjacentes têm duas células de sobreposição entre si (em cada orientação), células estas, de dimensão  $v * v$  píxeis.

Passo 6: Aplicação de uma máscara gaussiana de dimensão  $(n_c * v) * (n_c * v)$  a cada um dos blocos da matriz de gradientes, dando assim um peso reduzido

aos gradientes consoante se encontrem mais afastados dos gradientes centrais do bloco.

Passo 7: Cálculo de histogramas de orientação, com  $n_b$  divisões, em cada uma das células. Cada orientação é pesada no histograma pela respectiva magnitude do seu gradiente.

Passo 8: Concatenação dos  $n \frac{2}{c}$  histogramas, formando um descritor HOG.

Passo 9: Normalização do descritor, segundo a norma l2:  $\frac{x}{\sqrt{|x^2+c|}}$

Passo 10: Repetição dos passos 6 a 9, para cada bloco e concatenação de todos os descritores, obtendo assim o resultado final.

### 2.3.2 Histograma de Vetores Normais Orientados

Histograma de Vetores Normais Orientados (HONV) foi projetado por Tang et. al em 2012 (Tang S. et al., 2012) para obter características da geometria 3D a partir da profundidade da imagem, ou seja, é aplicado apenas a imagens que apresentem informação relativa à profundidade, imagens tridimensionais. É esperado que um objeto seja reconhecido, tendo em conta a sua superfície 3D e não a sua textura. No processamento é usado um filtro gaussiano, é um filtro que é utilizado para filtrar ou desfocar a imagem com o objetivo de reduzir os ruídos presentes na imagem de profundidade. HONV são recursos baseados em histogramas, como os recursos HOG. Presume-se que a superfície do objeto represente informação de categorias de objetos, porque a superfície do objeto pode ser representada por um plano tangente, isto é, um vetor normal para dar a coordenada da superfície. As características da geometria 3D podem ser representadas como uma distribuição local da orientação do vetor normal. Tang et. Al (Tang S. et al., 2012) fez uma reformulação, que mostra que o vetor normal pode ser representado como um par ordenado de ângulos denominados de ângulos de Azimuth e de Zenith representados na figura 11, que podem ser facilmente calculados a partir do gradiente de imagem de profundidade. HONV é a combinação do histograma local dos ângulos, para que possa ser usado como recurso na deteção e classificação de objetos.



Figura 10 (a) Exemplo de imagem RGB (b) representação em imagem da informação de profundidade

O vetor normal na posição  $p = (x, y)$  é o produto de dois vetores tangentes. Através de uma reformulação, Tang et. al obtém a fórmula do vetor normal do pixel da imagem de profundidade  $(x, y, d(x, y))$ . As coordenadas esféricas são usadas para codificar informações da orientação.

O angulo Zenith representado na figura 11 é calculado através da seguinte equação:

$$\theta = \tan^{-1} \left( \left( \frac{\partial d(x, y)}{\partial x} \right)^2 + \left( \frac{\partial d(x, y)}{\partial y} \right)^2 \right)^{\frac{1}{2}} \quad (1)$$

,e o Azimuth também representado na figura 11, é calculado através da seguinte equação:

$$\varphi = \tan^{-1} \left( \frac{\partial d(x, y)}{\partial y} / \frac{\partial d(x, y)}{\partial x} \right) \quad (2)$$

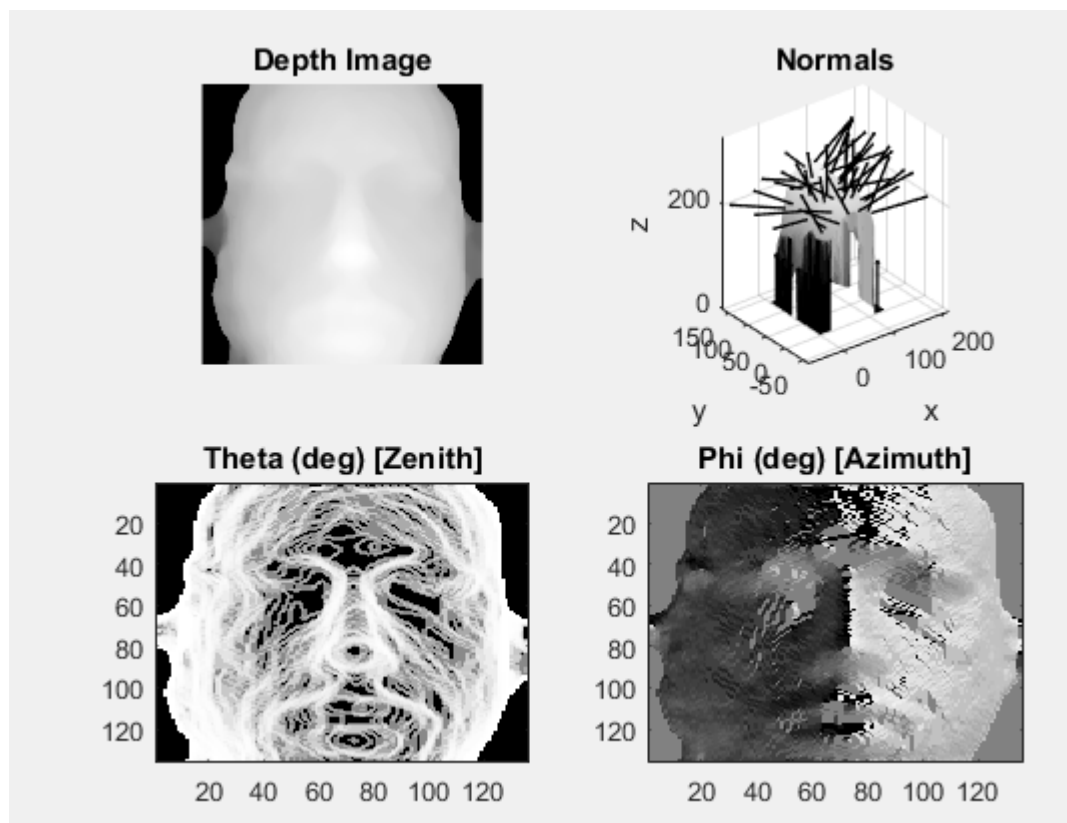


Figura 11 Representação dos ângulos e vetores calculados em HONV

### 2.3.2.1 Pipeline

Passo1: Dividir a imagem em janelas de detecção de tamanho de  $m * n$  células.

Passo2: A orientação do vetor normal em cada célula é processada e transformada em histogramas.

Passo3: A informação final obtida é conseguida através da combinação das características de HONV de cada célula.

## 2.4 – Classificação/Reconhecimento

As tarefas relacionadas com reconhecimento de imagem, podem ser resumidas essencialmente a um processo de categorização de dados. Essa categorização é feita recorrendo a classificadores específicos a cada situação. Os classificadores são usados na resolução de problemas em que o objetivo é identificar, a partir de um conjunto de dados observados, a classe a que um novo dado observado pertence. Isto é feito com base num conjunto de dados observados de treino cuja categoria associada é conhecida. Os novos dados individuais são então analisados e traduzidos para um conjunto de propriedades quantificáveis.

Um classificador pode ser visto como que o resultado da análise feita por um algoritmo de Machine Learning a um conjunto de dados. Ou seja, cada algoritmo de aprendizagem, através das suas assunções próprias, encontra uma explicação diferente sobre os dados analisados, criando cada um deles um classificador diferente. Estes classificadores podem cometer erros em certas ocasiões, por isso, são criados conjuntos de classificadores para que estes se complementem, tendo desta forma melhores resultados que individualmente.

#### 2.4.1 Principal Component Analysis (PCA)

A Análise de Componentes Principais (ACP) ou Principal Component Analysis (PCA), também conhecida como Eigenfaces (Sirovich L. e Kirby M., 1987) é um método de identificar características extraídas de dados. Torna-se vantajoso quando os dados têm muitas dimensões mesmo que não seja possível uma representação gráfica dos mesmos, no entanto, é também útil em dimensões menores. Esse método de redução de dados foi desenvolvido por Pearson e Hotelling, tem como objetivo compressão ou de redução de dimensões de forma a capturar o essencial da informação envolvida.

As etapas que permitem calcular as componentes principais são:

- Cálculo da média ou do vetor médio dos dados;
- Subtração da média a cada elemento dos dados;
- Cálculo da matriz de covariância.
- Cálculo dos “valores próprios” e “vetores próprios” da matriz de covariância.
- Obtenção da matriz resultante do PCA, na qual as colunas são formadas a partir dos vetores próprios da matriz de covariância obtidos de forma a que a primeira coluna, o elemento (0,0), seja o vetor próprio correspondente ao maior valor próprio, e assim sucessivamente, até que a última linha corresponda ao menor valor próprio.

O vetor próprio com o maior valor próprio associado, corresponde à componente principal do conjunto dos dados utilizados. O que significa que esse é o dado mais significativo do processo.

### 2.4.1.1 Matriz de Covariância

Na matemática, existem várias análises que podem ser feitas sobre um conjunto de dados, como, por exemplo, a média aritmética, o desvio padrão e a variância. Tanto o desvio padrão como a variância medem a distância dos dados em relação a média, na qual a variância é igual ao quadrado do desvio padrão. A covariância é a medida entre duas dimensões, na medida em que, no cálculo da covariância entre uma dimensão e ela mesma, o resultado é a variância. A forma da covariância para duas dimensões está representada a baixo.

$$cov(x, y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{(n - 1)} \quad (3)$$

Na fórmula 3, o X representa a primeira dimensão e o Y a segunda. Os elementos com uma barra por cima representam a média. Os elementos  $X_i$  e  $Y_i$  representam, por sua vez, as dimensões dos elementos na  $i$ -ésima posição e a variável  $n$  representa o número de elementos dos dados obtidos.

Se os dados tiverem mais de duas dimensões é necessário calcular a covariância entre cada par de dimensões, obtendo assim a matriz de covariância. Se, por exemplo, forem usadas três dimensões (x,y,z), a matriz de covariância terá o formato representado a baixo.

$$C = \begin{pmatrix} cov(x, x) & cov(x, y) & cov(x, z) \\ cov(y, x) & cov(y, y) & cov(y, z) \\ cov(z, x) & cov(z, y) & cov(z, z) \end{pmatrix} \quad (4)$$

A partir de uma matriz M de dados de ordem  $n \times p$  é possível obter a matriz de covariância representada por:

$$C = \begin{bmatrix} var(x_1) & cov(x_1x_2) & cov(x_1x_3) & \dots & cov(x_1x_p) \\ cov(x_2x_1) & var(x_2) & cov(x_2x_3) & \dots & cov(x_2x_p) \\ cov(x_3x_1) & cov(x_3x_2) & var(x_3) & \dots & cov(x_3x_p) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ cov(x_px_1) & cov(x_px_2) & cov(x_px_3) & \dots & var(x_p) \end{bmatrix} \quad (5)$$

A diagonal principal da matriz de covariância representa as variâncias e as restantes posições da matriz representam a correlação entre as direções. A matriz de covariância é uma matriz simétrica, uma matriz diz-se simétrica se coincidir com

a sua transposta, ou seja,  $C = C^T$ , sendo deste modo possível encontrar um conjunto de vetores próprios ortonormais. Vetores são ortonormais se forem vetores perpendiculares e unitários.

No entanto, para que covariância possa ser calcula pelo método descrito em cima é necessário que o número de amostras seja maior que o número de dimensões. Se o número de amostras for menor que o número de dimensões a matriz de covariância pode ser muito grande, no entanto os valores e vetores próprios podem ser calculados de uma matriz mais pequena derivada da matriz de dados. Para calculada uma estimativa da covariância aplica-se a fórmula seguinte:

$$cov = \frac{1}{(n-1)} \sum_{i=1}^n (x_i - \bar{x})(x_i - \bar{x})^T \quad (6)$$

#### 2.4.1.2 Vetores próprios e valores próprio

Diz-se que um vetor  $v$  é um vetor próprio de uma matriz quadrada  $M$  se a multiplicação da matriz  $M$  pelo vetor  $v$  resultar num múltiplo de  $v$ , ou seja, em  $\lambda v$ . O  $\lambda$  é o chamado valor próprio de  $M$  associado ao vetor próprio  $v$ . Como representa a seguinte formula:

$$Mv = \lambda v \quad (7)$$

Para matrizes de duas ou até três dimensões, os valores próprios podem ser calculados usando a equação característica de  $M$ :

$$\det(M - \lambda I) = 0 \quad (8)$$

Onde,  $I$  representa a matriz de identidade, o  $M$  a matriz de dados e o  $\lambda$  os escalares não nulos.

No caso de dimensões maiores, é aplicado um algoritmo numérico iterativo. O último passo é ordenar os vetores próprios por ordem decrescente.

#### 2.4.1.3 Cálculo PCA

Sendo a matriz covariância simétrica é sempre possível encontrar um conjunto de  $n$  vetores próprios ortonormais. Considerando que  $n$  vetores próprios

encontram-se ordenados de modo decrescente de acordo com os  $n$  valores próprios, chamamos de  $e_1$  ao vetor próprio correspondente ao maior valor próprio que chamamos de  $\lambda_1$ , e assim sucessivamente, de modo a chamarmos  $e_{n-1}$  ao vetor próprio correspondente ao menor valor próprio que chamamos de  $\lambda_{n-1}$ .

Consideramos, então, uma matriz  $A$ , cujas colunas são vetores próprios de  $C_x$  que se encontram ordenados, como demonstrado no parágrafo anterior, e agora consideramos uma transformação definida por esta matriz do seguinte modo:

$$b = A(x - m_x) \quad (9)$$

Os valores de  $x$  vão ser mapeados, em valores de  $b$ , cuja média será zero, isto é,  $m_y = 0$ , e cuja matriz de covariância de  $Y$  pode ser obtida a partir de  $A$  e  $C_x$ :

$$C_y = A C_x A^T \quad (10)$$

Assim sendo, a Análise dos Componentes Principais (PCA) é um método que encontra os valores próprios e vetores próprios da matriz de covariância dos dados adquiridos. Através do resultado do PCA é possível realizar uma redução dimensional dos mesmos. Este método auxilia na separação da informação importante daquela que é redundante e aleatória, ajudando a comprimir os dados necessários, reduzindo o espaço necessário para a representação das imagens. Com o uso deste método a visualização de diversas variáveis em um determinado conjunto de dados torna-se mais rápida, objetiva e eficiente.

#### 2.4.1.4 Pipeline

Passo 1: Obter o número de dimensões e o número de amostradas.

Passo 2: calcular a media dos dados.

Passo 3: Subtrair a media a cada elemento dos dados.

Passo 4: calcular a matriz de covariância. Tem que se ter atenção porque se as dimensões forem maiores que as amostras não é possível realizar o cálculo da covariância pela forma standard, podendo se usada a função `cov` que devolve a covariância de uma matriz. No entanto se número de amostras for menor que o número de dimensões é calculada pela forma standard.

Passo 5: Calcular os valores próprios e os vetores próprios da matriz de covariância.

Passo 6: Obter o número de vetores próprios necessários para cumprir a percentagem de redução pretendida.

Passo 7: Obter variância para cada dimensão, mas só para os vetores próprios necessários obtidos no passo anterior.

Passo 8: Obtenção da matriz que representa os dados comprimidos pelo PCA.

#### 2.4.2 Linear Discriminant Analysis (LDA)

O método LDA (do Inglês: Linear Discriminant Analysis) é um método de análise multivariada proposto por Fisher (Fisher R.A. ,1936) que tem como objetivo identificar as variáveis que melhor diferenciam dois ou mais grupos de indivíduos e construir uma regra de classificação que permita prever a que grupo um novo indivíduo pertence. Este método calcula o hiperplano de maior divisão entre os grupos de indivíduos analisados. O cálculo desse hiperplano considera o conhecimento prévio dos objetos ou grupo de cada amostra. A análise LDA considera que a distribuição de probabilidade das amostras é conhecida e pode ser representada pela média e pela dispersão das amostras. O método baseia-se na diminuição da dissipação das amostras com relação ao grupo ao qual pertencem e, ainda, na maximização da distância da média entre estes grupos. Ou seja, são calculadas as matrizes de dissipação inter-classes e intra-classes com objetivo de discriminar os grupos de amostras pela maximização da separabilidade entre objetos, enquanto é minimizada a variabilidade dentro das mesmas.

As matrizes de dissipação inter-classes ( $S_b$ ) e intra-classes ( $S_w$ ) são definidas como:

$$S_b = \sum_{i=1}^g N_i (\bar{x}_i - \bar{x}) (\bar{x}_i - \bar{x})^T \quad (11)$$

$$S_w = \sum_{i=0}^g \sum_{j=1}^{N_i} (x_{i,j} - \bar{x}) (x_{i,j} - \bar{x})^T \quad (12)$$

Sendo que,  $g$  é o número de grupos analisados,  $N_i$  a quantidade de amostras do grupo  $i$ ,  $\bar{x}$  e  $\bar{x}_i$  são a média total e a média das amostras do objeto  $i$ , respectivamente, e  $x_{i,j}$  é a amostra  $j$  do grupo  $i$ .

O principal objetivo do LDA é encontrar a matriz de projeção  $P_{LDA}$  que maximiza a razão entre o determinante da matriz de dissipação inter-classes  $S_b$  e o

determinante da matriz de dissipação intra-classes  $S_w$ , conhecido como critério de Fisher (Fisher R.A. ,1936), definido como:

$$P_{LDA} = \arg \max \frac{|S_b|}{S_w} \quad (13)$$

O critério descrito pela equação (13) é atingido quando a matriz de projeção  $P_{LDA}$  é composta, no máximo, pelos  $(g - 1)$  vetores próprios de  $S_w^{-1}S_b$ , cujos valores próprios correspondentes são não nulos. Na prática, esse critério apenas pode ser calculado se a matriz de dissipação intra-classe  $S_w$  for não singular. Deve ser observada a proporção entre o número de amostras e de variáveis. Em cenários nos quais o número total de amostras  $N$  é bem menor que o número de variáveis  $n$ , ocorre uma instabilidade no cálculo da matriz inversa de  $S_w$ . A quantidade de amostras necessárias para evitar essa instabilidade no cálculo da matriz inversa de  $S_w$  deve ser igual ou superior a cinco vezes a quantidade de destas variáveis.

#### 2.4.2.1 Classificação

Considere que  $G$  populações, onde estas têm distribuição normal multivariada, com matriz de covariâncias comum  $\Sigma$  e de dimensão  $p \times p$  e vetores de médias  $\mu_g$  ( $g = 1, \dots, G$ ).

Considere uma amostra aleatória dessas populações  $n$  ( $n = n_1 + n_2 + \dots + n_G$ ). O objetivo da LDA é identificar a que população a amostra pertence.

#### 2.4.2.2 Função discriminante

Seja  $x_{g,i}$  a  $i$ -ésima amostra da população  $g$ , a ideia do LDA é classificar a amostra  $x_{g,i}$  como pertencente à população  $\tilde{g}$  que minimiza a sua distância ao centro:

$$(x_{g,i} - \mu_{\tilde{g}})^T \Sigma^{-1} (x_{g,i} - \mu_{\tilde{g}}) \quad (14)$$

que identifica a população que maximiza a verosimilhança.

Em vez de maximizar a verosimilhança, é possível maximizar a probabilidade de modo a que  $\pi_1 + \dots + \pi_G = 1$ , de forma a que amostra pertença a determinada população:

$$x_{g,i} \in \text{população } \tilde{g} = \arg \min_{g'} \left[ \frac{1}{2} (x_{g,i} - \mu_{g'})^T \Sigma^{-1} (x_{g,i} - \mu_{g'}) \right] \quad (15)$$

A linearidade do método resulta do pressuposto de matriz de covariâncias comum, que simplifica o critério anterior:

$$x_{g,i} \in \text{população } \tilde{g} = \arg \max_{g'} d_{g'}(x_{g,i}) \quad (16)$$

Sendo que:

$$d_g(x) = x^T \sum^{-1} \mu_g - \frac{1}{2} \mu_g^t \sum^{-1} \mu_g + \log \pi_g \quad (17)$$

é a função discriminante (Guo T. et al.,2007). As variáveis  $\pi_g$  e  $\Sigma$  são desconhecidas e têm que ser estimados a partir da amostra. Usando o método de maximizar verosimilhança, obtém-se (Hastie et al.,2009):

$$\hat{\mu}_g = \hat{x}_g = \frac{1}{n_g} \sum_{i=1}^{n_g} x_{g,i} \quad (18)$$

e

$$\widehat{\Sigma} = \frac{1}{n - G} \sum_{g=1}^G \sum_{i=1}^{n_g} (x_{g,i} - \hat{\mu})(x_{g,i} - \hat{\mu})^T \quad (19)$$

Obtendo, assim, a função discriminante amostral dada por:

$$\widehat{d}_g(x) = x^T \widehat{\Sigma}^{-1} \bar{x}_g - \frac{1}{2} \bar{x}_g^t \widehat{\Sigma}^{-1} \bar{x}_g + \log \pi_g \quad (20)$$

### 2.4.2.3 Pipeline

Passo 1: Obter  $S_b$  e  $S_w$ .

Passo 2: Obter os valores e vetores próprios do LDA.

Passo 3: Calcular os pesos.

Passo 4: Calcular os pesos do LDA.

### 2.4.3 Quadratic Discriminant Analysis (QDA)

Análise quadrática discriminante, do inglês Quadratic Discriminant Analysis, designado de QDA, não é muito diferente do LDA descrito na secção anterior, exceto que assumimos que a matriz de covariância pode ser diferente para cada classe e, portanto, estimaremos a matriz de covariância  $\Sigma_k$  separadamente para cada classe  $k$ ,  $k = 1, 2, \dots, K$ .

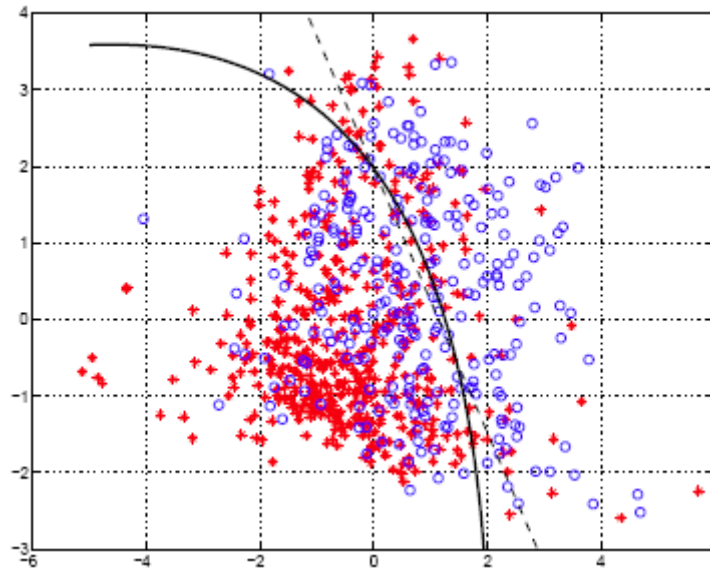
$$\delta_k(x) = -\frac{1}{2} \log |\Sigma_k| - \frac{1}{2} (x - \mu_k)^t \Sigma_k^{-1} (x - \mu_k) + \log \pi_k \quad (21)$$

A função 21 é a função discriminante quadrática é muito parecida com a função discriminante linear, exceto pelo fato de que  $\Sigma_k$ , a matriz de covariância, não é idêntica. Essa função discriminante é uma função quadrática e conterá termos de segunda ordem.

$$\hat{G}(x) = \arg \max_k \delta_k(x) \quad (22)$$

A regra de classificação definida pela equação 22 também é semelhante. Onde a classe  $k$  maximiza a função discriminante quadrática. Os limites de decisão são equações quadráticas em  $x$ .

O QDA, por permitir mais flexibilidade para a matriz de covariância, tende a ajustar os dados melhor que o LDA, mas, em seguida, possui mais parâmetros para estimar. O número de parâmetros aumenta significativamente com o QDA. Porque, com o QDA, existe uma matriz de covariância separada para cada classe. Se existir muitas classes e poucas amostras por classe, pode ser um problema, ou seja, o QDA necessita de muitas amostras por classe.



**Figura 12** A linha tracejada no gráfico é um limite de decisão fornecido pela LDA. A linha curva é o limite de decisão resultante do método QDA.

Na figura 12 é possível comparar as duas linhas do resultado dos dois métodos LDA e QDA na mesma amostra de dados.

#### 2.4.4 K - Nearest Neighbors (K-NN)

K - Vizinhos mais próximos, do inglês K - Nearest Neighbors, designado de K-NN é um algoritmo de aprendizagem supervisionada usado no campo de data mining e machine learning, é um classificador onde a aprendizagem é baseada na similaridade de dois vetores. O KNN foi proposto por Fukunaga e Narendra em 1975 (Fukunaga e Narendra, 1975). É um classificador simples de ser implementado e de fácil compreensão.

A ideia principal do KNN é determinar a identidade de uma amostra baseada nas amostras vizinhas pertencentes a um conjunto de dados de treino.

De forma a exemplificar o funcionamento do KNN, vamos analisar o exemplo da figura 13. Na figura 13 temos duas classes A e B com  $K=7$ .

São calculadas as distâncias de uma nova amostra, representada por uma estrela, às restantes amostras de treino, representadas pelos pontos azuis e amarelos. A variável  $k$  representa a quantidade de vizinhos mais próximos que serão utilizados para averiguar de qual classe a nova amostra pertence. Dos sete ( $K$ ) vizinhos de treino mais próximos da nova amostra representada pela estrela, quatro são da classe A e três da classe B, conclui-se que a nova amostra será identificada como pertencente à classe A.

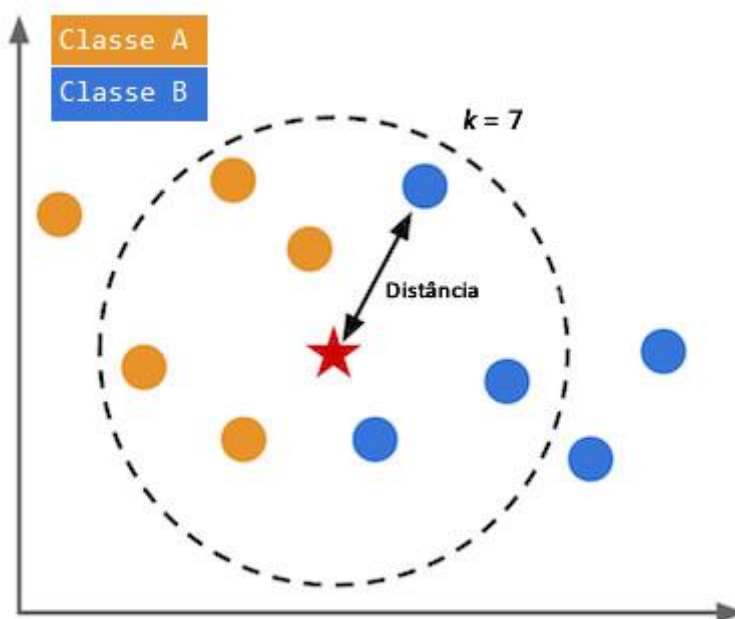


Figura 13 exemplo de classificação do KNN com duas classes e  $k = 7$

Dois pontos-chave que devem ser determinados para aplicação do KNN são: a distância e o valor de  $k$ .

Calcular a distância é fundamental para o KNN. Existem diversos métodos de cálculo da distância. A mais utilizada é a distância Euclidiana, descrita pela equação 23.

$$D_E(p, q) = \sqrt{\sum_{i=1}^n (p_i - q_i)^2} \quad (23)$$

Na equação, p e q são dois pontos de i dimensões. No exemplo da figura 13 as distâncias são calculadas entre os pontos a azul e a laranja que representam as classes de treino e a estrela vermelha que representa a nova amostra.

Em relação à escolha do valor k, não existe um valor único, é recomendável utilizar sempre valores ímpares/primos, mas o valor ótimo varia de acordo com o tipo de aplicação, onde por vezes são realizados vários testes alterando o valor de K até obtermos um resultado satisfatório para o problema.

#### 2.4.5 Support vector machine (SVM)

Uma máquina de vetores de suporte (SVM, do inglês: support vector machine) é um conceito que designa um conjunto de métodos de aprendizagem automática (do inglês: machine learning) que, por sua vez, analisam os dados e reconhecem padrões, tratando-se de algoritmos de otimização matemática baseados na aprendizagem supervisionada. Esta aplicação foi apresentada em 1992 (Boser, 1992) como um algoritmo de treino que maximiza a margem entre os padrões de treino de diferentes classes. Os resultados deste método são muito satisfatórios, mesmo quando comparados a métodos clássicos como redes neurais e árvores de decisão.

Trata-se, fundamentalmente, de um classificador de duas classes, mas que também pode ser adaptado e utilizado em mais de dois objetos. O SVM apresenta algumas vantagens, como a boa capacidade de generalização e robustez em grandes dimensões. Para encontrar a melhor solução do classificador é usada uma função quadrática, na qual não existe a presença de vários mínimos locais, mas sim de apenas um mínimo global, o que facilita a obtenção do melhor valor.

As SVM lineares com dados linearmente separáveis também são designadas por margens rígidas. O objetivo é conhecer qual o melhor hiperplano que separa

linearmente dois objetos, designado por hiperplano ótimo. Este hiperplano, por sua vez, será aquele cujas margens de separação são maiores no seu espaço geométrico. Os pontos  $x$  que se encontram sobre o hiperplano terão que satisfazer a condição (Boser, 1992):

$$w * x + b = 0 \quad (24)$$

Sendo que  $w$  corresponde ao vetor normal ao hiperplano,  $b$  à distância perpendicular do hiperplano à origem e  $w * x$  ao produto escalar entre os vetores  $w$  e  $x$ . Neste método de dados separáveis, o algoritmo do vetor de suporte escolhe o hiperplano de separação com maior margem, obedecendo assim as restrições:

$$w * x_i + b \geq +1 \text{ para } y_i = +1 \quad (25)$$

$$w * x_i + b \leq -1 \text{ para } y_i = -1 \quad (26)$$

O classificador a obter pode ser representado pela função decisão  $D(x)$ :

$$D(x) = \text{sign}(w * x + b) \quad (27)$$

A figura seguinte apresenta os dados separados e a função decisão  $D(x) = w * x + b$ . Os vetores que se encontram sobre os hiperplanos  $H_{-1}$  e  $H_1$  que formam as margens maximizadas são os vetores suporte.

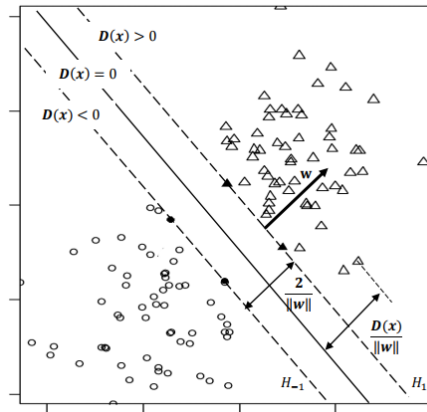


Figura 14 Caso linear separável.

O hiperplano ótimo é aquele que deriva das desigualdades  $y_i (w \cdot x_i + b) - 1 \geq 0$  em que a norma  $\|w\|$  é mínima, ou seja, um problema de otimização com as condições:

$$\min_{w,b} \frac{1}{2} \|w\|^2 \quad (28)$$

com as restrições

$$y_i (w \cdot x_i + b) - 1 \geq 0 \quad (29)$$

A função linear classificadora  $g(x)$  é obtida através de vários cálculos para os quais são utilizadas várias restrições, esta representando o hiperplano que separa os dados pela maior margem ao ter em conta aquele com maior capacidade de generalização.

$$g(x) = \text{sgn} \left( \sum_{x_i \in VS} a_i y_i (x_i \cdot x_j) + b \right) \quad (30)$$

No entanto, a existência de dados linearmente separáveis não é muito comum em problemas reais, onde a existência de diversas fontes de ruído dificulta a aplicação deste método.

Na realidade, raramente os dados são os ideais, tornando-se útil a aplicação a dados não separáveis. Para se compreender este método de dados não separáveis, torna-se necessário atribuir alguma folga às restrições. Esta folga é efetuada pela inclusão de um custo através da introdução de variáveis maiores ou iguais a zero. O hiperplano encontrado designa-se por hiperplano de margens suaves para dados lineares não separáveis.

#### 2.4.5.1 SVM de margens suaves

Considerando que os dados não podem ser processados pelo método de SVM de dados separáveis sem a existência de erro de classificação, torna-se necessário a introdução das variáveis de folga, mas de forma a minimizar os erros obtidos, conforme exposto na figura 15. As restrições passam a ter a seguinte configuração (Burgess, 1998, Cristianini e Shawe-Taylor, 2000):

$$w * x_i + b \geq +1 - \varepsilon_i \text{ para } y_i = +1 \quad (31)$$

$$w * x_i + b \leq -1 + \varepsilon_i \text{ para } y_i = -1 \quad (32)$$

A função custo passa a ser acrescentada à função objetivo na qual se aplica as novas restrições, passando a:

$$\min_{w,b} \frac{1}{2} ||w||^2 + C \sum_{i=0}^n \varepsilon_i^\delta \quad (33)$$

Onde,  $C$  corresponde a um valor que calcula a penalidade para os erros. Um  $C$  maior corresponde a um valor de erro maior.

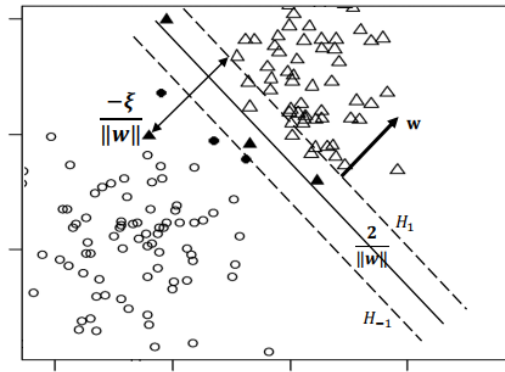


Figura 15 Caso linear não separável.

Apesar da introdução de variáveis que permitem uma folga, são suavizadas as margens, permitindo uma aplicação mais alargada do que o método das margens rígidas. Esta abordagem também apresenta limitações por ser linear.

#### 2.4.5.2 SVM não linear e “kernel trick”

O método designado de “kernel trick” permite considerar um espaço de dimensão superior onde é realizada uma separação linear, sendo que no espaço das características a superfície de separação pode não ser linear (Burgess, 1998, Mueller 2001).

Os procedimentos que servem de base a esta transformação são suportados pelas capacidades de separação estabelecidas pelo Teorema de Cover (Cover, 1965). Para simplificar todo o processo são utilizadas equações de kernel designadas por “kernel trick”, que consiste em substituir o produto interno  $x_i \cdot x_j$  por uma função  $K(x_i, x_j)$  no algoritmo de treino, sendo que o kernel  $k$  terá de satisfazer alguns pressupostos evidenciados pelo teorema de Mercer.

Atendendo às restrições já mencionadas para o método linear de margens suaves, o classificador é alterado para o formato:

$$g(x) = \text{sgn} \left( \sum_{x_i \in VS} a_i y_i \Phi(x_i) * \Phi(x_j) + b \right) \quad (34)$$

Porém, problemas de maior dimensão apresentam algumas dificuldades de controlo, por essa razão, é feita a aplicação do “kernel trick” que permite recorrer ao produto interno entre dois espaços característicos  $x$  e  $y$ . Obtém-se:

$$\Phi(x_i) * \Phi(x_j) := K(x, y) \quad (35)$$

Adicionando às restrições já mencionadas para o caso não separável, a equação (36) será dada por:

$$g(x) = \text{sgn} \left( \sum_{x_i \in VS} a_i y_i K(x_i, x_j) + b \right) \quad (36)$$

Se tivermos uma função kernel que satisfaça as condições do teorema de Mercer, podemos aplicar o “kernel trick” e obter uma superfície não-linear de separação para o problema. Normalmente, são consideradas quatro funções kernel básicas (C. W. Hsu et al., 2010):

Linear:  $K(x_i, x_j) = x_i^t x_j$ ;

Polinomial:  $K(x_i, x_j) = (\gamma x_i^t x_j + r)^d$ ,  $\gamma > 0$ ;

RBF (Radial Basis Function) gaussiana:  $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$ ,  $\gamma > 0$ ;

Sigmoidal:  $K(x_i, x_j) = \tanh(\gamma x_i^t x_j + r)$ .

sendo que  $\gamma$ ,  $r$  e  $d$  são parâmetros kernel. O número de vetores suporte necessários para usar as funções kernel polinomial, RBF ou sigmoide não diverge muito, sendo a RBF aquela que apresenta menor número de vetores na construção no modelo de decisão (Schölkopf et al., 1995).

### 2.4.5.3 Pipeline

Passo 1: Prepare o padrão de entrada.

Passo 2: Selecione a função de kernel.

Passo 3: Selecione os parâmetros da função de kernel.

Passo 4: Execute o algoritmo de treino e obtenha  $\alpha_i$ .

Passo 5: Os novos dados podem ser classificados usando  $\alpha_i$  e os vetores de suporte.

## **CAPÍTULO 3**

### **Aplicação e desenvolvimento de conhecimentos**

Neste capítulo será refletida a fase de implementação que que foram necessárias para o projeto.

#### **3.1 – Introdução**

Para que fosse possível testar os vários métodos referidos, foi criada uma base de dados de imagens RGB e dados de profundidade com o auxílio do Kinect v2, dados que foram recolhidos e processados através de uma aplicação desenvolvida em C++ utilizando o SDK do Kinect disponível pela Microsoft.

Uma vez que obtida a base de dados de imagens, de modo a comparar os vários métodos de extração de características e de classificação, foi desenvolvida uma aplicação em Matlab de forma a obter resultados experimentais, tendo como suporte a mesma base de dados de imagens adquiridas anteriormente.

#### **3.2 – Descrição do pipeline**

O projeto implementa um pipeline de reconhecimento facial bastante padrão. O pipeline consiste em quatro estágios:

1. Recolha de dados e deteção da face
2. Normalização
3. Extração de características
4. Reconhecimento / Classificação

Para o primeiro estágio da recolha e deteção das faces foi desenvolvida uma aplicação em C++ com o SDK da Kinect e o OpenCV, criando, deste modo, uma base de dados de pastas com todos os dados necessários das imagens. Para cada registo foi guardado um ficheiro de texto que identifica a quem pertencem os dados, uma imagem RGB no formato png e bmp (que são formatos sem compressão de dados de forma a que não sejam perdidos nenhuns dados das imagens), adicionalmente, foi guardada informação de profundidade em ficheiro CSV de forma a não existir compressão nem perda de informação, no qual fica guardado o valor da distância da profundidade para cada pixel da imagem RGB e, por fim, foi guardado

num ficheiro de texto as coordenadas do retângulo que representam a posição da face detetada na imagem RGB.

O segundo estágio foi pouco aprofundado, aplicando apenas uma normalização simples de redimensionamento das imagens, como demonstrado na figura 16, uma vez que o foco principal passou pelos estágios seguintes. Nos dados de profundidade a normalização consistiu essencialmente em convertermos os dados recebidos do Kinect em 16bits para 8bits, já que para a representação da face humana não são necessárias distâncias superiores a 8bits, para converter baseamos-nos pelo nariz, uma vez que na imagem da face o nariz é o ponto mais próximo da câmara, também evidenciado na figura 16.

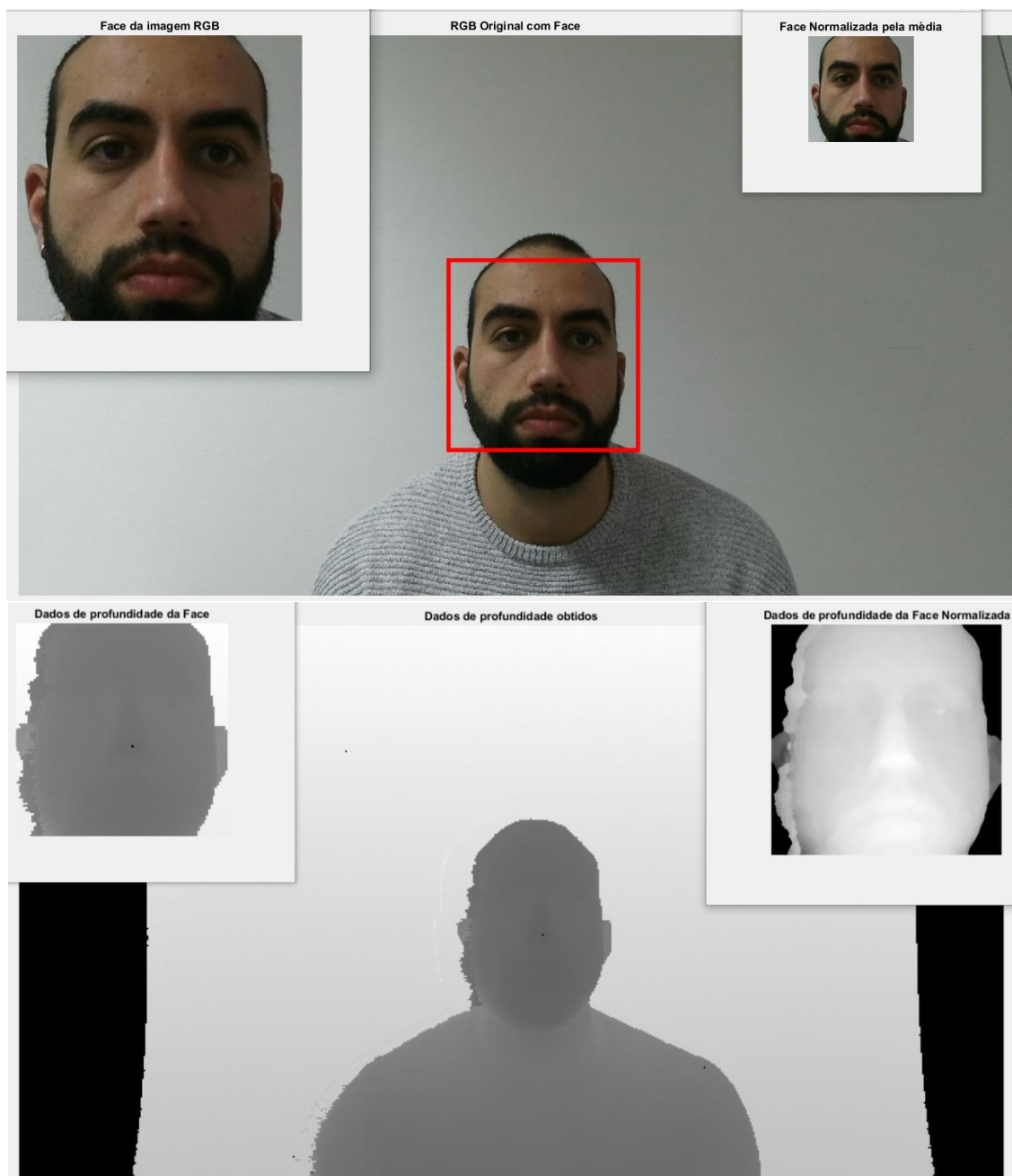
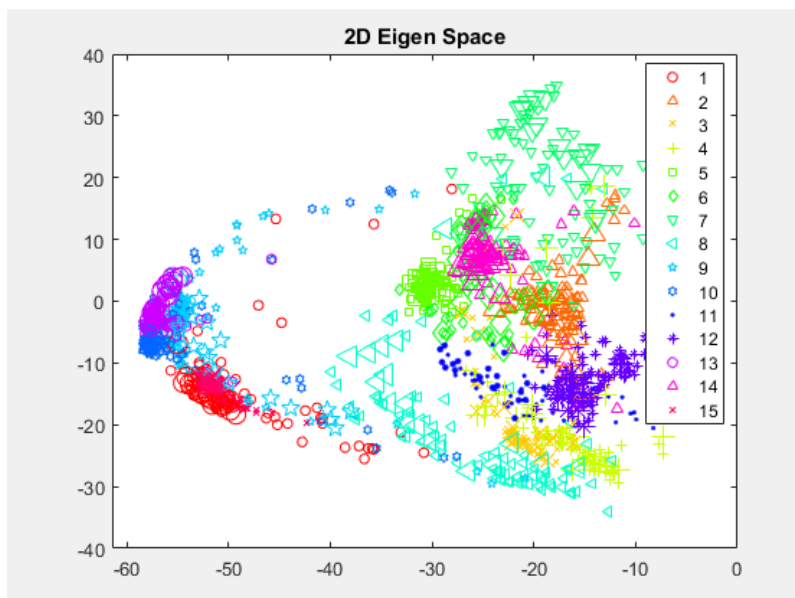


Figura 16 Exemplo de normalização das imagens da base de dados

No terceiro estágio, foram implementados métodos simples de extração de características com o uso de dados RGB e dados de profundidade sem processamento usando simplesmente os valores da cor RGB de cada pixel da imagem e as distâncias das imagens já normalizadas, e ainda métodos mais avançados de extração de características da imagem com HOG e o HONV.



**Figura 17 Representação 2D das características das 80 imagens de treino dos 15 indivíduos**

No último estágio de classificação, foi implementado o método de redução de dados PCA, como objetivo de compressão e redução de dimensões de forma a capturar o essencial da informação envolvida. A Figura 18 representa a face média do PCA, do conjunto de treino utilizado neste trabalho que serve como base para a reconstrução das diversas faces e, na figura 19, é possível observar uma face que é projetada no subespaço PCA, obtida através dos vetores próprios somados à face média. Para a classificação também foram implementados os métodos LDA, SVM, QDA e K-NN permitindo estudar e comparar os vários métodos de classificação.



**Figura 18 Média de todas as faces da base de dados usando o PCA**



Figura 19 Representação de uma face do subespaço do PCA

### 3.3 – Aquisição de dados

Para a aquisição dos dados, foi utilizado o sensor Kinect na versão 2 e desenvolvido um software na linguagem C++, usando o SDK do Kinect. Desta forma, foi possível extrair informação em imagens RGB e informação das distâncias de profundidade da imagem.

#### 3.3.1 Sensor Kinect V2

O Kinect é um sensor de deteção de movimentos, criado pela Microsoft para ser usado com os vídeos jogos da XBOX, melhorando a experiência de jogabilidade relativamente ao comando controlador da consola. O Kinect melhorou a forma de interação entre o utilizador e a consola, uma vez que o utilizador não necessita de nenhum outro dispositivo para jogar. O Kinect permite interagir com a consola XBOX através de comandos de voz e pelo simples reconhecimento dos movimentos dos jogadores.

O Kinect começou a ser comercializado em novembro de 2010, sendo considerado como "fastest selling consumer electronics device" pelo Guinness World Records, em 2011, tendo sido vendidos mais de 8 milhões de unidades nos primeiros 60 dias (Pagliari, 2011).

Devido às suas potencialidades e ao seu baixo custo, várias empresas demonstraram interesse na utilização do sensor para funções distintas daquela para a qual tinha sido criado, levando a Microsoft a expandir as funcionalidades e aplicações do seu sensor, lançando em outubro de 2011 uma versão gratuita do Kinect para Windows SDK, o que permitiu às empresas externas à Microsoft começarem a explorar o potencial do Kinect.

Com o lançamento do SDK várias empresas em todo o mundo começaram a explorar e a desenvolver aplicações, mudando drasticamente o paradigma do sensor que tinha sido criado para a indústria de vídeo jogos, sendo agora utilizado em vários

projetos completamente distintos, como a indústria cinematográfica, a saúde, o ensino, entre outras (Pagliari, 2011).

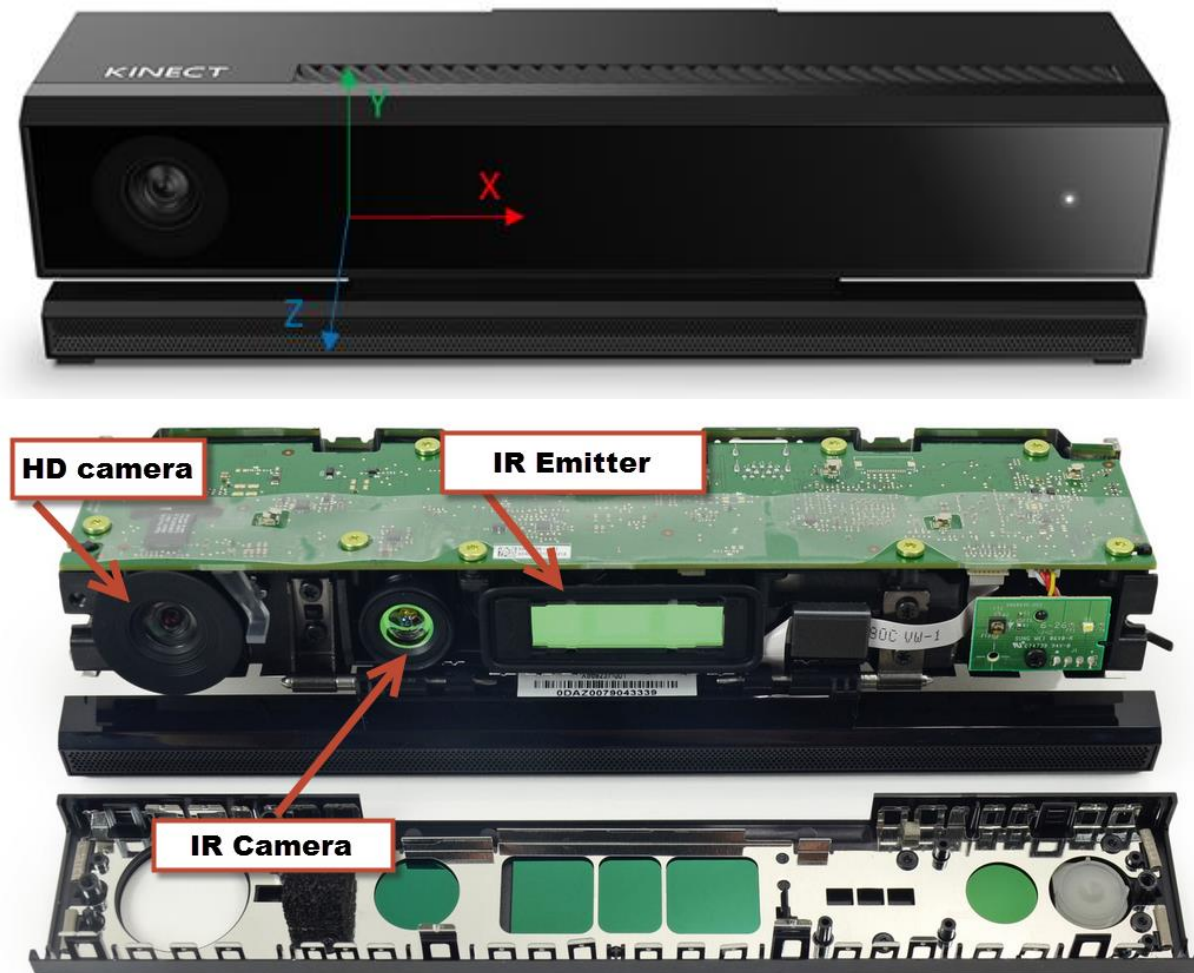


Figura 20 kinect versão 2

O Kinect foi de tal forma um sucesso que a Microsoft decidiu investir no seu sensor, tendo lançado em 2014, juntamente com a sua nova consola Xbox One, o Kinect V2, que, comparado com a versão original, sofrera uma evolução muito significativa, permitindo uma melhor qualidade de captura de informação.

### 3.3.1.1 Hardware

O Kinect V2 é formado por uma câmara RGB, um sensor infravermelho com 3 iluminadores de infravermelhos e 4 sensores de deteção de áudio. Para este projeto, os sensores de áudio não são utilizados, uma vez que o áudio não será uma

fonte de informação necessária. Nas figuras seguintes podemos visualizar o Kinect V2 e os seus sensores.

### 3.3.1.2 Características de hardware

Os sensores do Kinect v2 permitem a aquisição de informação com as seguintes características:

- Resolução do sensor IR é 512 x 424 píxeis
- Resolução do sensor RGB é 1920 x 1080 píxeis
- Campo de Visão 70 x 60 graus
- Taxa de aquisição de 30 frames por segundo
- Distâncias de aquisição de 0.5m a 4.5 m
- Dimensão do pixel (GSD) entre 1.4 mm (0.5 m) a 12 mm (4.5 m)

### 3.3.1.3 Software

A Microsoft fornece gratuitamente o Software development kit (SDK) 2.0 que contém várias ferramentas, desde a configuração até à própria gravação, através do Kinect Studio v2.0 (Pagliari, 2011). Estas ferramentas disponibilizam vários exemplos de código fonte e de executáveis para que os utilizadores consigam desenvolver conteúdos de aplicações para o Kinect da forma mais rápida e eficaz.

A lógica de funcionamento do sensor baseia-se num funcionamento em Stream, ou seja, recebe a informação e guarda a mesma em buffers. Para os casos de estudo em causa, torna-se fulcral conseguir armazenar a informação para, posteriormente, serem criadas rotinas de análise dos dados. Deste modo, foi desenvolvido um software de aquisição e processamento da informação gerada pelo Kinect V2, onde os dados são adquiridos em simultâneo, mas gravados de forma distinta, o que constitui uma grande vantagem, uma vez que permite obter a informação em formatos conhecidos, possibilitando a sua manipulação posteriormente em outros softwares como o MatLab.

### 3.3.1.4 Funcionamento do Kinect

O Kinect V2 utiliza o método tempo de propagação de sinal (Time of flight) que se baseia na diferença de tempo entre a emissão e a receção do sinal, tendo

em consideração que a velocidade da luz é constante e determina o tempo que o sinal demora a ser retornado ao sensor (Castaneda e Navab, 2011).

O Kinect usa um sistema de coordenadas cartesiano centrado no sensor IR do Kinect. O eixo Y positivo aponta para cima, o eixo Z positivo aponta para onde o Kinect está a apontar e o eixo X positivo remete para a esquerda. Uma unidade neste sistema de coordenadas é igual a um metro.

Para obter informação 3D do kinnect, temos de criar uma nuvem de pontos, utilizando os dois buffers de informação, onde um tem informação de profundidade e outro as imagens coloridas. Se sobrepor diretamente a profundidade e as imagens coloridas, de modo a que o pixel de profundidade (x, y) vai com o pixel da imagem (x, y). No entanto, isso lhe daria um mapa de profundidade de baixa qualidade, onde as bordas dos objetos não se alinham corretamente. Isso acontece porque a câmara RGB e a câmara de profundidade estão localizadas em pontos diferentes no Kinect, e desta forma as câmaras não estão exatamente a ver as mesmas coisas! Normalmente, temos que fazer algum tipo de alinhamento das duas câmaras para que possamos saber a que pixel de cor combina com o pixel de profundidade.

### 3.4 – Detetor de faces

Essa etapa é importante porque elimina da imagem informações desnecessárias. Muitos dos algoritmos de deteção facial precisam de ser testados exaustivamente antes de obterem um resultado satisfatório.

Neste trabalho, para a realização desta etapa foi utilizada a biblioteca do opencv em C++ em conjunto com a etapa anterior de aquisição de dados. O opencv usa para a deteção de faces o algoritmo de Viola-Jones (R. Brunelli e T. Poggio, 1993). Embora o algoritmo possa ser treinado para reconhecer qualquer objeto, o objetivo principal da abordagem de Viola e Jones foi o reconhecimento facial. O ponto forte deste algoritmo é a rapidez com que é executado, no entanto, possui baixas taxas de deteção para faces de perfil ou inclinadas, que tem sido estudado e melhorado ao longo dos anos.



Figura 21 Representação das coordenadas da face detetadas com o OpenCV

### 3.5 – Registo de base de dados

Para a aquisição de dados foi usada a segunda versão do sensor Kinect e desenvolvida uma aplicação em linguagem C++ com o SDK do Kinect, de forma a que fosse possível obtenção de dados do sensor, processados e armazenados em vários ficheiros.

Do sensor são recolhidos dois tipos de dados, um relativo à imagem RGB e outro relativo à distância, estes dois com resoluções diferentes. Os dados RGB encontram-se com a resolução de 1920 píxeis de largura por 1080 píxeis de altura, enquanto que os dados relativos à distância se encontram com a resolução de 512 píxeis de largura por 424 píxeis de altura, o que significa que não conseguimos fazer uma combinação direta com os dois tipos de dados, não conseguimos atribuir a distância a cada pixel da imagem RGB. Para que isso seja possível, precisamos que ambos os dados tenham a mesma resolução. Utilizando o SDK do Kinect foi possível fazer a conversão dos dados das distâncias para a mesma resolução dos dados RGB, assim sendo, tanto os dados RGB como os dados das distâncias ficaram com a resolução de 1920 píxeis por 1080 píxeis, tornando possível a combinação direta entre os dois tipos de dados. Para cada imagem é também armazenada a posição da face, em que para a sua deteção é usada a biblioteca opencv que, através do algoritmo de Viola-Jones (R. Brunelli e T. Poggio, 1993), nos indica a posição da face na imagem.

Foram recolhidos dados de 15 indivíduos voluntários, sendo que, para cada um dos indivíduos, foram guardadas 100 imagens da face diferentes, perfazendo assim uma base de dados com 1500 imagens. Para cada registo foi guardado um ficheiro texto que identifica a quem pertencem os dados, uma imagem RGB no formato png e bmp que são formatos sem compressão de dados de forma a não se perder nenhum dado das imagens, foi guardada informação de profundidade em ficheiro CSV de forma a não existir compressão nem perda de informação onde existe o valor da distância da profundidade para cada pixel da imagem RGB e, por fim, foi guardado num ficheiro texto as coordenadas do retângulo na imagem RGB no qual a face foi detetada. As imagens da base de dados através das quais foram realizados todos os testes, foram recolhidas todas no mesmo ambiente, ou seja, com iluminação igual.

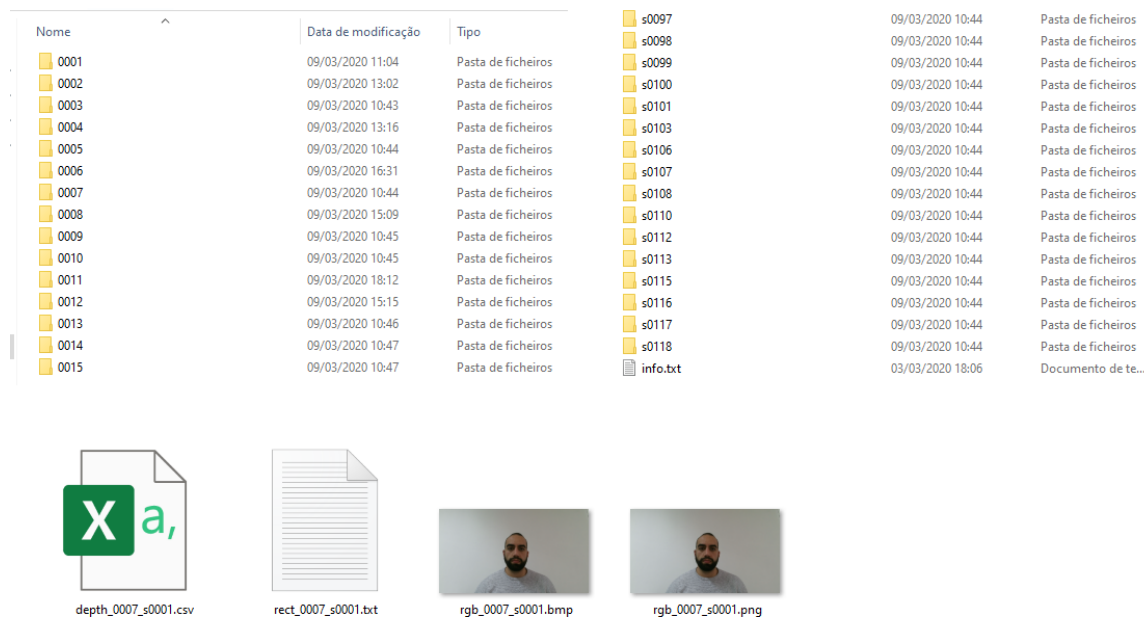


Figura 22 Base de dados de faces reconhidas com o Kinect

### 3.6 – Extração de características

Outra etapa de um sistema de reconhecimento facial é a extração de características. O objetivo passa por localizar partes da imagem que contenham características significativas. Essas partes podem ser distinguidas por texturas, formas, intensidades, propriedades, estatísticas, entre outras particularidades. De um modo geral, tenta-se extrair um conjunto de características da face. Na extração de características tentamos encontrar fatores discriminantes que permitam que cada uma das faces possa ser diferenciada matematicamente das restantes. Os métodos de extração de características abordados neste trabalho necessitam de um conjunto de faces, ou seja, funcionam de modo a representar as diferenças da face em relação às faces armazenadas na base de dados e não por caracterizar as faces independentemente.

Foram implementados em Matlab os vários métodos de extração de características, de modo a obter as diferenças de performance dos vários métodos em situações reais.

Começamos por implementar os métodos mais simples e menos complexos até chegar a implementações mais complexas, desta forma, ao longo do projeto fomos obtendo uma melhoria de resultados à medida que era aumentada a complexidade dos métodos, obtendo assim o melhor método de extração de características para o reconhecimento facial tridimensional.

O primeiro método implementado resultou ser o mais simples, extraindo os dados RGB dos pixéis da imagem e, também, os valores relativos as cores RGB por cada pixel da imagem da face, sem qualquer processamento adicional, no qual não é usada informação relativa às distâncias de profundidade da imagem. De seguida, implementamos o método mais complexo, denominado de histograma da orientação de gradientes (HOG), estudado no capítulo 2.3.1 onde extraímos informações referentes à orientação das arestas existentes na imagem da face. Uma vez implementado, testamos o método ajustando o valor do tamanho das células e o número de orientações até obtermos os melhores resultados possíveis. Não existe uma regra específica que determine quais os melhores valores para este tipo de variáveis, uma vez que depende do problema em estudo. Para tal, chegamos ao valor do número de células em três e o número de orientações em nove, uma vez que foi através destes valores que obtivemos melhores resultados com recurso a testes de tentativa e erro. Tal como no método anterior, este também não usa informação relativa às distâncias de profundidade da imagem.

Após obtermos bons resultados usando apenas as imagens RGB, passamos, agora, a tentar obter bons resultados usando apenas a informação das distâncias de profundidade das imagens da face. Para tal, tal como acontece nas imagens RGB, começamos a partir do mais simples para o mais complexo, sendo que no primeiro método usamos os dados relativos às distâncias de profundidade normalizadas. Em primeiro lugar, tivemos que lidar com o ruído dos dados de profundidade, uma vez que o Kinect devolva estes dados com “buracos”, ou seja, existe alguns pixéis que por vezes não têm qualquer valor e estes iriam corresponder a um ruído de alta frequência que afeta diretamente os dados do PCA. Para corrigir este problema aplicamos a mediana por secção de nove por nove pixéis, preenchendo com a mediana o pixel sem valor. De seguida, para normalizar os dados das distâncias, convertimos os dados recebidos do Kinect em 16bits para 8bits, uma vez que para a representação da face humana não são necessárias distâncias superiores a 8bits. Para converter baseamo-nos pelo nariz, uma vez que na imagem da face humana o nariz é o ponto mais próximo da câmara, desta forma obtemos então a distância mais curta presente na face, que corresponde à distância da câmara até ao nariz, e subtraímos a todos os pixéis da face, fazendo com que o nariz fique com distância de zero e todos os outros pontos ajustados em relação ao zero, ficando apenas com distâncias de 8bits, no qual usamos apenas dados normalizados. Posteriormente, implementamos o método mais complexo denominado de Histograma de Vetores Normais Orientados (HONV), estudado no capítulo 2.3.2, e através dos dados de profundidade normalizados anteriormente obtemos as características da geometria 3D a partir da profundidade da imagem da face. Tal como no método HOG, depois de implementado testamos o método ajustando o valor do tamanho das células e o número de orientações até obtermos os melhores resultados possíveis e chegamos, assim, ao valor do número de células em três e o número de orientações em nove,

uma vez que foi a partir destes valores que obtivemos melhores resultados recorrendo a testes de tentativa e erro.

Por último, juntamos o método HOG de extração de características da face RGB, no qual obtemos melhores resultados como o método HONV de extração de características a partir da profundidade da imagem da face, permitindo obter todos os dados para a realização do reconhecimento facial tridimensional.

### **3.7 – Classificação**

O estágio final do pipeline usa as Características faciais extraídas para realizar o reconhecimento facial. Neste projeto o objetivo é desenvolver um algoritmo capaz de realizar reconhecimento facial tridimensional credível e, para isso, foram implementados vários classificadores de forma a comparar as suas performances em situações reais. Foram implementados os métodos PCA, K-NN, LDA, QDA e SVM passo a passo como estudados no capítulo 2.

Numa primeira fase foram implementados os métodos em análise para testar a sua performance, usando apenas as imagens da base de dados recolhida para o efeito, não só para objeto de treino, mas também para teste dos algoritmos. Uma vez implementados, testamos os vários métodos ajustando o valor de algumas variáveis e parâmetros até obtermos os melhores resultados possíveis. Não existe uma regra específica que determine quais os melhores valores para este tipo de variáveis e parâmetros, uma vez que depende do problema em estudo, sendo que os testes de tentativa e erro surgem como um meio para que possamos chegar perto dos melhores valores. No PCA o melhor valor encontrado para a redução da informação foi 0.98, o que equivale a 98%, valor que posteriormente foi usado em todos os testes. No método K-NN ajustamos a valor de número de vizinhos para três, uma vez que foi com este valor que obtivemos melhores resultados. No SVM usamos o tipo linear com custo de um, já que foi através destes parâmetros que obtemos os melhores resultados para este problema.

Numa segunda fase, após analisar os resultados e concluindo qual o melhor método a aplicar para o reconhecimento facial tridimensional, foram implementados esses métodos, mas agora em tempo real, ou seja, ligado diretamente ao Kinect, em que este fica a aguardar imagens faciais e identifica se as faces que vão sendo detetadas pertencem a indivíduos da base de dados ou se são faces desconhecidas.



## **CAPÍTULO 4**

### **Resultados**

Ao longo deste último capítulo, uma série de testes será realizada para testar a viabilidade do reconhecimento facial empregando vários métodos, com e sem imagens de profundidade. Importante referir que a obtenção de imagens foi implementada em C++ com o uso do SDK do Kinect e que os vários métodos de processamento de imagem e de reconhecimento facial foram implementados em MatLab.

#### **4.1 – Descrição da base de dados**

Para a obtenção dos resultados, foram recolhidos dados de 15 indivíduos, sendo que por cada indivíduo foram recolhidas 100 imagens RGB e Depth. Destas 100 imagens, 80 serão usadas para treino dos algoritmos e 20 para teste de resultados. Do conjunto das 100 imagens podem ser feitas várias combinações de imagens de treino e de teste, respeitando sempre que para treino são usadas 80 imagens e para testes são usadas 20 imagens; no entanto, nenhuma imagem pode ser repetida, ou seja, uma imagem de treino não será uma imagem de teste.

#### **4.2 – Tabela de resultados**

As tabelas seguintes representam as matrizes de confusão, utilizadas para avaliar o resultado de cada método, com vários métodos de extração de características das imagens, utilizando características bidimensionais e tridimensionais de forma a ter um comparativo dos vários métodos, tendo em conta que o objetivo será encontrar o melhor método e a melhor extração de características para um reconhecimento tridimensional. Os dados das tabelas são representados em percentagem que, por sua vez, é calculada com base na divisão entre o total de acertos pelo total de tentativas.

##### **4.2.1 Extração de características usando GreyLevel**

Utilizando as imagens RGB da base de dados, que são convertidas em imagens GreyLevel, é extraída uma única característica, sendo essa, como o próprio

nome indica, o valor de nível do cinza. Neste estudo não são utilizados dados de profundidade das imagens, constituindo-se, assim, um estudo bidimensional.

**4.2.1.1 Matriz de Confusão do método K-NN usando GreyLevel**

	Paula	Fábio	Sérgio	Daniel	Kevin	Carlos	Hugo	Pedro	Nadine	Tânia	Daniel S	Henrique	Ana P	José	Ana C
Paula	96	0	0	0	0	0	0	0	3	1	0	0	0	0	0
Fábio	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
Sérgio	0	0	99	1	0	0	0	0	0	0	0	0	0	0	0
Daniel	0	0	2	96	0	1	0	0	0	0	0	0	0	1	0
Kevin	0	0	0	0	98	0	0	0	0	0	0	0	0	2	0
Carlos	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
Hugo	0	0	0	0	0	1	98	0	0	0	0	1	0	0	0
Pedro	0	0	0	0	0	0	0	99	0	0	0	0	0	1	0
Nadine	0	0	1	0	0	0	0	0	95	3	0	0	0	0	1
Tânia	1	0	0	0	0	0	0	0	0	97	0	0	2	0	0
Daniel S	0	0	0	0	0	1	0	0	0	0	99	0	0	0	0
Henrique	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
Ana P	0	1	0	0	0	0	0	0	0	15	0	0	84	0	0
José	0	0	0	0	0	0	0	0	0	0	0	2	0	98	0
Ana C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Número de acertos = 1459    Percentagem =  $\frac{\text{Número de acertos} \times 100}{\text{Número total de testes}} = \frac{1459 \times 100}{1500} = 97.2667\%$

**Tabela 1 Matriz de confusão do K-NN com GreyLevel**

Constatamos que, apesar de usar apenas uma característica simples como o greylevel, o método KNN obteve bons resultados acertando 97.2667% dos testes. Estes bons resultados surgem também associados ao facto de as imagens terem sido todas tiradas no mesmo ambiente com iluminação igual, sendo que se existissem imagens de ambientes diferentes este método teria provavelmente resultados inferiores. Este método acertou em 100% dos testes no Fábio, no Carlos, no Henrique e na Ana C, não tendo confundido nenhum destes indivíduos com outros

existentes na base de dados. O indivíduo com pior resultado neste método foi a Ana P, sendo que o método acertou 84% dos testes e confundiu a Ana P com a Tânia em 15% dos testes e em 1% com o Fábio.

#### 4.2.1.2 Matriz de Confusão do método LDA usando GreyLevel

	Paula	Fábio	Sérgio	Daniel	Kevin	Carlos	Hugo	Pedro	Nadine	Tânia	Daniel S	Henrique	Ana P	José	Ana C
Paula	99	0	0	0	0	0	0	0	1	0	0	0	0	0	0
Fábio	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
Sérgio	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
Daniel	0	0	1	99	0	0	0	0	0	0	0	0	0	0	0
Kevin	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
Carlos	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
Hugo	0	0	0	0	0	1	99	0	0	0	0	0	0	0	0
Pedro	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
Nadine	0	0	0	0	0	0	0	0	99	1	0	0	0	0	0
Tânia	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
Daniel S	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
Henrique	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
Ana P	0	1	0	0	0	0	0	0	0	0	0	0	99	0	0
José	0	0	1	0	0	0	0	0	0	0	0	0	0	99	0
Ana C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Número de acertos = 1494 Percentagem =  $\frac{\text{Número de acertos} \cdot 100}{\text{Número total de testes}} = \frac{1494 \cdot 100}{1500} = 99,6\%$

**Tabela 2 Matriz de Confusão de LDA usando GreyLevel**

Verificamos que, apesar de usar apenas uma característica simples, o método LDA obteve excelentes resultados, falhando em apenas seis testes. Tal como referido anteriormente, estes bons resultados são um reflexo do facto de as imagens serem todas captadas no mesmo ambiente. Este método apenas falhou nos testes em 1% dos casos, sendo que na Paula confundindo-a com a Nadine, no Daniel confundi-o com o Sérgio, no Hugo confundiu também 1% das vezes com o Carlos, a

Nadine confundida com a Tânia também 1% das vezes, a Ana P confundida com o Fábio e, por fim, o José que também foi confundido com o Sérgio.

**4.2.1.3 Matriz de Confusão do método QDA usando GreyLevel**

	Paula	Fábio	Sérgio	Daniel	Kevin	Carlos	Hugo	Pedro	Nadine	Tânia	Daniel S	Henrique	Ana P	José	Ana C
Paula	81	0	0	1	0	0	6	4	8	0	0	0	0	0	0
Fábio	0	95	0	0	0	0	5	0	0	0	0	0	0	0	0
Sérgio	0	0	93	1	0	0	1	5	0	0	0	0	0	0	0
Daniel	0	0	0	88	0	0	12	0	0	0	0	0	0	0	0
Kevin	0	0	0	6	93	0	1	0	0	0	0	0	0	0	0
Carlos	0	0	0	0	0	93	7	0	0	0	0	0	0	0	0
Hugo	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
Pedro	0	0	0	1	0	0	4	95	0	0	0	0	0	0	0
Nadine	0	0	0	0	0	0	4	6	90	0	0	0	0	0	0
Tânia	0	0	0	0	0	0	4	15	1	80	0	0	0	0	0
Daniel S	0	0	0	1	0	0	0	1	0	0	98	0	0	0	0
Henrique	0	0	5	1	0	0	1	2	0	0	0	91	0	0	0
Ana P	0	0	0	0	0	1	0	6	0	13	0	0	80	0	0
José	0	0	0	4	0	2	0	14	0	0	0	0	0	80	0
Ana C	4	0	0	0	0	0	0	1	6	3	0	0	0	0	86

Número de acertos = 1343 Percentagem =  $\frac{\text{Número de acertos} \times 100}{\text{Número total de testes}} = \frac{1343 \times 100}{1500} = 89.5333\%$

**Tabela 3 Matriz de Confusão do QDA usando GreyLevel**

Constatamos que o método QDA obteve o pior resultado dos quatro métodos utilizados com a característica de greylevel, acertando 89.5333% dos casos. Ainda assim, acertou 100% dos casos de testes para o Hugo. Os piores resultados tiveram uma taxa de erro de 20% dos testes nos indivíduos Tânia, Ana P e José. Também conseguimos verificar que neste método a maior falha dos testes são indivíduos que são confundidos com o Hugo, à exceção do Daniel S, da Ana P do José e da Ana C, todos os outros foram confundidos pelo menos uma vez com o Hugo.

4.2.1.4 Matriz de Confusão do método SVM usando GreyLevel

	Paula	Fábio	Sérgio	Daniel	Kevin	Carlos	Hugo	Pedro	Nadine	Tânia	Daniel S	Henrique	Ana P	José	Ana C
Paula	97	1	0	0	0	0	0	0	2	0	0	0	0	0	0
Fábio	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
Sérgio	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
Daniel	0	0	1	99	0	0	0	0	0	0	0	0	0	0	0
Kevin	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
Carlos	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
Hugo	0	0	0	0	0	1	99	0	0	0	0	0	0	0	0
Pedro	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
Nadine	0	0	0	0	0	0	0	0	98	2	0	0	0	0	0
Tânia	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
Daniel S	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
Henrique	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
Ana P	0	1	0	0	0	0	0	0	0	1	0	0	98	0	0
José	0	0	1	0	0	0	0	0	0	0	0	0	0	99	0
Ana C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Número de acertos = 1490 Percentagem =  $\frac{\text{Número de acertos} \times 100}{\text{Número total de testes}} = \frac{1490 \times 100}{300} = 99.3333\%$

Tabela 4 Matriz de Confusão de SVM usando GreyLevel

Juntamente com o LDA, o SVM obteve melhores resultados usando apenas uma característica simples como o greylevel. À exceção da Paula, do Daniel, do Hugo, da Nadine, da Ana P e do José, todos os outros indivíduos acertaram 100% dos testes realizados neste método. O pior resultado corresponde à Paula, tendo acertado 97% dos testes, confundida com a Nadine em 2% dos testes e com o Fábio em 1% dos testes.

#### 4.2.1.5 Extração de características usando RGB

Utilizando as imagens RGB da base de dados, são extraídas as três características presentes na imagem, sendo estas os três valores presentes nas cores vermelho, verde e azul. Comparativamente com as características anteriores, onde apenas existia uma característica por imagem, neste estudo são usadas três características por cada imagem. As imagens são as mesmas do estudo anterior; no entanto, são utilizados mais dados da imagem. Neste estudo não são utilizados dados de profundidade das imagens, sendo este um estudo bidimensional.

#### 4.2.1.6 Matriz de Confusão do método K-NN usando RGB

	Paula	Fábio	Sérgio	Daniel	Kevin	Carlos	Hugo	Pedro	Nadine	Tânia	Daniel S	Henrique	Ana P	José	Ana C
Paula	96	0	0	0	0	0	0	0	3	1	0	0	0	0	0
Fábio	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
Sérgio	0	0	99	1	0	0	0	0	0	0	0	0	0	0	0
Daniel	0	0	2	96	0	1	0	0	0	0	0	0	0	1	0
Kevin	0	0	0	0	99	0	0	0	0	0	0	0	0	1	0
Carlos	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
Hugo	0	0	0	0	0	1	98	0	0	0	0	1	0	0	0
Pedro	0	0	0	0	0	0	0	99	0	0	0	0	0	1	0
Nadine	0	0	1	0	0	0	0	0	96	3	0	0	0	0	0
Tânia	1	0	0	0	0	0	0	0	0	97	0	0	2	0	0
Daniel S	0	0	0	0	0	1	0	0	0	0	99	0	0	0	0
Henrique	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
Ana P	0	0	0	0	0	0	0	0	0	16	0	0	84	0	0
José	0	1	0	0	0	0	0	0	0	0	0	2	0	98	0
Ana C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Número de acertos = 1461 Percentagem =  $\frac{\text{Número de acertos} \cdot 100}{\text{Número total de testes}} = \frac{1461 \cdot 100}{1500} = 97.4\%$

Tabela 5 Matriz de Confusão do K-NN usando RGB

Comparativamente com o mesmo método anterior K-NN testado para uma característica, este método teve ligeiramente melhor resultado, tendo acertado 97.4% dos testes. Este método acertou 100% dos testes para os indivíduos Fábio, Carlos, Henrique e Ana C. O indivíduo com o pior resultado neste método foi a Ana P, com 84% dos testes corretos e 16% dos testes errados, confundindo 16% dos testes com a Tânia. Todos os outros indivíduos, mesmo não tendo obtido 100% dos testes corretos, tiveram percentagem de acerto superior a 90%.

#### 4.2.1.7 Matriz de Confusão do método LDA usando RGB

	Paula	Fábio	Sérgio	Daniel	Kevin	Carlos	Hugo	Pedro	Nadine	Tânia	Daniel S	Henrique	Ana P	José	Ana C
Paula	100	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Fábio	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
Sérgio	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
Daniel	0	0	1	99	0	0	0	0	0	0	0	0	0	0	0
Kevin	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
Carlos	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
Hugo	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
Pedro	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
Nadine	0	0	0	0	0	0	0	0	99	1	0	0	0	0	0
Tânia	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
Daniel S	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
Henrique	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
Ana P	0	0	0	0	0	0	0	0	1	0	0	0	99	0	0
José	0	0	1	0	0	0	0	0	0	0	0	0	0	99	0
Ana C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Número de acertos = 1496 Percentagem =  $\frac{\text{Número de acertos} \times 100}{\text{Número total de testes}} = \frac{1496 \times 100}{1500} = 99.7333\%$

Tabela 6 Matriz de Confusão do LDA usando RGB

Comparativamente com o mesmo método anterior LDA testado para uma característica, este método obteve ligeiramente melhor resultado, com 99.7333% dos testes corretos. Continua a ser melhor que o método K-NN para este tipo de dados, com este tamanho de amostras. Os indivíduos Nadine, Ana P e José tiveram 99% dos testes corretos e 1% de testes errados, sendo que o Daniel foi confundido com o Sérgio 1% das vezes, a Nadine foi confundida com a Tânia 1% das vezes, a Ana P foi confundida com a Nadine 1% das vezes e o José foi confundido com o Sérgio 1% das vezes, enquanto que os restantes indivíduos tiveram os testes todos corretos.

#### 4.2.1.8 Matriz de Confusão do método QDA usando RGB

	Paula	Fábio	Sérgio	Daniel	Kevin	Carlos	Hugo	Pedro	Nadine	Tânia	Daniel S	Henrique	Ana P	José	Ana C
Paula	80	0	0	1	0	0	6	5	8	0	0	0	0	0	0
Fábio	0	96	0	0	0	0	4	0	0	0	0	0	0	0	0
Sérgio	0	0	91	2	0	0	2	5	0	0	0	0	0	0	0
Daniel	0	0	0	88	0	0	11	1	0	0	0	0	0	0	0
Kevin	0	0	0	3	93	0	4	0	0	0	0	0	0	0	0
Carlos	0	0	0	0	0	95	5	0	0	0	0	0	0	0	0
Hugo	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
Pedro	0	0	0	3	0	0	1	96	0	0	0	0	0	0	0
Nadine	0	0	0	3	0	0	4	3	90	0	0	0	0	0	0
Tânia	0	0	0	1	0	0	4	15	0	80	0	0	0	0	0
Daniel S	0	0	0	1	0	0	1	0	0	0	98	0	0	0	0
Henrique	0	0	6	0	0	0	2	1	0	0	0	91	0	0	0
Ana P	0	0	0	0	0	1	0	6	0	13	0	0	80	0	0
José	0	0	0	3	0	2	1	14	0	0	0	0	0	80	0
Ana C	4	0	0	0	0	0	0	1	6	3	0	0	0	0	86
Número de acertos = 1344 Percentagem = $\frac{\text{Número de acertos} \times 100}{\text{Número total de testes}} = \frac{1344 \times 100}{1500} = 89.6\%$															

Tabela 7 Matriz de Confusão do QDA usando RGB

Em análise comparativa com o mesmo método anterior QDA testado para uma característica, este método teve ligeiramente melhor resultado, devido a um maior número de dados das imagens. No entanto, continua a ser o pior método para este tipo de dados com este tamanho de amostras. Este método acertou 100% dos casos de testes para o Hugo. Os piores resultados tiveram uma taxa de erro de 20% dos testes nos indivíduos Tânia, Ana P e José, sendo estes, curiosamente, os mesmos resultados obtidos no método QDA usando a característica greylevel. Adicionalmente, conseguimos também verificar que neste método a maior falha dos testes acontece nos indivíduos que são confundidos com o Hugo, já que, à exceção da Ana P e da Ana C, todos os outros foram confundidos pelo menos uma vez com o Hugo.

#### 4.2.1.9 Matriz de Confusão do método SVM usando RGB

	Paula	Fábio	Sérgio	Daniel	Kevin	Carlos	Hugo	Pedro	Nadine	Tânia	Daniel S	Henrique	Ana P	José	Ana C
Paula	97	0	0	0	0	0	0	0	3	0	0	0	0	0	0
Fábio	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
Sérgio	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
Daniel	0	0	1	99	0	0	0	0	0	0	0	0	0	0	0
Kevin	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
Carlos	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
Hugo	0	0	0	0	0	1	99	0	0	0	0	0	0	0	0
Pedro	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
Nadine	0	0	0	0	0	0	0	0	98	2	0	0	0	0	0
Tânia	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
Daniel S	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
Henrique	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
Ana P	0	1	0	0	0	0	0	0	0	1	0	0	98	0	0
José	0	0	1	0	0	0	0	0	0	0	0	0	0	99	0
Ana C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Número de acertos = 1490 Percentagem =  $\frac{\text{Número de acertos} \times 100}{\text{Número total de testes}} = \frac{1490 \times 100}{1500} = 99.3333\%$

Tabela 8 Matriz de Confusão do SVM usando RGB

Comparando com o mesmo método anterior SVM testado para a característica greylevel, este método teve o mesmo resultado com 99.3333% de testes corretos, apesar de possuir mais dados das imagens. E continua a ser, juntamente com o LDA, o melhor resultado para estas características, com este tamanho de amostras. Como nos testes do SVM para a característica greylevel, à exceção da Paula, do Daniel, do Hugo, da Nadine, da Ana P e do José, todos os outros indivíduos acertaram 100% dos testes realizados neste método. O pior resultado surgiu na Paula acertando 97% dos testes, sendo confundida com a Nadine em 3% dos testes.

Os indivíduos Daniel, Hugo e José tiveram 99% dos testes corretos e 1% de testes errados, sendo que o Daniel foi confundido com o Sérgio 1% das vezes, o Hugo foi confundido com o Carlos 1% das vezes e o José foi confundido com o Sérgio 1% das vezes. Enquanto Ana P., obteve 98% dos testes corretos e 2% de testes errados, sendo confundida com o Fábio e com a Tânia em 1% das vezes.

#### **4.2.2 Extração de características usando RGB-D**

Tal como no estudo das características anterior, são utilizadas as imagens RGB da base de dados, de onde são extraídas as três características presentes na imagem. Foi adicionado a estas três características um novo dado que se trata da informação relativa à profundidade das imagens, tendo assim agora quatro características da imagem. Ficando, desta forma, perante um estudo de reconhecimento facial tridimensional, que nos dá mais robustez e fiabilidade, ao contrário dos dois estudos anteriores sobre os quais não tínhamos informação acerca da profundidade das imagens.

Ao introduzir esta nova dimensão o sistema de reconhecimento facial fica mais flexível, uma vez que o sistema tem mais informação, o que permite reconhecer faces humanas mesmo que recolhidas em ângulos um pouco diferentes. O sistema acaba também ficar mais fiável, uma vez que se torna muito mais difícil conseguir enganar o sistema através de imagens, fotografias e fotocópias.

## 4.2.2.1 Matriz de Confusão do método K-NN usando RGB-D

	Paula	Fábio	Sérgio	Daniel	Kevin	Carlos	Hugo	Pedro	Nadine	Tânia	Daniel S	Henrique	Ana P	José	Ana C
Paula	94	0	0	0	0	0	0	0	4	2	0	0	0	0	0
Fábio	0	95	1	0	0	2	0	0	0	0	1	0	0	1	0
Sérgio	0	0	98	0	0	0	0	0	0	0	0	2	0	0	0
Daniel	0	0	0	96	1	0	3	0	0	0	0	0	0	0	0
Kevin	0	0	0	0	99	0	0	1	0	0	0	0	0	0	0
Carlos	0	0	0	0	0	96	0	0	0	2	2	0	0	0	0
Hugo	0	0	0	3	0	0	97	0	0	0	0	0	0	0	0
Pedro	0	0	2	0	0	0	0	96	0	1	0	1	0	0	0
Nadine	8	1	1	0	0	1	0	0	89	0	0	0	0	0	0
Tânia	2	0	0	0	0	0	0	1	0	97	0	0	0	0	0
Daniel S	0	0	0	0	0	0	0	0	0	0	99	1	0	0	0
Henrique	0	0	4	0	0	0	0	0	0	0	0	96	0	0	0
Ana P	1	0	0	0	0	0	0	0	0	14	0	0	84	1	0
José	0	0	0	0	0	0	0	1	0	0	0	2	0	97	0
Ana C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Número de acertos = 1433 Percentagem =  $\frac{\text{Número de acertos} \times 100}{\text{Número total de testes}} = \frac{1433 \times 100}{1500} = 95.5333\%$

Tabela 9 Matriz de Confusão do K-NN usando RGB-D

Ao adicionar os dados de profundidade, que nos dá mais robustez e fiabilidade, o método piorou comparativamente aos dois métodos K-NN anteriores, embora os resultados continuem a ser satisfatórios com 95.5333% dos testes corretos. Este método acertou 100% dos casos de testes para a Ana C. O indivíduo que obteve o pior resultado foi a Ana P com apenas 84% dos testes corretos, onde as maiores falhas foram em 14% dos testes confundidos com a Tânia, 1% dos testes confundida com o José e 1% dos testes confundidos com a Paula. A Nadine, que teve 11% de testes errados onde foi confundida com a Paula, com o Fábio, com o Sérgio e com o Carlos. Todos os restantes não mencionados tiveram testes corretos acima de 94%.

4.2.2.2 Matriz de Confusão do método LDA usando RGB-D

	Paula	Fábio	Sérgio	Daniel	Kevin	Carlos	Hugo	Pedro	Nadine	Tânia	Daniel S	Henrique	Ana P	José	Ana C
Paula	91	0	1	0	0	0	0	1	4	1	1	0	0	0	1
Fábio	0	95	2	0	0	1	0	0	1	0	1	0	0	0	0
Sérgio	1	0	98	0	0	0	0	0	0	0	0	1	1	0	0
Daniel	0	0	0	96	0	0	3	0	0	0	0	0	0	4	0
Kevin	0	0	0	0	95	0	0	0	0	0	0	1	0	0	0
Carlos	2	0	0	0	0	96	1	0	0	1	0	0	0	0	0
Hugo	0	0	1	2	0	1	96	0	0	0	0	0	0	0	0
Pedro	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
Nadine	7	0	3	0	0	0	0	1	83	3	3	0	0	0	0
Tânia	1	0	0	1	0	0	0	0	0	98	0	0	0	0	0
Daniel S	0	0	0	0	0	0	0	0	0	0	99	1	0	0	0
Henrique	0	0	3	0	0	0	0	0	0	0	0	97	0	0	0
Ana P	0	0	0	0	0	0	0	0	0	10	0	0	99	0	0
José	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
Ana C	1	0	0	0	0	0	0	0	0	0	0	0	0	0	99

Número de acertos = 1433 Percentagem =  $\frac{\text{Número de acertos} \cdot 100}{\text{Número total de testes}} = 1433 = 95.5333\%$

Tabela 10 Matriz de Confusão do LDA usando RGB-D

Ao adicionar os dados de profundidade, que nos permite mais robustez e credibilidade, o método piorou comparativamente aos dois métodos LDA anteriores, originando 95.5333% de testes corretos, sendo o método que apresentou piores resultados comparativamente com os restantes métodos. Este método obteve 100% dos testes corretos para o indivíduo Pedro e José. A Nadine com apenas 83% dos testes corretos, sendo o pior resultado, confundida com a Paula 7% dos testes, com o Sérgio, a Tânia e o Daniel 3% dos testes e com o Pedro 1% dos testes. Todos os restantes não mencionados tiveram testes corretos acima de 91%.

## 4.2.2.3 Matriz de Confusão do método QDA usando RGB-D

	Paula	Fábio	Sérgio	Daniel	Kevin	Carlos	Hugo	Pedro	Nadine	Tânia	Daniel S	Henrique	Ana P	José	Ana C
Paula	77	6	0	0	0	6	0	7	4	0	0	0	0	0	0
Fábio	0	93	1	0	0	5	0	1	0	0	0	0	0	0	0
Sérgio	0	0	91	0	0	5	0	2	0	0	0	2	0	0	0
Daniel	0	0	0	95	1	0	4	0	0	0	0	0	0	0	0
Kevin	0	0	0	0	99	1	0	0	0	0	0	0	0	0	0
Carlos	0	0	0	0	0	98	2	0	0	0	0	0	0	0	0
Hugo	0	0	0	0	0	4	93	3	0	0	0	0	0	0	0
Pedro	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
Nadine	10	0	1	0	0	6	0	2	78	0	2	1	0	0	0
Tânia	0	8	0	1	0	2	0	3	0	86	0	0	0	0	0
Daniel S	0	0	0	0	0	2	0	1	0	0	97	0	0	0	0
Henrique	0	0	5	0	0	2	0	1	0	0	0	92	0	0	0
Ana P	0	0	0	0	0	3	0	4	0	11	0	0	82	0	0
José	0	2	0	0	0	1	0	17	0	0	0	0	0	80	0
Ana C	0	1	0	0	0	0	0	2	8	1	0	0	0	0	88
Número de acertos = 1349 Percentagem = $\frac{\text{Número de acertos} \times 100}{\text{Número total de testes}} = \frac{1349 \times 100}{1500} = 89.9333\%$															

Tabela 11 Matriz de Confusão do QDA usando RGB-D

Ao contraio dos métodos anteriores, o método melhorou ligeiramente comparativamente aos dois métodos QDA anteriores, ficando com 89.9333% de testes corretos. Este método alcançou 100% dos testes corretos para o individuo Pedro, sendo este o melhor resultado obtido com este método. O pior resultado surgiu na Paula com 77% dos testes corretos, onde no seu pior resultado foi confundida com o Pedro 7% dos testes, a Paula também foi confundida com o Fábio e Carlos 6% dos testes e com a Nadine em 4% dos testes.

## 4.2.2.4 Matriz de Confusão do método SVM usando RGB-D

	Paula	Fábio	Sérgio	Daniel	Kevin	Carlos	Hugo	Pedro	Nadine	Tânia	Daniel S	Henrique	Ana P	José	Ana C
Paula	96	0	0	0	0	0	0	0	1	3	0	0	0	0	0
Fábio	2	96	0	0	0	0	0	0	1	1	0	0	0	0	0
Sérgio	0	0	99	0	0	0	0	0	0	0	0	1	0	0	0
Daniel	0	0	0	95	0	2	3	0	0	0	0	0	0	0	0
Kevin	1	0	0	0	99	0	0	0	0	0	0	0	0	0	0
Carlos	0	1	0	0	0	97	0	0	0	0	2	0	0	0	0
Hugo	0	0	0	3	0	3	92	2	0	0	0	0	0	0	0
Pedro	0	0	0	0	0	0	0	99	0	1	0	0	0	0	0
Nadine	0	1	0	0	0	3	0	0	96	0	0	0	0	0	0
Tânia	0	0	0	1	0	0	0	0	0	99	0	0	0	0	0
Daniel S	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
Henrique	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
Ana P	1	0	0	0	0	0	0	0	0	1	0	0	98	0	0
José	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
Ana C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Número de acertos = 1466 Percentagem =  $\frac{\text{Número de acertos} \times 100}{\text{Número total de testes}} = \frac{1466 \times 100}{1500} = 97.7333\%$

Tabela 12 Matriz de Confusão do SVM usando RGB-D

Ao adicionar os dados de profundidade verificamos que todos os métodos a exceção do QDA pioraram; no entanto, relativamente aos métodos com melhores resultados o método SVM foi o método que piorou menos, tornando-se assim o melhor para este tipo de características, com 97.7333% de testes corretos. Este método acertou 100% dos casos de testes para a Ana C, para o José, para o Henrique e para o Daniel. O indivíduo que obteve o pior resultado foi o Hugo com 92% dos testes corretos, onde as maiores falhas aconteceram em 3% dos testes confundidos com o Carlos e Daniel, e 2% dos testes confundidos com o Pedro. Todos os indivíduos tiveram testes corretos acima de 90%.

### 4.2.3 Extração de características usando HOG

Utilizando as imagens RGB da base de dados, são extraídas as trinta e uma características presentes na imagem através do método HOG. Este método extrai informações referentes à orientação das arestas existentes na imagem. Neste estudo não são utilizados dados de profundidade das imagens, sendo este um estudo bidimensional.

#### 4.2.3.1 Matriz de Confusão do método K-NN usando HOG

	Paula	Fábio	Sérgio	Daniel	Kevin	Carlos	Hugo	Pedro	Nadine	Tânia	Daniel S	Henrique	Ana P	José	Ana C
Paula	98	0	0	0	0	0	0	0	2	0	0	0	0	0	0
Fábio	0	99	0	1	0	0	0	0	0	0	0	0	0	0	0
Sérgio	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
Daniel	0	0	0	99	1	0	0	0	0	0	0	0	0	0	0
Kevin	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
Carlos	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
Hugo	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
Pedro	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
Nadine	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
Tânia	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
Daniel S	0	0	0	0	0	1	0	0	0	0	99	0	0	0	0
Henrique	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
Ana P	0	0	0	0	0	0	0	0	0	1	0	0	99	0	0
José	0	0	1	0	0	0	0	0	0	0	0	0	0	99	0
Ana C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Número de acertos = 1493 Percentagem =  $\frac{\text{Número de acertos} \times 100}{\text{Número total de testes}} = \frac{1493 \times 100}{1500} = 99.5333\%$

Tabela 13 Matriz de Confusão do K-NN usando HOG

Comparativamente com as características anteriores utilizando apenas imagens RGB, o método K-NN apresentou ligeiras melhorias, tendo errado apenas sete testes, ficando com 99.5333% dos testes assertivos. À exceção da Paula, do Fábio, do Daniel, do Daniel S, da Ana P e do José, todos os outros indivíduos acertaram 100% dos testes realizados neste método. O pior resultado aconteceu na Paula acertando 98% dos testes, sendo confundida com a Nadine em 2% dos testes.

**4.2.3.2 Matriz de Confusão do método LDA usando HOG**

	Paula	Fábio	Sérgio	Daniel	Kevin	Carlos	Hugo	Pedro	Nadine	Tânia	Daniel S	Henrique	Ana P	José	Ana C
Paula	97	0	0	0	0	2	0	0	1	0	0	0	0	0	0
Fábio	0	98	0	0	0	2	0	0	0	0	0	0	0	0	0
Sérgio	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
Daniel	0	0	0	98	0	0	0	0	0	0	2	0	0	0	0
Kevin	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
Carlos	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
Hugo	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
Pedro	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
Nadine	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
Tânia	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
Daniel S	0	1	0	0	0	0	0	0	0	0	99	0	0	0	0
Henrique	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
Ana P	0	0	0	0	0	0	0	0	0	4	0	0	96	0	0
José	0	0	1	0	0	0	0	0	0	0	0	0	0	99	0
Ana C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Número de acertos = 1487 Percentagem =  $\frac{\text{Número de acertos} \times 100}{\text{Número total de testes}} = \frac{1487 \times 100}{1500} = 99.1333\%$

**Tabela 14 Matriz de Confusão do LDA usando HOG**

Em comparação com as características anteriores utilizando apenas imagens RGB, o método LDA apresentou pioras ligeiras, mas, ainda assim, continuou acima

de 99% com 99.1333%. Tal como no método anterior, à exceção da Paula, do Fábio, do Daniel, do Daniel S, da Ana P e do José, todos os outros indivíduos acertaram 100% dos testes realizados neste método. O pior resultado surgiu na Ana P, acertando 96% dos testes, sendo confundida com a Tânia em 4% dos testes.

#### 4.2.3.3 Matriz de Confusão do método QDA usando HOG

	Paula	Fábio	Sérgio	Daniel	Kevin	Carlos	Hugo	Pedro	Nadine	Tânia	Daniel S	Henrique	Ana P	José	Ana C
Paula	87	4	5	3	0	0	0	0	1	0	0	0	0	0	0
Fábio	0	85	8	0	0	7	0	0	0	0	0	0	0	0	0
Sérgio	0	24	72	0	0	0	0	0	3	1	0	0	0	0	0
Daniel	0	25	1	63	0	1	0	0	0	9	1	0	0	0	0
Kevin	0	0	1	1	91	0	0	3	1	2	0	0	0	1	0
Carlos	0	15	0	0	0	85	0	0	0	0	0	0	0	0	0
Hugo	0	17	1	0	0	1	78	0	0	3	0	0	0	0	0
Pedro	0	17	0	5	0	0	0	77	0	1	0	0	0	0	0
Nadine	0	5	0	4	0	0	0	0	86	0	0	5	0	0	0
Tânia	0	2	0	5	0	0	0	5	1	85	0	2	0	0	0
Daniel S	0	10	13	1	0	53	3	0	0	2	3	14	1	0	0
Henrique	0	9	1	0	0	0	1	0	1	0	0	88	0	0	0
Ana P	0	3	0	1	0	0	0	1	6	11	0	2	76	0	0
José	0	2	2	3	0	1	1	0	1	0	0	2	0	88	0
Ana C	0	0	0	0	0	0	0	0	3	5	0	1	0	0	91

Número de acertos = 1155 Percentagem =  $\frac{\text{Número de acertos} \times 100}{\text{Número total de testes}} = \frac{1155 \times 100}{1500} = 77\%$

**Tabela 15 Matriz de Confusão do QDA usando HOG**

Comparativamente com as características anteriores utilizando apenas imagens RGB, o método QDA piorou bastante, cerca de 12,6%, relativamente aos métodos usando características da imagem RGB, ficando com 77% de casos assertivos. Este método também não obteve nenhum individuo com 100% dos testes

corretos, sendo o melhor resultado aquele observado nos indivíduos Kevin e Ana C com 91% dos testes assertivos. O pior resultado surgiu no Daniel S com apenas 3% dos testes assertivos, sendo nos seus piores resultados confundido com o Carlos 53% dos testes, com o Henrique 14% dos testes, com o Sérgio 13% dos testes e com o Fábio 10% dos testes.

#### 4.2.3.4 Matriz de Confusão do método SVM usando HOG

	Paula	Fábio	Sérgio	Daniel	Kevin	Carlos	Hugo	Pedro	Nadine	Tânia	Daniel S	Henrique	Ana P	José	Ana C
Paula	99	0	0	0	0	0	0	0	1	0	0	0	0	0	0
Fábio	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
Sérgio	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
Daniel	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
Kevin	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
Carlos	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
Hugo	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
Pedro	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
Nadine	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
Tânia	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
Daniel S	0	1	0	0	0	0	0	0	0	0	99	0	0	0	0
Henrique	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
Ana P	0	0	0	0	0	0	0	0	1	0	0	0	99	0	0
José	0	0	1	0	0	0	0	0	0	0	0	0	0	99	0
Ana C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Número de acertos = 1496 Percentagem =  $\frac{\text{Número de acertos} \times 100}{\text{Número total de testes}} = \frac{1496 \times 100}{1500} = 99.7333\%$

Tabela 16 Matriz de Confusão do SVM usando HOG

O método SVM continuou a ser o mais assertivo e manteve o seu excelente resultado com 99.7333% de testes assertivos. À exceção da Paula, do Daniel S, da Ana P e do José, que obtiveram 99% dos testes assertivos, sendo confundidos

apenas 1% dos testes, todos os outros indivíduos acertaram 100% dos testes realizados neste método.

#### 4.2.4 Extração de características usando HONV

Utilizando os dados de profundidade das imagens da base de dados, são extraídas as dezoito características presentes na imagem através do método HONV. Este método obtém as características da geometria tridimensional da imagem.

##### 4.2.4.1 Matriz de Confusão do método K-NN usando HONV

	Paula	Fábio	Sérgio	Daniel	Kevin	Carlos	Hugo	Pedro	Nadine	Tânia	Daniel S	Henrique	Ana P	José	Ana C
Paula	97	0	0	0	0	0	0	0	0	1	0	0	0	2	0
Fábio	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
Sérgio	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
Daniel	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
Kevin	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
Carlos	0	0	0	0	0	99	0	0	0	0	1	0	0	0	0
Hugo	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
Pedro	0	0	0	0	0	0	0	99	0	0	0	0	0	1	0
Nadine	2	0	0	0	0	0	0	0	98	0	0	0	0	0	0
Tânia	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
Daniel S	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
Henrique	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
Ana P	0	0	0	0	0	0	0	0	0	1	0	0	99	0	0
José	0	0	0	0	0	0	0	1	0	0	0	0	0	99	0
Ana C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Número de acertos = 1491 Percentagem =  $\frac{\text{Número de acertos} \cdot 100}{\text{Número total de testes}} = \frac{1491 \cdot 100}{1500} = 99,4\%$

Tabela 17 Matriz de Confusão do K-NN usando HONV

Tendo em conta que este teste tem apenas características de profundidade, mostrou-se muito eficaz, tendo uma percentagem de acerto de 99.4%. À exceção da

Paula, do Carlos, do Pedro, da Nadine, da Ana P e do José, todos os outros indivíduos acertaram 100% dos testes realizados neste método. O indivíduo com pior resultado neste método foi a Paula, onde o método acertou 97% dos testes, e confundiu a Paula com a Tânia em 1% das vezes e 2% com o José.

**4.2.4.2 Matriz de Confusão do método LDA usando HONV**

	Paula	Fábio	Sérgio	Daniel	Kevin	Carlos	Hugo	Pedro	Nadine	Tânia	Daniel S	Henrique	Ana P	José	Ana C
Paula	88	2	4	0	0	2	0	2	1	0	0	0	1	0	0
Fábio	0	85	5	0	0	1	0	1	1	2	0	3	1	0	1
Sérgio	0	0	96	0	0	0	0	1	1	0	0	1	0	1	0
Daniel	1	0	2	81	0	3	1	5	2	0	2	0	0	1	2
Kevin	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
Carlos	3	3	1	1	0	80	1	2	2	0	6	0	0	1	0
Hugo	3	1	4	3	0	3	71	1	3	1	1	3	1	4	1
Pedro	0	5	3	0	0	0	0	87	1	1	2	0	0	1	0
Nadine	4	1	1	0	0	1	0	1	84	0	2	4	1	1	0
Tânia	0	1	1	0	0	1	1	2	1	87	0	1	3	0	2
Daniel S	0	2	0	0	0	1	0	2	0	0	93	2	0	0	0
Henrique	0	0	0	0	0	0	0	0	0	0	1	97	0	1	1
Ana P	0	6	0	0	0	0	0	0	0	2	0	0	90	1	1
José	0	1	1	0	0	0	0	0	0	0	2	0	0	95	1
Ana C	0	0	0	0	0	0	0	0	0	1	0	2	0	0	97

Número de acertos = 1331 Percentagem =  $\frac{\text{Número de acertos} \times 100}{\text{Número total de testes}} = \frac{1331 \times 100}{1500} = 88.7333\%$

**Tabela 18 Matriz de Confusão do LDA usando HONV**

Este método demonstrou ser ligeiramente pior que o K-NN, tendo uma percentagem de asserto de 88.7333%. Ainda assim, acertou 100% dos casos de testes para o Kevin. O pior resultado foi o do individuo Hugo, com apenas 71% dos testes assertivos, tendo sido confundido em 3% dos testes com a Paula, com o

Daniel, com a Nadine e com o Henrique, em 1% dos testes como o Fábio, com o Pedro, com a Tânia, com o Daniel S., com a Ana P. e com a Ana C., e, por fim, em 4% dos testes com o Sérgio e com o José. Além do Hugo todos os outros indivíduos obtiveram resultados aceitáveis.

#### 4.2.4.3 Matriz de Confusão do método QDA usando HONV

	Paula	Fábio	Sérgio	Daniel	Kevin	Carlos	Hugo	Pedro	Nadine	Tânia	Daniel S	Henrique	Ana P	José	Ana C
Paula	78	5	0	0	0	0	0	11	6	0	0	0	0	0	0
Fábio	0	93	0	7	0	0	0	0	0	0	0	0	0	0	0
Sérgio	0	1	92	0	0	0	0	6	1	0	0	0	0	0	0
Daniel	0	3	0	41	0	40	16	0	0	0	0	0	0	0	0
Kevin	0	0	9	0	91	0	0	0	0	0	0	0	0	0	0
Carlos	0	7	0	0	0	85	5	2	1	0	0	0	0	0	0
Hugo	0	0	0	0	0	16	84	0	0	0	0	0	0	0	0
Pedro	0	0	0	4	0	0	0	96	0	0	0	0	0	0	0
Nadine	0	2	3	1	0	0	3	1	88	2	0	0	0	0	0
Tânia	0	4	0	0	0	0	0	3	5	88	0	0	0	0	0
Daniel S	0	0	3	0	0	53	0	35	0	0	6	3	0	0	0
Henrique	0	5	0	0	0	0	0	0	4	0	0	91	0	0	0
Ana P	0	3	8	1	0	0	0	10	0	1	0	0	77	0	0
José	0	12	0	0	0	0	0	12	0	0	0	0	0	76	0
Ana C	0	5	2	0	0	0	0	0	2	0	0	0	0	0	91

Número de acertos = 1177 Percentagem =  $\frac{\text{Número de acertos} \times 100}{\text{Número total de testes}} = \frac{1177 \times 100}{1500} = 78.4667\%$

**Tabela 19 Matriz de Confusão do QDA usando HONV**

Este método demonstrou ser o pior para este tipo de características com este número de amostras, tendo uma percentagem de acerto de apenas 78.4667%, estando muito a baixo dos outros métodos. Este método não obteve nenhum indivíduo com 100% dos testes assertivos, sendo o melhor resultado referente ao

indivíduo Pedro com 96% dos testes assertivos. O pior resultado corresponde ao Daniel S com apenas 6% dos testes assertivos, sendo nos seus piores resultados confundido com o Carlos 53% dos testes, com o Pedro 35% dos testes, e ainda com o Henrique e com o Sérgio em 3% dos testes.

#### 4.2.4.4 Matriz de Confusão do método SVM usando HONV

	Paula	Fábio	Sérgio	Daniel	Kevin	Carlos	Hugo	Pedro	Nadine	Tânia	Daniel S	Henrique	Ana P	José	Ana C
Paula	97	1	0	0	0	0	0	0	0	2	0	0	0	0	0
Fábio	0	99	0	0	0	0	0	0	1	0	0	0	0	0	0
Sérgio	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
Daniel	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
Kevin	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
Carlos	0	1	0	0	0	99	0	0	0	0	0	0	0	0	0
Hugo	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
Pedro	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
Nadine	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
Tânia	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
Daniel S	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
Henrique	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
Ana P	0	0	0	0	0	0	0	1	0	0	0	0	99	0	0
José	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
Ana C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Número de acertos = 1494 Percentagem =  $\frac{\text{Número de acertos} \times 100}{\text{Número total de testes}} = \frac{1494 \times 100}{1435} = 99.6\%$

Tabela 20 Matriz de Confusão do SVM usando HONV

O método SVM mostrou ser, mais uma vez que, o melhor método para este tipo de características, tendo uma percentagem de acerto de 99.6%. Este método acertou 100% dos casos de testes para diversos indivíduos. O indivíduo que obteve o pior resultado foi a Paula com 97% dos testes assertivos, tendo as falhas

acontecido em 1% dos testes confundidos com o Fábio e 2% dos testes confundidos com a Tânia. À exceção da Paula, todos os restantes tiveram testes assertivos acima de 99%.

#### 4.2.5 Extração de características usando HOG com HONV

Combinando as características dos dois testes anteriores feitos com o HOG e com o HONV, obtemos, assim, quarenta e nove características, ficando, desta forma, perante um estudo de reconhecimento facial tridimensional, que permite mais robustez e fiabilidade.

##### 4.2.5.1 Matriz de Confusão do método K-NN usando HOG com HONV

	Paula	Fábio	Sérgio	Daniel	Kevin	Carlos	Hugo	Pedro	Nadine	Tânia	Daniel S	Henrique	Ana P	José	Ana C
Paula	97	0	0	0	0	0	0	0	0	1	0	0	0	2	0
Fábio	0	99	0	0	0	0	0	0	0	0	1	0	0	0	0
Sérgio	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
Daniel	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
Kevin	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
Carlos	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
Hugo	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
Pedro	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
Nadine	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
Tânia	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
Daniel S	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
Henrique	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
Ana P	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0
José	0	0	0	0	0	0	0	1	0	0	0	0	0	99	0
Ana C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Número de acertos = 1495 Percentagem =  $\frac{\text{Número de acertos} \times 100}{\text{Número total de testes}} = \frac{1495 \times 100}{1500} = 99.6667\%$

**Tabela 21 Matriz de Confusão do K-NN usando HOG com HONV**

O método K-NN continua a apresentar-se como um método aceitável para este tipo de estudo, estando próximo do método com melhor resultado, tendo obtido uma percentagem de 99.6667% de testes assertivos. Todos os indivíduos, à exceção da Paula, do Fábio e do José obtiveram excelentes resultados com testes assertivos de 100%. O pior resultado foi na Paula com 97% dos testes assertivos, sendo confundido com o José em 2% dos testes e com a Tânia em 1% dos testes.

**4.2.5.2 Matriz de Confusão do método LDA usando HOG com HONV**

	Paula	Fábio	Sérgio	Daniel	Kevin	Carlos	Hugo	Pedro	Nadine	Tânia	Daniel S	Henrique	Ana P	José	Ana C
Paula	96	0	0	0	0	0	0	0	3	0	0	1	0	0	0
Fábio	0	97	2	0	0	0	0	0	0	0	0	0	0	1	0
Sérgio	0	1	99	0	0	0	0	0	0	0	0	0	0	0	0
Daniel	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
Kevin	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
Carlos	0	1	0	0	0	99	0	0	0	0	0	0	0	0	0
Hugo	0	0	1	0	0	0	99	0	0	0	0	0	0	0	0
Pedro	0	1	1	0	0	0	0	97	0	0	0	0	0	1	0
Nadine	0	0	1	0	0	0	0	0	97	2	0	0	0	0	0
Tânia	1	1	0	0	0	0	0	1	2	95	0	0	0	0	0
Daniel S	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
Henrique	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
Ana P	2	0	0	0	0	0	0	0	0	2	0	0	95	1	0
José	0	0	0	0	0	0	0	2	0	0	0	0	0	98	0
Ana C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Número de acertos = 1472 Percentagem =  $\frac{\text{Número de acertos} \times 100}{\text{Número total de testes}} = \frac{1472 \times 100}{1500} = 98.1333\%$

**Tabela 22 Matriz de Confusão do LDA usando HOG com HONV**

O método LDA continua a ser um método aceitável para este tipo de estudo, com 98.1333% dos testes assertivos. Todos os indivíduos detiveram resultados assertivos superiores a 90%. O pior resultado surgiu nos indivíduos Tânia e Ana P. com 95% dos testes assertivos. A Tânia foi confundida em 2% dos testes com a Nadine, em 1% dos testes com a Paula, com o Fábio e com o Hugo. No caso da Ana P., foi confundida em 2% dos testes com a Paula e com a Tânia e em 1% dos testes com o José.

#### 4.2.5.3 Matriz de Confusão do método QDA usando HOG com HONV

	Paula	Fábio	Sérgio	Daniel	Kevin	Carlos	Hugo	Pedro	Nadine	Tânia	Daniel S	Henrique	Ana P	José	Ana C
Paula	82	11	0	1	0	0	0	0	6	0	0	0	0	0	0
Fábio	0	85	0	10	0	2	0	0	2	1	0	0	0	0	0
Sérgio	0	17	83	0	0	0	0	0	0	0	0	0	0	0	0
Daniel	0	16	0	45	0	39	0	0	0	0	0	0	0	0	0
Kevin	0	0	6	0	91	0	0	3	0	0	0	0	0	0	0
Carlos	0	10	0	3	0	85	2	0	0	0	0	0	0	0	0
Hugo	0	1	2	1	0	16	80	0	0	0	0	0	0	0	0
Pedro	0	13	0	6	0	0	0	81	0	0	0	0	0	0	0
Nadine	0	13	0	2	0	0	0	0	85	0	0	0	0	0	0
Tânia	0	18	0	0	0	0	0	1	0	81	0	0	0	0	0
Daniel S	0	12	12	1	0	53	0	9	4	0	6	3	0	0	0
Henrique	0	14	0	0	0	0	0	0	0	0	0	86	0	0	0
Ana P	0	7	1	1	0	0	0	1	6	10	0	0	74	0	0
José	0	4	3	0	0	0	0	8	1	1	0	0	0	83	0
Ana C	0	2	0	0	0	0	0	0	6	0	0	0	0	0	92

Número de acertos = 1139 Percentagem =  $\frac{\text{Número de acertos} \times 100}{\text{Número total de testes}} = \frac{1139 \times 100}{1500} = 75.9333\%$

Tabela 23 Matriz de Confusão do QDA usando HOG com HONV

O método QDA apresenta-se como o pior método para este tipo de estudo, estando muito abaixo dos resultados apresentados pelos outros métodos, com apenas 75.9333% de testes assertivos. Neste método surgiu um resultado desastroso para o Daniel S, uma vez que apenas 6% dos testes foram assertivos, tendo uma grande parte dos testes, 53% dos testes mais precisamente, o confundido com o Carlos. Existem apenas dois testes com percentagem de assertividade superior a 90%, todos os restantes indivíduos têm uma percentagem de testes assertivos a rondar os 70% e os 80%.

#### 4.2.5.4 Matriz de Confusão do método SVM usando HOG com HONV

	Paula	Fábio	Sérgio	Daniel	Kevin	Carlos	Hugo	Pedro	Nadine	Tânia	Daniel S	Henrique	Ana P	José	Ana C
Paula	99	0	0	0	0	0	0	0	0	1	0	0	0	0	0
Fábio	0	100	0	0	0	0	0	0	0	0	0	0	0	0	0
Sérgio	0	0	100	0	0	0	0	0	0	0	0	0	0	0	0
Daniel	0	0	0	100	0	0	0	0	0	0	0	0	0	0	0
Kevin	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
Carlos	0	0	0	0	0	100	0	0	0	0	0	0	0	0	0
Hugo	0	0	0	0	0	0	100	0	0	0	0	0	0	0	0
Pedro	0	0	0	0	0	0	0	100	0	0	0	0	0	0	0
Nadine	0	0	0	0	0	0	0	0	100	0	0	0	0	0	0
Tânia	0	0	0	0	0	0	0	0	0	100	0	0	0	0	0
Daniel S	0	0	0	0	0	0	0	0	0	0	100	0	0	0	0
Henrique	0	0	0	0	0	0	0	0	0	0	0	100	0	0	0
Ana P	0	0	0	0	0	0	0	0	0	0	0	0	100	0	0
José	0	0	0	0	0	0	0	0	0	0	0	0	0	100	0
Ana C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	100

Número de acertos = 1499 Percentagem =  $\frac{\text{Número de acertos} \times 100}{\text{Número total de testes}} = \frac{1499 \times 100}{1500} = 99.9333\%$

Tabela 24 Matriz de Confusão do SVM usando HOG com HONV

O método SVM continua a ser o método que apresenta melhores resultados, tendo uma percentagem de acerto de 99.9333%. À exceção da Paula, todos os outros indivíduos acertaram 100% dos testes realizados neste método. O pior resultado foi observado na Paula acertando 99% dos testes, sendo confundido com a Tânia em 1% dos testes.

#### 4.3 – Tabela de Conclusão de resultados

		<b>K-NN</b>	<b>LDA</b>	<b>QDA</b>	<b>SVM</b>
<b>2D</b>	<b>GreyLevel</b>	97.2667%	99.6%	89.5333%	99.3333%
	<b>RGB</b>	97.4%	99.7333%	89.7333%	99.3333%
	<b>HoG</b>	99.5333%	99.1333%	77%	99.7333%
<b>3D</b>	<b>RGB-D</b>	95.5333%	95.5333%	89.9333%	97.7333%
	<b>HoNV</b>	99.4%	88.7333%	78.4667%	99.6%
	<b>HoG + HoNV</b>	99.6667%	98.1333%	75.9333%	99.9333%

Tabela 25 Tabela de Conclusão de resultados

Como podemos observar na presente tabela de resumo de resultados, o método SVM foi sempre o melhor neste tipo de estudo e o QDA o pior, tendo em conta o tamanho da nossa amostra. No entanto, verificamos que ao acrescentar os dados de profundidade das imagens, alguns dos resultados pioram ligeiramente, cerca de 1% a 5%, mas, ainda assim, obtivemos melhorias em alguns dos métodos como no caso do HOG com HONV no SVM que obtivemos o melhor resultado com 99.9333% de testes assertivos.

Podemos evidenciar que usando apenas dados bidimensionais, para o método de extração de características greylevel o melhor método de classificação corresponde ao LDA, com 99.6% de testes assertivos, para o método de extração de características RGB o melhor método de classificação foi também o LDA, com 99.7333% de testes assertivos, e para o método de extração de características HOG

o melhor método de classificação foi o SVM, com 99.7333% de testes assertivos. Concluindo, a melhor combinação de métodos para reconhecimento facial com apenas dados bidimensionais é a combinação da extração de características RGB com o classificador LDA e também a extração de características da HOG com o classificador SVM com 99.7333% de casos assertivos.

Para os casos onde são usados dados tridimensionais, para o método de extração de características RGB-D o melhor método de classificação é o SVM, com 97.7333% de testes assertivos, para o método de extração de características HONV o melhor método de classificação corresponde também ao SVM, com 99.6% de testes assertivos, e para o método de extração de características HOG combinado com HONV o melhor método de classificação foi também o SVM, com 99.9333% de testes assertivos. Concluimos, assim, que a melhor combinação de métodos para reconhecimento facial com dados tridimensionais é a combinação de HOG combinado com HONV com o classificador SVM, com 99.9333% de casos assertivos.

## Conclusão e trabalho futuro

A pesquisa apresentada nesta tese teve como objetivo principal estudar qual o método mais exequível para o problema do reconhecimento facial tridimensional, possibilitando a sua contribuição a algumas pesquisas já realizadas sobre o assunto em análise. Como se pode verificar através dos resultados presentes no capítulo 4, o melhor método para o reconhecimento facial tridimensional passa pela extração das características da imagem usando o HOG em combinação com o HONV e a classificação a ser determinada pelo SVM, sendo alcançadas taxas de reconhecimento satisfatórias. Relativamente ao reconhecimento facial bidimensional, o melhor método passa pela extração das características da imagem usando HOG e a classificação a ser determinada pelo SVM. No entanto, como se pode verificar, a introdução dos dados de profundidade nas imagens, leva a que o reconhecimento facial bidimensional passe a tridimensional, melhorando os principais resultados, levando à concepção de que o reconhecimento facial tridimensional se torna mais robusto e eficiente relativamente ao reconhecimento facial bidimensional. Um dos factos a considerar diz respeito aos modelos de treino, ou seja, às imagens de treino. Essas imagens constituem um fator importante para alcançar um bom reconhecimento e, portanto, devem ser cuidadosamente escolhidas e, ainda, ser treinadas com um bom número de imagens de treino para que os algoritmos sejam eficazes.

O reconhecimento facial poderá ser dificultado através de diversos fatores, são eles; a iluminação dos ambientes; a intensidade e o ângulo da iluminação, que altera muito as propriedades das imagens e que constitui alguns problemas para o reconhecimento. No entanto, para este estudo as imagens da base de dados através das quais foram realizados todos os testes foram todas recolhidas no mesmo ambiente, ou seja, com iluminação igual. No entanto, o desempenho dos métodos implementados pode não ser tão favorável se forem usadas imagens em ambientes diferentes. Ainda assim, esse fator pode ser aprimorado no futuro, para isso deverá ser implementado um sistema de normalização mais avançado, de modo a melhorar os problemas de iluminação. Assim sendo, este poderá ser um dos próximos passos a seguir, de forma a alcançar bons resultados, com imagens recolhidas em outros ambientes. Também como trabalho futuro, devem ser estudadas redes neuronais que possam contribuir no reconhecimento facial de forma a melhorarem os resultados obtidos nos métodos apresentados.

Adicionalmente, outro assunto que poderá ser alvo de pesquisa é o uso de outros métodos de recolha de dados, substituindo o kinnect. Neste trabalho esse assunto não foi aprofundado, não sendo esse o objetivo principal.

Em suma, atendendo aos dados recolhidos e trabalhados, podemos verificar que o melhor método de reconhecimento facial bidimensional é a combinação da extração de características RGB com o classificador LDA, mas também, e

apresentando os mesmos resultados, a combinação da extração de características HOG com o classificador SVM. Não obstante, aquele que pretendemos colocar em evidência, sendo que o melhor método de reconhecimento facial tridimensional é a combinação de extração de características HOG com HONV e o classificador SVM, por apresentar os melhores resultados.

## REFERÊNCIAS BIBLIOGRÁFICAS

Face Recognition Wei-Lun Chao GICE, National Taiwan University. 2010.

A.L. Yuille, D.S. Cohen e P. Hallinam. "Feature extraction from faces using deformable templates". International Journal of Computer Vision. 1992.

J. Waite e W. Welsh. "Head boundary location using snakes". British Telecom Technology Journal. 1990.

M.Turk e A. Pentland. "Eigenfaces for recognition". Journal of Cognitive Neuroscience. 1991.

H. Wang e S. Chang. "A highly efficient system for automatic face region detection in MPEG video". IEEE Transactions on Circuits and Systems for Video Technology. 1997.

R. Brunelli e T. Poggio. "Face recognition: features versus templates". IEEE Transactions on Pattern Analysis and Machine Intelligence. 1993.

Rolf P. Würtz. "Object recognition robust under translations, deformations and changes in the background". IEEE Transactions on Pattern Analysis and Machine Intelligence. 1997.

B. Moghaddam e A. Pentland. "Probabilistic visual learning for object representation". IEEE Transactions on Pattern Analysis and Machine Intelligence. 1997.

Joseph J. Atick, Paul A. Griffin e A. Norman Redlich. "Statistical approach to shape and shading: reconstruction of 3D face surfaces from single 2D images". Neural Computation. 1996.

Pagliari, D., Pinto, L, 2105. Calibration of Kinect for Xbox One and Comparison between the Two Generations of Microsoft Sensors. Sensors, 15, 27569-27589.

Castaneda, V. & Navab, N., 2014. Time-of-Flight and Kinect Imaging. , pp.1–27.

Fankhauser, P. et al., 2015. Kinect v2 for mobile robot navigation: Evaluation and modeling. Proceedings of the 17th International Conference on Advanced Robotics, ICAR 2015, pp.388–394.

Xiaohu Shao, Xi Zhou e Cheng Cheng. "3D Face Reconstruction and Dynamic Feature Extraction for Pose-Invariant Face Recognition", 2nd International Symposium on Computer, Communication, Control and Automation. 2013.

Seema Mehra e Saurabh Charaya. “Enhancement of Face Recognition technology in Biometrics”. *International Journal of Scientific Research*. 2016.

TRIGGS, B.; DALAL, N. Histograms of Oriented Gradients for Human Detection. *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Wang, S., LI, W., Wang, Y., Jiang, Y., Jiang, S., and Zhao, R. (2012). An improved difference of gaussian filter in face recognition. *Journal of Multimedia*, 7:429–433.

Iacopo Masi, Anh Tuãn Trãn, Jatuporn Toy Leksut, Tal Hassner e G´erard Medioni. “Do We Really Need to Collect Millions of Faces for Effective Face Recognition?”. 2016.

C. W. Hsu, C. C. Chang and C. J. Lin, “A Practical Guide to Support Vector ... *JOURNAL NAME: Intelligent Information Management*, Vol.2 No.6, June 29, 2010.

Richard Szeliski. “Computer Vision: Algorithms and Applications”. 2011.

Gonzalez e Woods. “Digital Image Processing (3rd Edition)”. 2006.

Fu-Pen Chiang. “Moiré methods of strain analysis.” *Experimental Mechanics*. 1979.

DÉNIZ, O.; BUENO, B.; SALIDO, J.; TORRE, F. de La. 2011. Face recognition using Histograms of Oriented Gradients. *Pattern Recognition*, setembro de 2011.

Tang S, Wang X, Lv X, Han T, Keller J, He Z, et al. Histogram of Oriented Normal Vectors for Object Recognition with a Depth Sensor. In: Lee K, Matsushita Y, Rehg J, Hu Z, editors. *Computer Vision – ACCV 2012*, vol. 7725, Springer Berlin Heidelberg. 2013: 525–38.

Sirovich, L. and Kirby, M. (1987) Low-Dimensional Procedure for the Characterization of Human Faces. *Journal of the Optical Society of America A*, 4, 519-524.

Fisher, R.A. (1936) The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7, 179-188.

Guo T, et al. (2007) A signal from inside the peroxisome initiates its division by promoting the remodeling of the peroxisomal membrane. *J Cell Biol*177(2):289-303

Hastie, T., Tibshirani, R., Friedman, J. (2009). *The elements of statistical learning: data mining, inference and prediction*. Springer.

Fukunaga, K.; Narendra, P. M. A branch and bound algorithm for computing k-nearest neighbors. *IEEE Transactions on Computers*, v. 100, n. 7, p. 750–753, 1975.

Boser E., Guyon I.M. and Vapnik V.N., 1992, A Training Algorithm for Optimal Margin Classifiers, *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, 5, pp 144-152.

Burges, C. J. C., A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.

Cristianini, N., and Shawe-Taylor, J, *An Introduction to Support Vector Machines*. Cambridge University Press, Cambridge, UK, 2000.

MUELLER, K.-R., MIKA, S., R " ATSCH, G., TSUDA, K. and SCH " OLKOPF, B. " (2001): An Introduction to Kernel-Based Learning Algorithms. *IEEE Transactions on Neural Networks*, 12(2), 181–201.

COVER, Thomas M., 1965. Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition. *IEEE Transactions on Electronic Computers*, 14(3), 326–334.

Schölkopf, B., Burges, C., & Vapnik, V. (1995). Extracting support data for a given task. In: U.M. Fayyad, R. Uthurusamy (Eds.), *Proceedings, First International Conference on Knowledge Discovery and Data Mining*, Menlo Park. AAAI Press.

Jorge Alves da Silva. "Aquisição e Processamento de informação tridimensional" *Dissertação de Doutorado apresentada à Fac. Eng.<sup>a</sup> da Univ. do Porto Departamento de Eng.<sup>a</sup> Electrotécnica e de Computadores*. 1994.