

Miguel Coelho de Pinho

Detecção e seguimento automáticos de faces humanas em vídeos capturados em
ambientes não controlados

Miguel Coelho de Pinho

Detecção e seguimento automáticos de faces humanas em vídeos capturados em
ambientes não controlados

Detecção e seguimento automáticos de faces humanas em vídeos capturados em
ambientes não controlados

Por:

Miguel Coelho de Pinho

Co-Orientadores

Professor Doutor Nuno Magalhães Ribeiro

Professor Doutor Feliz Ribeiro Gouveia

Dissertação apresentada à Universidade
Fernando Pessoa como parte de requisitos
para a obtenção do grau de Mestre em
Engenharia Informática – Sistemas de
Informação e Multimédia

Resumo

O trabalho apresentado nesta dissertação desenvolve um caso de estudo sobre detecção e seguimento automático de faces humanas concretizado em duas aplicações distintas: uma aplicação que actua em tempo real sobre vídeo proveniente de câmaras de vigilância e outra aplicação que actua sobre ficheiros de vídeo previamente recolhidos e armazenados em disco. Este trabalho incide em situações e locais que possuem câmaras de videovigilância designados por ambientes não controlados. Esta dissertação inicia-se com uma revisão das técnicas utilizadas para a detecção facial.

Em seguida faz-se a definição dos requisitos das aplicações e a descrição do problema. A seguir faz-se a descrição detalhada do desenvolvimento das duas aplicações, justifica-se a escolha do sistema de reconhecimento humano e a estrutura de armazenamento dos dados adquiridos durante as detecções.

Detalha-se ainda a implementação das aplicações e o respectivo funcionamento, incluindo as detecções, o armazenamento e a determinação das orientações das faces humanas detectadas, bem como as várias funcionalidades que integram as aplicações.

Faz-se igualmente a avaliação das aplicações, verificando os seus desempenhos ao longo do processamento e a precisão dos respectivos resultados de reconhecimento: a verificação da acuidade dos resultados relativos aos trajectos de uma pessoa e das respectivas orientações ajuda a obter informações reais e concretas que podem posteriormente ser utilizadas de forma a trazer benefícios para os utilizadores das aplicações.

A conclusão identifica claramente as utilidades de ambas as aplicações, destacando os pontos fortes e as respectivas limitações. Aponta-se ainda um conjunto de melhorias que, no futuro, podem ser introduzidas de modo a aumentar o desempenho das aplicações.

Abstract

The work presented in this dissertation develops a case study about an automatic human face detection and tracking of two distinct applications: an application that acts through video surveillance cameras in real time and an application for the videos saved in previous collections of reality. This case study explores the various locals that have video surveillance cameras designed for uncontrolled environments. It begins by presenting the techniques used for facial detection and a few definitions and a description of the development applications used on the creation of these case study applications.

Forward, it is then made a definition of the applications requirements and a description of the problems to be resolved. Then it describes the most important steps in the development of the two applications, the choice of human recognition system and the structure to store the data acquired in the detections.

Regarding the applications implementation, they present various phases and is described theirs functionalities in detail. This description includes detections, storage and the determination of the face orientations and also the various functions in the applications.

A significant aspect of this work is also an opportunity to evaluate the applications, analyzing their performance throughout the process and the accuracy of the results. The verification the authenticity of the results about the movements of a person and orientations helps to give real information that can be used in ways that brings benefits to the users of these applications.

Finally is made the conclusion about the utility of both applications, which identifies its strengths and limitations. At this phase, is described the possible improvements that can be done to improve the application in the future.

À minha Família

Agradecimentos

Agradeço aos meus Pais e Irmão que sempre me incentivaram e apoiaram nas decisões que fui tomando ao longo da minha vida.

Aos meus Avós, Padrinho, Tia e Primos.

A ambos os professores Doutores Nuno Magalhães Ribeiro e Feliz Ribeiro Gouveia por todo o tempo dispensado e pela ajuda durante a sua orientação deste trabalho.

A todos os professores do curso pela ajuda, apoio e passagem de conhecimentos ao longo do curso, nas disciplinas por eles leccionadas e não só.

A todos os amigos pelo companheirismo e ajuda durante a frequência do curso e pela participação directamente ou indirectamente neste trabalho.

Índice

Resumo	I
Abstract.....	III
Agradecimentos	VII
Índice	IX
Índice de figuras	XIII
Índice de tabelas	XV
1. Introdução.....	1
1.1. Descrição do problema	2
1.2. Objectivos do trabalho	2
1.3. Motivação	3
1.4. Metodologia.....	3
1.5. Estrutura do documento	4
2. Detecção de Faces Humanas	6
2.1. Visão Computacional.....	6
2.2. Métodos e Técnicas de Detecção Humana	7
3. Aplicações para a Detecção de Faces Humanas.....	12
3.1. Microsoft Visual Studio 2008.....	13
3.2. Biblioteca OpenCV	13
3.2.1. Estrutura da biblioteca OpenCV	14
4. Análise e concepção dos Sistemas “ <i>FaceDetect</i> ”	16
4.1. Descrição do Problema	16
4.2. Metodologia de Engenharia de Software utilizada	17
4.3. Análise do Sistema.....	17
4.3.1. Requisitos Funcionais das Aplicações.....	17

4.3.2.	Requisitos Não-Funcionais.....	18
4.4.	Abordagem ao Problema	19
4.4.1.	Sistema de Reconhecimento Humano	19
4.4.2.	Sistema de Armazenamento de Dados	20
5.	Implementação do Sistema.....	23
5.1.	Aplicação FaceDetect	23
5.1.1.	Inicialização e Fases das Aplicações	25
5.1.2.	Análise de <i>Frames</i>	27
5.1.2.1.	Detecção Facial/ Nasal/Ocular	29
5.1.2.2.	Armazenamento de Dados/Informações.....	33
5.1.2.3.	Determinação das Orientações	37
5.1.3.	Visualização de <i>Frames</i> /Dados	38
5.1.4.	Eliminação de Faces	41
5.1.5.	Incrementação.....	42
5.1.6.	Menu de Resultados	42
5.1.6.1.	Trajectos de uma Face	42
5.1.6.2.	Número de Orientações Reconhecidas	44
5.2.	Diagrama de Fluxo.....	44
6.	Teste e Avaliação	46
6.1.	Técnicas de avaliação	46
6.2.	Avaliação do Sistema.....	47
6.3.	Análise e discussão dos Resultados	48
6.3.1.	Classificadores <i>Haar</i>	48
6.3.2.	Armazenamento das Detecções	50
6.3.3.	Orientações Reconhecidas Correctamente	51
6.3.4.	Tempo de Processamento	52
7.	Conclusão	54

7.1. Considerações Finais	54
7.2. Limitações.....	55
7.3. Trabalho Futuro	56
Bibliografia.....	57

Índice de figuras

Figura 2-1 – Características rectangulares numa área de detecção	7
Figura 2-2 – Matriz de quatro referências numa dada área	8
Figura 2-3 – Características Faciais.....	9
Figura 2-4 – Cascata de Classificadores.....	10
Figura 2-5 – Expansão das características quadráticas.....	10
Figura 2-6 – Exemplo de uma Subtracção de Fundo.....	11
Figura 3-1 – Estrutura básica do <i>OpenCV</i>	14
Figura 4-1 – Modelo de Prototipagem.....	17
Figura 4-2 – Estrutura das Listas Ligadas	20
Figura 4-3 – Exemplo de Dados Armazenados nas Listas	22
Figura 5-1 – Criação Novo Projecto.....	23
Figura 5-2 – Criação Ficheiro C++	24
Figura 5-3 – Ligação do Projecto às bibliotecas <i>OpenCV</i>	25
Figura 5-4 – Exemplo da Execução das Aplicações.....	26
Figura 5-5 – Tons de Cinzento/ Tons de Cinzento com equalização do histograma	28
Figura 5-6 – Funcionamento da Detecção	31
Figura 5-7 – Imagem Original com Reconhecimento Facial (esquerda) e Área da pesquisa Nasal e Ocular (direita).....	32
Figura 5-8 – Reconhecimento Facial/Ocular/Nasal.....	33
Figura 5-9 – Ponto da Detecção Facial Analisado	33
Figura 5-10 – Inserção de Valores nas Listas.....	34
Figura 5-11 – Centro da Janela de Pesquisa	36
Figura 5-12 – Limitação das Áreas Oculares	37
Figura 5-13 – Orientação à Esquerda/Orientação à Direita.....	37
Figura 5-14 – Visualização de uma <i>Frame/Dados</i> (1)	39
Figura 5-15 – Visualização de uma <i>Frame/Dados</i> (2)	40
Figura 5-16 – Visualização de uma <i>Frame/Dados</i> (3)	40
Figura 5-17 – Exemplo de uma Falsa Detecção/Falso Positivo	41
Figura 5-18 – Exemplo da Informação/Trajeto de uma Face	43
Figura 5-19 – Exemplo do Número de Orientações Reconhecidas.....	44

Figura 5-20 – Diagrama de Fluxo	45
Figura 6-1 – Exemplo de Uma Orientação Incorrecta.....	52

Índice de tabelas

Tabela 4-1 – Descrição do Problema.....	16
Tabela 4-2 – Requisitos Não-Funcionais	18
Tabela 4-3 – Lista de Variáveis/Informações dos Nós.....	21
Tabela 5-1 – Fases das Aplicações	26
Tabela 5-2 – Ficheiros dos Classificadores <i>Haar</i> Utilizados	27
Tabela 5-3 – Parâmetros da Função <i>cvHaarDetectObjects</i>	30
Tabela 5-4 – Armazenamento de Dados Oculares	36
Tabela 6-1 – Características Principais do Vídeo de Testes.....	48
Tabela 6-2 – Desempenho do <i>haarcascade_frontalface_alt2</i>	49
Tabela 6-3 – Desempenhos Oculares e Nasais	50
Tabela 6-4 – Resultados do Armazenamento dos Dados	50
Tabela 6-5 – Resultados das Orientações	51
Tabela 6-6 – Resultados do Vídeo de Testes com Diferentes Taxas de <i>Frames</i> Por Segundo	52
Tabela 6-7 – Resultados do Vídeo de Testes com Diferentes Resoluções	53

1. Introdução

A detecção e o seguimento de faces humanas em sequências de vídeo é uma área muito importante e fundamental na visão computacional. Esta é uma das áreas especialmente activa devido às suas inúmeras aplicações e à sua complexidade inerente.

Com o mundo em contínua evolução, o uso das tecnologias torna-se cada vez mais uma realidade fundamental nas tarefas diárias. Mais especificamente, não existem ainda soluções suficientemente desenvolvidas que permitem que as empresas processem os registos vídeo obtidos através de câmaras de videovigilância de forma a auxiliar a tomada de decisão sobre a actividade humana nas áreas e locais abrangidos pela câmara.

Neste contexto, o desenvolvimento deste trabalho pretendeu conceber uma possível solução para este problema, consistindo em duas aplicações que adquirem automaticamente informações sobre o ambiente, e demonstrando que o processamento desta informação permite obter dados úteis aos respectivos utilizadores.

O problema concreto abordado neste caso de estudo partiu do facto de não existirem actualmente meios automáticos e de baixo custo para se obter informação sobre as orientações do olhar das pessoas e sobre as respectivas deslocações num determinado espaço físico, o que obriga os profissionais a observar as gravações, ou as imagens captadas directamente pela câmara, para a obtenção da informação pretendida. Esta forma de proceder conduz a um grande desperdício de tempo e obtenção de informação pouco fidedignas devido ao cansaço que produz nas pessoas.

Esta dissertação propõe uma solução que consiste em analisar a informação recolhida automaticamente por duas aplicações de *software* que identificam as várias orientações e deslocações das pessoas num determinado espaço físico. Uma das aplicações actua directamente sobre as imagens que estão a ser recolhidas em tempo real por uma câmara de videovigilância já instalada na organização. A outra aplicação actua somente nos ficheiros de vídeos que já foram recolhidos e armazenados em recolhas anterior. Nesta dissertação, propõe-se pois uma solução que assenta no desenvolvimento de aplicações que permitem detectar faces humanas e também alguns dos elementos que as

caracterizam (olhos e nariz) no sentido de reconhecer a orientação da cabeça e determinar “para onde a pessoa está olhar” num determinado instante.

1.1. Descrição do problema

Em ambientes físicos não controlados existem diversas situações nas quais é útil efectuar o processamento e a análise de imagens (*frames*¹ de vídeo) para identificar com uma certa precisão a orientação do campo de visão das pessoas e a respectiva deslocação dentro desses espaços físicos. As imagens podem ser obtidas a partir de câmaras de videovigilância normais já instaladas, não sendo necessário recorrer a equipamento mais sofisticado, ou sequer a alterações de equipamentos já existentes, limitando assim os custos de tal solução.

Entende-se como ambientes físicos não controlados todos os locais onde, por motivos de segurança, estão instaladas câmaras de videovigilância e nos quais as pessoas são informadas que estão a ser filmadas, mas não se sentem observadas, criando assim um ambiente de normalidade para a recolha de dados. Alguns exemplos destes ambientes incluem, entre outros, as grandes superfícies comerciais tais como os hipermercados e lojas temáticas, áreas de exposições, estações de comboios ou de transportes terrestres e aeroportos.

No âmbito do caso de estudo descrito nesta dissertação, a criação de uma aplicação que determina a direcção do campo de visão das pessoas em *frames* de vídeo, aqui designada por “*FaceDetect*”, permite determinar os objectos ou motivos que despertam a atenção das pessoas, e que são, portanto, alvo da respectiva visualização, em determinados espaços físicos, proporcionando assim informações úteis aos profissionais e empresas que operam nesse ambiente.

1.2. Objectivos do trabalho

O objectivo principal deste trabalho foi o de desenvolver uma solução tecnológica que permitisse identificar percursos e orientações das pessoas numa determinada área ou local físico, com o intuito de analisar e avaliar os tipos de reconhecimento facial existentes actualmente, bem como identificar os problemas que podem surgir ao longo da sua realização. O trabalho descrito nesta dissertação incidiu sobre a concepção e implementação de uma aplicação de identificação facial, bem como alguns elementos

¹ *Frames* são um conjunto de imagens que pertencem à sequência de um vídeo.

representativos da face e o respectivo seguimento em tempo real. Uma outra aplicação, a ser desenvolvida parte dos mesmos princípios da aplicação anterior, mas destina-se a processar vídeos já existentes.

Pretende-se ainda avaliar o nível de desempenho das aplicações medindo o tempo que demoram a efectuar a análise dos vídeos e avaliar a eficácia da informação armazenada e a precisão das orientações reconhecidas automaticamente pelas aplicações.

1.3. Motivação

Uma das razões que motivou a realização deste trabalho foi a de tentar reconhecer faces humanas e determinar a sua direcção com uma simples câmara de vídeo para que possa ser utilizada em ambientes não controlados sem introduzir grandes custos.

O segundo motivo foi a constatação do facto desta área tecnológica estar ainda relativamente pouco desenvolvida, nomeadamente a área dos sistemas que permitem reconhecer pessoas em *frames* de vídeo e respectivas características de processamento e análise de imagens, essenciais para o desenvolvimento de soluções nesta área. Embora já existam várias aplicações nesta área, ainda existem muitos aspectos que podem ser melhorados, tais como, seguimento de pessoas e a identificação das suas orientações, permitindo assim abordar áreas de aplicação menos exploradas.

De facto, a introdução de novos factores nas aplicações de reconhecimento facial ajuda a que estes sistemas não sirvam só para detectar faces mas também para obterem informações sobre os focos de interesse das pessoas num determinado local.

1.4. Metodologia

A metodologia usada para a elaboração deste trabalho foi a seguinte:

1. Identificar as características do problema e efectuar o levantamento de requisitos e a arquitectura da aplicação no sentido de:
 - a. Identificar a face de uma pessoa numa imagem;
 - b. Reconhecer os olhos na área abrangida pela identificação da face;
 - c. Localizar o nariz na região identificada como face humana;

- d. Efectuar o seguimento de uma pessoa através dos dados obtidos no reconhecimento da face.
2. Estudar os diferentes algoritmos que foram desenvolvidos para o mesmo efeito e com base na análise dos requisitos recolhidos desenvolver a concepção das aplicações.
 3. Analisar as características das plataformas de desenvolvimento de software a utilizar para a implementação da aplicação e estudar as diferentes formas de o efectuar.
 4. Escolha do modelo de Engenharia de Software que melhor se adapta para o desenvolvimento do sistema.
 - a. Modelo de Prototipagem que envolve o desenvolvimento rápido de uma versão piloto executável de partes do sistema onde são feitas várias modificações subsequentes até se atingir o produto final desejado.
 5. Desenvolver a aplicação através da implementação de algoritmos, estruturas de dados com base na biblioteca de visão computacional *OpenCv* (Bradski & Kaehler 2008) na linguagem de programação C.
 6. Identificar os métodos de avaliação usados de forma a efectuar testes à precisão da identificação e determinação da orientação. Escolher o método mais apropriado para a realização da avaliação da aplicação com base em casos já existentes.
 7. Analisar os resultados e obter conclusões acerca das vantagens da aplicação desenvolvida, ou seja, verificar se a sua utilização traz benefícios aos utilizadores na análise dos ambientes não controlados.

1.5. Estrutura do documento

Esta dissertação encontra-se organizada em sete capítulos. No primeiro capítulo faz-se uma introdução ao tema desta dissertação, incluindo a descrição do problema abordado, os respectivos objectivos, a motivação para o trabalho de desenvolvimento das aplicações aqui descritas e a respectiva metodologia nela associada.

No capítulo 2 inicia-se com uma pequena introdução à visão computacional, seguida pela análise das técnicas de detecção facial mais importantes que existem actualmente. Descreve-se com maior detalhe as abordagens sugeridas por vários autores no que diz respeito à resolução do problema de detecção de faces humanas, ou da própria identificação de um ser humano.

No capítulo 3 é feita uma introdução às tecnologias utilizadas no desenvolvimento das aplicações. Apresentam-se as plataformas de desenvolvimento, as bibliotecas utilizadas e é feita uma pequena introdução às características das tecnologias utilizadas durante a realização deste trabalho.

O capítulo 4 descreve a análise e concepção da aplicação “FaceDetect”. Apresenta-se a descrição do problema, identificam-se a metodologia de engenharia de software utilizada, definem-se os requisitos das aplicações e é apresentada uma abordagem ao problema. Esta abordagem identifica as técnicas específicas que foram utilizadas para efectuar a detecção de facial, e a forma como os dados recolhidos durante a detecção foram armazenados.

No capítulo 5 detalha-se o conjunto de fases do processamento e análise de imagem realizado pelas aplicações através de uma descrição dos passos realizados durante a fase de análise das *frames*, da eliminação de faces, visualização e incrementação, e identificam-se os resultados que são adquiridos ao longo de todo o processamento realizado pelas aplicações propostas. É ainda apresentada o diagrama de fluxo das aplicações “FaceDetectRT” e “FaceDetectVC”.

O capítulo 6 descreve a avaliação que foi realizada sobre as aplicações desenvolvidas. Identificam-se as técnicas de avaliação utilizadas e efectua-se uma avaliação do sistema desenvolvido ao nível desempenho dos classificadores, armazenamento dos dados, reconhecimento de orientações das pessoas, tempo de processamento e, finalmente, apresenta-se uma análise crítica dos resultados obtidos.

No último capítulo são apresentadas as conclusões do estudo descrito nesta dissertação, as suas limitações e o trabalho futuro que pode ser desenvolvido de forma a melhorar o desempenho de ambas as aplicações desenvolvidas.

2. Detecção de Faces Humanas

Segundo Benezeth et al. (2009), o interesse da criação de aplicações que detectam humanos tem vindo a aumentar nos últimos anos devido ao aumento da robustez, entre outros sistemas, na área da videovigilância, na robótica, e na indexação baseada em conteúdos multimédia.

De facto, Kollreider et al. (2007) afirmam que a análise facial biométrica pode ser baseada em duas aproximações:

- Detecção facial;
- Reconhecimento facial.

A detecção facial permite localizar uma ou mais instâncias correspondentes à face humana contidas num vídeo ou numa imagem. Por outro lado, o reconhecimento facial estabelece uma relação única entre dois ou mais traços independentes que correspondem ao rosto dessa mesma pessoa (Kollreider et al. 2007).

De acordo com os autores destes trabalhos, os desafios, tanto em relação à detecção humana como à detecção facial, dizem respeito ao ambiente onde essas imagens são retiradas, pois, muitas das vezes a complexidade do ambiente dificulta a acção da detecção devido a aspectos tais como, entre outros, a iluminação, plano de fundo, cor da pele, e expressões faciais, (Benezeth et al. 2009) e (Kollreider et al. 2007).

2.1. Visão Computacional

A visão computacional realiza a transformação dos dados recolhidos por uma máquina fotográfica ou de vídeo para que os dados possam ser interpretados pelo computador. Essa nova representação pode ser ou não a reprodução da realidade, ou seja, uma imagem normal (real) pode ser representada em tons de cinzento (Bradski & Kaehler 2008).

A visão computacional utiliza métodos estatísticos que separam os dados com base em modelos construídos e é assistida por áreas da geometria, física e a teoria da aprendizagem (Forsyth & Ponce 2002).

2.2. Métodos e Técnicas de Detecção Humana

A técnica proposta por Viola & Jones (2001) dá a possibilidade de efectuar a detecção de qualquer tipo de objecto (neste caso faces humanas) de forma genérica e em tempo real com uma taxa de sucesso bastante elevada. A abordagem proposta baseia-se em algoritmos de aprendizagem automática (*machine learning*²), aplicados de forma a identificar objectos, e capazes de processar imagens com muita rapidez.

Segundo estes autores, a utilização das características de um objecto, ao invés da utilização de pixéis³ contidos numa imagem, permite aumentar a velocidade da análise porque o número de características de um objecto é substancialmente inferior ao número de pixéis de um objecto (Viola & Jones 2001).

Na concepção de uma ferramenta com as características descritas anteriormente, os autores principiaram por basear-se na técnica proposta por Papageorgiou et al. (1998) na qual a representação do objecto é feita de forma rectangular dentro de uma área de detecção baseada em cálculos semelhantes aos que são realizados para uma transformada de Haar (a partir daqui designados por *Haar-like*⁴).

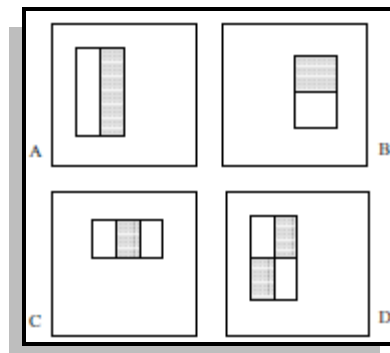


Figura 2-1 – Características rectangulares numa área de detecção (fonte: Viola & Jones 2001)

Na figura 2-1 ilustram-se três tipos distintos de características:

- 1) A característica de dois rectângulos (A e B) cujo valor é a diferença entre a soma dos pixéis numa área com dois objectos rectangulares do mesmo tamanho;

² Machine Learning são algoritmos que permitem ao computador aprender através de dados fornecidos por base de dados ou outros recursos. Isto permite que o computador possa tomar decisões conforme os dados recebidos.

³ Pixel (Picture Element) componente de brilho e cor de formato matricial rectangular que constitui uma imagem.

⁴ Haar-like provém do nome Alfréd Haar que propôs o primeiro método baseado em funções quadráticas.

- 2) A característica de três rectângulos (C) calcula a soma dos dois rectângulos exteriores e subtrai o do centro;
- 3) A característica de quatro rectângulos (D) faz a diferença entre os pares das diagonais dos rectângulos.

De forma a aumentar a rapidez da computação dos dados das características rectangulares referidas acima, Viola & Jones (2001) usaram uma representação intermédia que designaram por imagem integral (“*Integral Image*”). Esta define que numa dada localização, definida por um par de coordenadas x e y , é efectuado o somatório de todos os pixéis acima e à esquerda do ponto definido. Após obter os cálculos de todos os pixéis nessa área é possível obter os cálculos de qualquer área rectangular através de uma representação matricial de quatro referências, tal como se ilustra na figura 2-2.

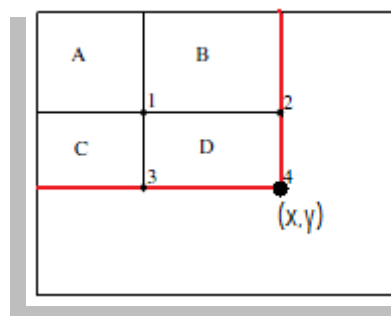


Figura 2-2 – Matriz de quatro referências numa dada área (adaptado de: Viola & Jones, 2001)

Na Figura 2-2 é possível determinar a região D pois, sabendo que o valor na posição 1 é a soma dos pixéis do rectângulo A, o valor da posição 2 será $A+B$, a posição 3 será $A+C$ e a posição 4 será $A+B+C$. Tendo as quatro referências a soma de D é definida pelos valores das posições $4+1 - (2+3)$.

O passo seguinte sugerido por Viola & Jones (2001) consiste em treinar os classificadores⁵ através de um conjunto de imagens no qual existem as designadas imagens positivas e negativas: as imagens positivas contêm o objecto que se deseja encontrar e as imagens negativas evidenciam a ausência desse objecto. Os autores

⁵ Classificador contém dados armazenados com base nas amostras ou exemplos fornecidos pelo utilizador de forma coerente.

utilizaram uma variante do algoritmo “*AdaBoost*”⁶ criado por Freund & Robert (1997) para seleccionar um pequeno conjunto de características e para treinar o classificador.

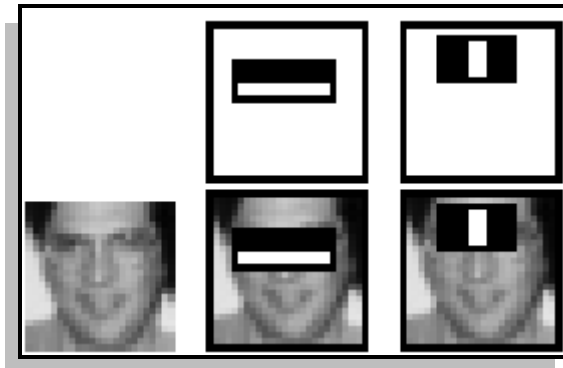


Figura 2-3 – Características Faciais (fonte: Viola & Jones 2001)

Na figura 2-3 mostra-se a forma como o classificador percorre a imagem de modo a encontrar áreas com as mesmas características do objecto desejado (neste caso a face humana a detectar). Os dois quadrados acima são características seleccionadas pelo algoritmo “*AdaBoost*” sendo sobrepostas na imagem de modo a permitir a possibilidade de verificar o seguinte:

- No primeiro caso (à esquerda na figura 2-3), a região dos olhos e as bochechas da pessoa possuem diferentes intensidades, sendo a região olhos mais escura do que a região das bochechas;
- O segundo caso (à direita na figura 2-3) permite comparar a intensidade da região dos olhos com a intensidade dos pixéis da região do nariz.

De forma a descartar várias áreas da imagem onde não existe o objecto desejado, os autores utilizaram classificadores em cascata possibilitando assim um maior foco nas áreas mais promissoras da imagem (Viola & Jones 2001).

A cascata de classificadores representada na figura 2-4 possui n etapas, sendo que em cada etapa se determina se a área de pesquisa pode ou não corresponder a uma face humana. Caso o objecto seja aceite como possível face, passa então para a etapa seguinte (F2); caso contrário será rejeitado. Em cada etapa, os classificadores são cada vez mais precisos. Caso a área em análise na última etapa seja classificada como face,

⁶ AdaBoost é um algoritmo de Machine Learning que permite impulsionar o desempenho de classificadores.

então assegura-se, com uma probabilidade entre 85 e 95%, que nessa área existe o rosto de uma pessoa.

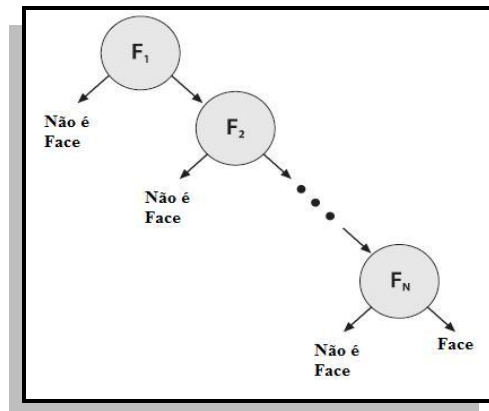


Figura 2-4 – Cascata de Classificadores (adaptado de: Bradski & Kaehler 2008)

No sentido de melhorar este método, Lienhart & Maydt (2002) decidiram expandir as representações quadráticas *Haar* através da introdução de novas representações (figura 2-5).

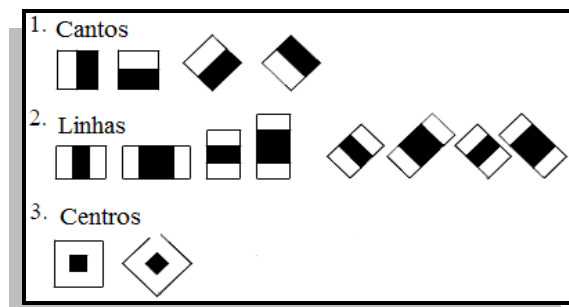


Figura 2-5 – Expansão das características quadráticas (adaptado de: Lienhart & Maydt, 2002)

Desta forma, conseguiram reduzir a taxa de falsos positivos em cerca de 10% em relação ao método anterior, possibilitando pois um aumento na eficácia da detecção de faces humanas.

Já a aplicação proposta por Benezeth et al. (2009) pretendeu obter um sistema com um elevado grau de desempenho para ser usado em ambientes interiores não controlados. O algoritmo que propuseram para a realização da detecção do corpo de um ser humano em vídeos, e em tempo real, baseia-se em 3 fases distintas.

Inicialmente recorreram à técnica de subtração do plano de fundo (*background subtraction*) que permite limitar o espaço de pesquisa dos classificadores *Haar-like*.

Esta técnica permite retirar as áreas de interesse que se encontram na imagem. A escolha destas áreas é feita através do cálculo da diferença entre a imagem actual em relação à imagem obtida anteriormente. Essencialmente, esta técnica permite encontrar as movimentações dos objectos presentes numa imagem.



Figura 2-6 – Exemplo de uma Subtracção de Fundo (fonte: Benezeth et al. 2009)

O passo seguinte proposto por Benezeth et al. (2009) corresponde à classificação de cada objecto retirado através da subtracção do fundo (designado por “blob”). Cada um destes objectos será posteriormente seguido individualmente, já que pode corresponder a uma pessoa. Finalmente utilizam-se os classificadores *Haar-like* para determinar de forma rápida se na verdade se trata de uma pessoa e, portanto, aceitar ou rejeitar o objecto.

Os resultados obtidos por Benezeth et al. (2009) foram de aproximadamente 97% em ambientes interiores não controlados para os quais os autores afirmam que o método de subtracção permite reduzir o número de falsas detecções através da redução do espaço de pesquisa onde os classificadores vão actuar.

3. Aplicações para a Detecção de Faces Humanas

As aplicações que suportam a detecção de faces humanas têm vindo a aumentar cada vez mais ao longo dos tempos, devido ao facto de não só ser uma área em crescente evolução, mas também, devido às várias utilidades que estas podem proporcionar para o dia-a-dia, tais como, segurança, ajuda, entre outros. No sentido de apresentar uma possível solução para o problema descrito na introdução desta dissertação, pretendeu-se tirar partido das tecnologias existentes que proporcionam melhores resultados. Desta forma, os resultados obtidos permitem afirmar que a adição de novas características a tais tecnologias contribuem positivamente para a respectiva evolução em aspectos menos explorados até ao momento. Para tal, investiu-se num sistema de informação que disponibiliza métodos de desenvolvimento que facilitam a implementação deste tipo de sistemas e contribuem para uma boa obtenção de resultados.

Assim, neste trabalho foram desenvolvidas duas aplicações:

- A aplicação *FaceDetectRT* que permite detectar faces humanas e alguns elementos de cada face (olhos, nariz) de uma pessoa com uma determinada precisão em vídeos que estão a ser adquiridos em tempo real;
- A aplicação *FaceDetectVC* que permite efectuar a detecção de rostos humanos e elementos em ficheiros de vídeo já existentes e para os quais há necessidade de efectuar a respectiva detecção e seguimento, proporcionando assim maior liberdade de utilização em diferentes situações e contextos.

As tecnologias utilizadas no âmbito deste trabalho incluem as seguintes:

- Plataforma de desenvolvimento *Microsoft Visual Studio 2008 Professional*;
- Biblioteca de programação *OpenCV v2.0* para processamento de imagens e vídeo;
- Linguagem de Programação *C*.

Este capítulo apresenta uma introdução à plataforma *Visual Studio* e à biblioteca *OpenCV*, onde são descritas as características de cada uma, no qual, demonstra as razões da sua escolha para o desenvolvimento das aplicações que fazem parte do trabalho descrito nesta dissertação.

3.1. Microsoft Visual Studio 2008

A plataforma *Visual Studio 2008* é um produto Microsoft que visa proporcionar uma plataforma de desenvolvimento e execução de sistemas e aplicações. Desta forma, oferece soluções para a análise e avaliação de código que permite identificar erros ou problemas existentes no mesmo. Oferece um ambiente com vários recursos para a construção de aplicações tirando partido das recentes tecnologias.

O tipo de desenvolvimento de software abrangido pelo *Visual Studio* é diverso, dando ao utilizador a escolha da linguagem de programação que proporciona melhores características para o desenvolvimento da aplicação. Este inclui *framework .NET* e linguagem de programação *Visual Basic*, *C*, *C++*, entre outros. Inclui ainda áreas de desenvolvimento Web tais como a plataforma do *ASP.NET*.

Visual Studio dá a possibilidade de instalar e trabalhar com bibliotecas externas à ferramenta de software, permitindo a expansão de mais as áreas de desenvolvimento de software.

3.2. Biblioteca OpenCV

A biblioteca *Open Source Computer Vision Library*⁷ foi originalmente criada pela *Intel*⁸. Esta provém de um sistema de código fonte livre e gratuito (“OpenSource”), o qual, é constituído por um conjunto de ferramentas com classes e funções, que oferecem características de desenvolvimento e implementação de algoritmos ao nível de processamento de imagem e visão computacional.

As linguagens de programação associadas são *C* e *C++* e é compatível com múltiplos sistemas operativos, tais como, *Linux*, *Windows* e *Mac OS X*.

⁷ Para obtenção de mais informação acerca da biblioteca *OpenCv* consultar a sua página principal em <http://opencv.willowgarage.com/wiki/>. A página inclui tópicos desde documentação, versão actual até a obtenção do ficheiro de instalação da biblioteca.

⁸ *Intel* é uma abreviatura de **I**ntegrated **E**lectronics e consiste numa empresa multinacional de origem americana fabricante de circuitos micro-processadores.

Os dados fornecidos pela biblioteca permitem a criação de aplicações de forma simples na área da visão computacional, onde estas podem ser ou não em tempo real. Como exemplos de aplicações temos: reconhecimento facial, gestos, objectos entre outros.

3.2.1. Estrutura da biblioteca OpenCV

De acordo com Bradski & Kaehler (2008), a biblioteca encontra-se estruturada em cinco componentes principais:

- CV – constituído por algoritmos de processamento de imagem e algoritmos de visão computacional de alto nível;
- ML – contém bibliotecas “Machine Learning” que inclui vários classificadores estatísticos⁹ e ferramentas de clustering¹⁰;
- HighGUI – não só possui funções GUI¹¹, mas também, possui rotinas e funções I/O¹² para armazenamento e processamento de vídeos e imagens;
- CXCore – abrange as estruturas de dados básicas e conteúdo;
- CvAux – contem áreas extintas e algoritmos experimentais.

A figura seguinte mostra a forma como os componentes estão organizados.

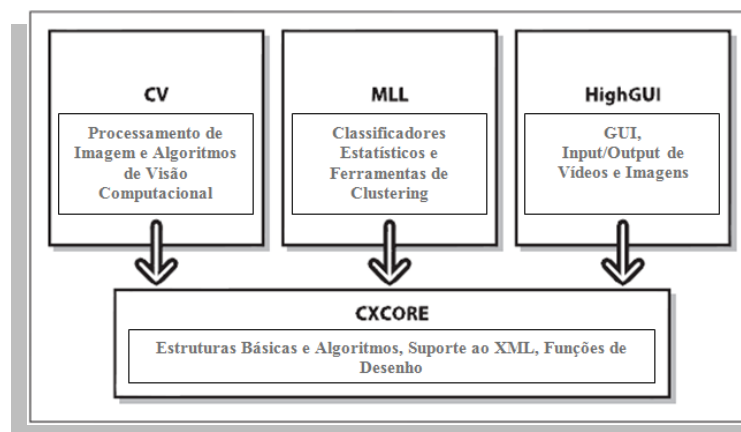


Figura 3-1 – Estrutura básica do *OpenCV* (adaptado de: Bradski & Kaehler, 2008)

⁹ Classificadores estatísticos que permitem organizar de forma coerente os dados recebidos com base nas amostras ou exemplos fornecidos pelo utilizador.

¹⁰ Ferramentas Cluster fazem o agrupamento dos dados automaticamente de acordo com o nível de semelhança.

¹¹ Funções GUI (Graphical User Interface) são funções que permitem ao utilizador interagir com as aplicações.

¹² Funções I/O (Input/Output) são funções para inserção/obtenção de dados.

Para o problema em análise nesta dissertação vamos usar os componentes CV e HighGUI. Sendo estes os componentes de processamento de imagem, de construção de interfaces e obtenção de dados, possuem assim, as características e funções necessárias para o desenvolvimento das aplicações nesta dissertação.

4. Análise e concepção dos Sistemas “FaceDetect”

Neste capítulo faz-se a descrição do problema abordado e identifica-se a metodologia associada às aplicações desenvolvidas. Para além disso pretende-se ainda identificar e definir os requisitos das aplicações e descrever a abordagem ao problema. Assim, partindo dos dados e requisitos obtidos, determina-se a abordagem a seguir para o desenvolvimento das aplicações de detecção de faces humanas em tempo real ou numa fonte de vídeo.

4.1. Descrição do Problema

As câmaras de videovigilância são dos métodos mais utilizados para ajudar a manter a segurança em estabelecimentos públicos/comerciais, o facto de poder analisar o ambiente permite a que esta análise não só possa ser usada para segurança mas também para outras funções. Na tabela 4-1 resume-se o problema.

Problema	As câmaras de videovigilância são usadas somente como método de segurança. Por vezes é necessário a compra de equipamentos de forma a permitir elaborar a análise facial e determinar a sua orientação.
Ambientes alvo	Hipermercados, aeroportos, estações de comboios, parques, entre outros.
Impacto	Apenas uma utilidade (segurança).
Uma boa solução traria benefícios como	Inexistência de custos, isto é, a aplicação faz o reconhecimento de faces humanas nas imagens gravadas ou captadas no momento através das câmaras de vídeo comuns. Simplicidade de execução, apenas é necessário executar a aplicação. Capacidade de Seguimento Humano, ou seja, a possibilidade de identificar os deslocamentos das pessoas e a sua orientação oferece vantagem aos profissionais que lá trabalham dando assim informações úteis. Exemplo: determinação das preferências e escolhas das pessoas.

Tabela 4-1 – Descrição do Problema

Nesta tabela, identificam-se alguns dos diversos ambientes alvo e o impacto associado ao facto de não existir uma ferramenta informática que permita analisar as imagens.

Desta forma procura-se aumentar a recolha de informação acerca das áreas abrangidas pelas câmaras sem necessidade de observação ou perda de tempo na análise por parte dos utilizadores.

4.2. Metodologia de Engenharia de Software utilizada

De entre os vários modelos de engenharia de software existentes como o ciclo de vida clássico (método Waterfall), modelo iterativo e outros, o método escolhido para a realização desta aplicação é o da prototipagem. Este modelo envolve a criação de um protótipo executável mesmo que ainda não haja o conhecimento de todos dados necessários para a completa implementação do sistema. Desta forma, vão sendo produzidas partes do sistema onde estas sofrem diversas modificações até alcançar o objectivo principal.

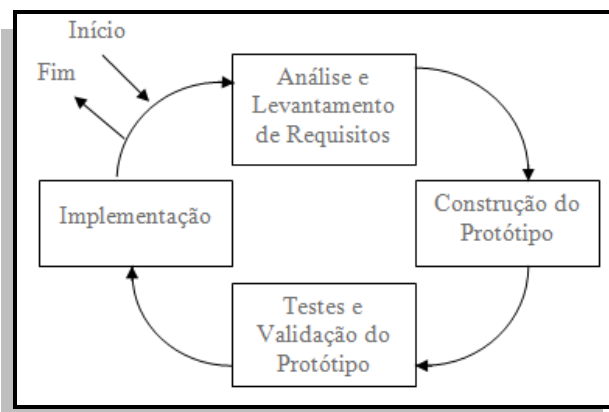


Figura 4-1 – Modelo de Prototipagem

4.3. Análise do Sistema

De forma a cumprir os objectivos propostos, apresenta-se a seguir todos os requisitos funcionais e não-funcionais que presidiram ao desenvolvimento das aplicações referidas.

4.3.1. Requisitos Funcionais das Aplicações

A aplicação que efectuará a detecção de faces em ficheiros de vídeo existentes deverá permitir ao utilizador o processamento de qualquer vídeo desejado, de acordo com os requisitos da aplicação específica que o utilizador pretenda dar à informação sobre a detecção e o seguimento das faces humanas.

Ambas as aplicações devem permitir ao utilizador analisar a informação recolhida de uma face à sua escolha após o processamento de um vídeo ou de uma recolha em tempo real. Esta informação será composta pelo trajecto da face detectada que pertence à pessoa em questão, e por informação sobre as orientações que foram tomadas pela face dessa pessoa durante a análise facial. Essencialmente, identificar-se-ão os lados para os quais a pessoa olhou (esquerda ou direita), bem como o tempo em que a face se demorou com cada uma dessas orientações.

Para além disso, ambas as aplicações terão que permitir visualizar a recolha de dados de forma geral ao nível das orientações. Isto significa que devem indicar claramente quantas orientações ocorreram no sentido da direita e no sentido da esquerda em todas as detecções faciais verificadas durante a análise do vídeo.

4.3.2. Requisitos Não-Funcionais

No que diz respeito aos requisitos Não-Funcionais, identificaram-se os seguintes: Facilidade de Uso, Detecção Automática, Visualização da Informação e Rapidez. Na tabela 4-2 pode consultar-se uma definição detalhada destes requisitos.

Requisito	Especificação do Requisito
Facilidade de Uso	As aplicações devem ser simples de utilizar e devem exigir um grau de aprendizagem tão reduzido quanto possível. O menu principal das aplicações deve ser facilmente acedido após o processamento das <i>frames</i> de vídeo e após a escolha de um dos submenus.
Detecção Automática	A detecção de faces, bem como os respectivos elementos característicos, neste caso os olhos e o nariz, deverá ser feita automaticamente sem a ajuda, ou interferência, do utilizador.
Visualização da Informação	As informações relevantes recolhidas durante o processamento deverão ser disponibilizadas ao, e visualizadas pelo, utilizador.
Rapidez	Pretende-se ainda que as aplicações possibilitem um processamento de <i>frames</i> de forma consideravelmente rápida para que a visualização em tempo real não seja comprometida e não existam atrasos na visualização do vídeo.

Tabela 4-2 – Requisitos Não-Funcionais

Apesar dos requisitos da tabela serem Não-Funcionais estes não deixam de ser importante nas aplicações, devido ao facto de apesar de não haver uma interacção com os utilizadores, estes devem sentir satisfação na sua utilização.

4.4. Abordagem ao Problema

Tendo em conta os requisitos recolhidos e especificados acima, especifica-se agora o tipo de reconhecimento humano que mais se adequa a uma aplicação deste género e que respeite as características identificadas acima.

4.4.1. Sistema de Reconhecimento Humano

Após a revisão bibliográfica efectuada sobre as técnicas específicas que têm sido utilizadas para o reconhecimento de faces humanas, incluindo, entre outras, a subtracção do fundo e filtros com informações da cor da pele, o método escolhido para atacar o problema descrito nesta dissertação foi o da utilização de classificadores *Haar* em cascata proposto em Viola & Jones (2001) com os melhoramentos propostos em Lienhart & Maydt (2002).

Uma das razões que motivaram esta escolha deveu-se ao facto destes classificadores poderem ser utilizados de forma directa, isto é, ao facto de realizarem a detecção nos dados após estes serem disponibilizados pelo utilizador sob a forma de um ficheiro de vídeo, ou directamente na *stream*¹³ de vídeo proporcionada por uma câmara de vídeo, permitindo assim a possibilidade de efectuar detecções em tempo real.

Outras características que motivaram a preferência pelos classificadores *Haar* incluem o bom desempenho e a eficiência da detecção, e o baixo custo de computação identificados por Benezeth et al. (2009). Note-se que a capacidade de aumentar a eficiência da detecção através de ajustes ao número de classificadores em cascata é ainda outro dos factores positivos que abonam em favor desta escolha (Viola & Jones 2001).

Para finalizar, apesar de se verificar a existência de vários tipos distintos de classificadores, outra das vantagens da utilização dos classificadores de *Haar* é a capacidade que evidenciam para serem treinados para efectuar detecções dos objectos pretendidos através de um conjunto de amostras positivas e negativas. Porém, esta

¹³ Stream é o fluxo de dados de um sistema computacional.

característica pode ser igualmente considerada como uma desvantagem já que, caso o classificador não seja devidamente treinado, existe a possibilidade real de ocorrerem mais erros e falsas detecções.

4.4.2. Sistema de Armazenamento de Dados

Apresenta-se agora a estrutura de dados escolhida para o armazenamento da informação obtida, definindo-se a seguir a informação que é guardada pelas aplicações desenvolvidas.

A estrutura de dados seleccionada para o armazenamento foi a lista, pois, para além de permitir uma organização dos dados de forma encadeada, possui também os requisitos necessários para as aplicações, já que proporciona uma forma rápida e eficaz de armazenar temporariamente os dados em memória.

A figura 4-2 ilustra a estrutura das listas ligadas desenvolvida para este efeito. Note-se que cada nó¹⁴ da Lista de Faces possui informações relativas à face detectada de cada pessoa. A segunda lista contém todas as coordenadas adquiridas nas detecções de faces, olhos e narizes de cada pessoa.

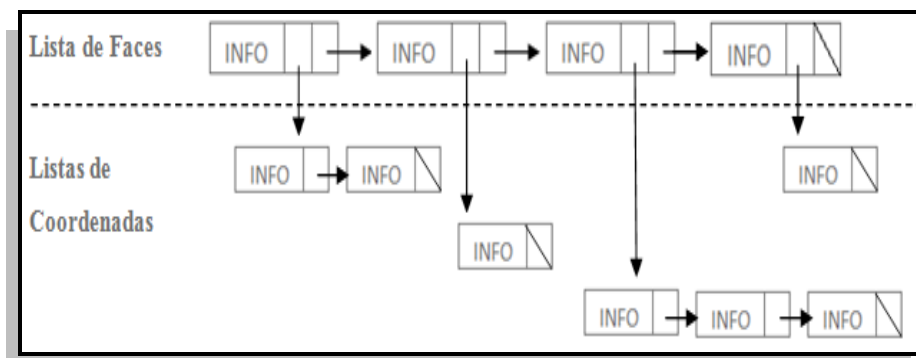


Figura 4-2 – Estrutura das Listas Ligadas

No que diz respeito à informação contida em cada nó existente nas listas ligadas descritas acima, atente-se à descrição apresentada na tabela 4-3.

¹⁴ Nó é a estrutura principal de uma lista que contém um ou mais campos de informação (INFO) e um ou mais campo de ligação/ELO (representado pela saída das setas na figura 4-2).(Vasconcelos & Carvalho 2005)

Nó da Lista	Nome	Descrição
<u>Faces</u>	ID	Valor que permite identificar uma face de forma única.
	NºFrames	Representa o número de <i>frames</i> que passaram desde a primeira detecção de uma determinada face.
	NºPontos	Número total de pontos guardados na lista de coordenadas desde a primeira detecção de uma determinada face.
	Fim	Variável binária que permite determinar a finalização de detecções referentes a uma determinada face.
	NóPontos	Elo de ligação à lista de coordenadas.
	Elo	Elo de ligação às faces seguintes.
<u>Coordenadas</u>	FaceX	Coordenada referente ao eixo das abcissas X de uma detecção facial que representa o topo superior esquerdo da detecção.
	FaceY	Coordenada referente ao eixo das ordenadas Y de uma detecção facial que representa o topo superior esquerdo da detecção.
	NarizX	Coordenada referente ao eixo das abcissas X de uma detecção nasal que representa o centro da detecção.
	NarizY	Coordenada referente ao eixo das ordenadas Y de uma detecção nasal que representa o centro da detecção.
	OlhoEsqX	Coordenada referente ao eixo das abcissas X representante do centro da detecção do olho esquerdo.
	OlhoEsqY	Coordenada referente ao eixo das ordenadas Y representante do centro da detecção do olho esquerdo.
	OlhoDirX	Coordenada referente ao eixo das abcissas X representante do centro da detecção do olho direito.
	OlhoDirY	Coordenada referente ao eixo das ordenadas Y representante do centro da detecção do olho direito.
	DirEsq	Variável que permite guardar a orientação de uma face.
	Elo2	Elo de ligação às coordenadas seguintes.

Tabela 4-3 – Lista de Variáveis/Informações dos Nós

É importante referir que a recolha dos dados enunciados na tabela 4-3 é muito importante, já que é por seu intermédio que é possível obter informação sobre os percursos das pessoas no espaço físico observado, bem como as respectivas orientações. Do mesmo modo, permitem igualmente controlar o processo da detecção na medida em que identificam as faces de pessoas que já foram detectadas em *frames* anteriores do vídeo que se está a processar.

A figura 4-3 apresenta um exemplo que mostra que tipo de informação é guardada e como funcionam as ligações entre as várias listas.

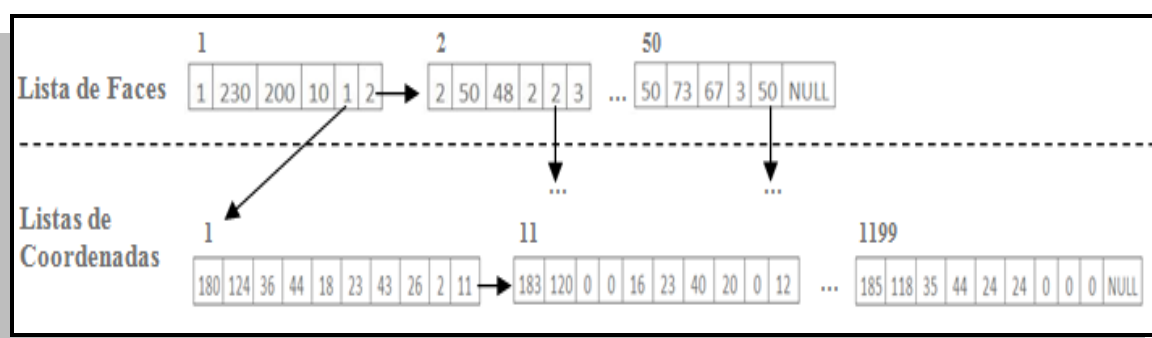


Figura 4-3 – Exemplo de Dados Armazenados nas Listas

Na lista de faces estão armazenadas 50 faces, o que significa que cada face é de uma pessoa diferente. No caso das listas de coordenadas no exemplo acima significa que a pessoa com a face cujo ID é 1 foi detectada 200 vezes, durante essa detecção poderá conter ou não informação acerca das regiões nasais, oculares e também da orientação dessa face.

5. Implementação do Sistema

Neste capítulo descreve-se o modo como as aplicações introduzidas anteriormente foram implementadas. Neste sentido, identificam-se, em primeiro lugar, todos os passos efectuados para a criação das aplicações de modo a satisfazer os requisitos recolhidos e enunciados no capítulo 4. Descrevemos aqui a implementação da aplicação *FaceDetectVC*, e da aplicação *FaceDetectRT*. Ambas foram implementadas recorrendo à ferramenta de desenvolvimento *Visual Studio 2008* e à biblioteca *OpenCV v2.0* da *Intel*. Para finalizar, apresenta-se um diagrama de fluxo que é comum às duas aplicações criadas.

5.1. Aplicação FaceDetect

Para a criação das aplicações com o *Visual Studio 2008*, começou-se por criar um novo projecto através do menu *File* e opção *New Project* que permite escolher um template entre vários que se encontram à disposição, neste caso escolheu-se um projecto vazio tendo como linguagem de programação para a escrita do código *C++* (Figura 5-1).

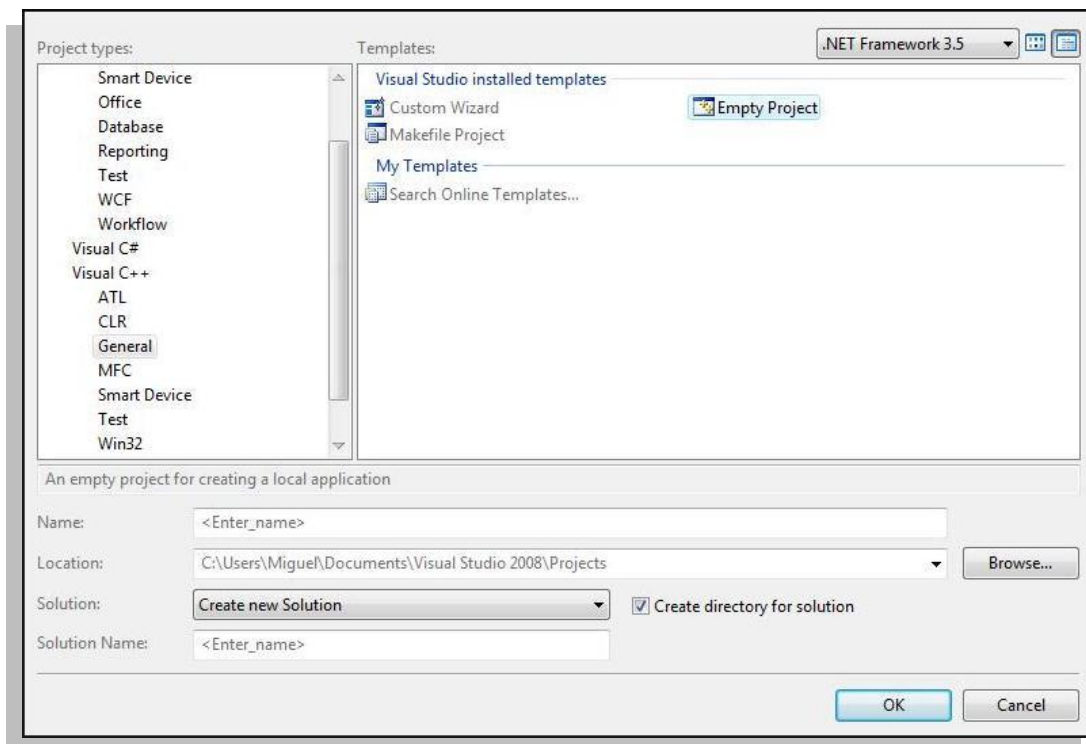


Figura 5-1 – Criação Novo Projecto

Ao ser efectuada esta escolha o projecto é criado automaticamente com o nome escolhido e com três pastas vazias, *Header Files*, *Resource Files* e *Source Files*. A pasta *Source Files* terá o ficheiro que contém o respectivo código C++ onde o ficheiro é criado através da selecção do projecto com o botão direito do rato e a escolha da opção *Add New Item* onde será possível seleccionar nas categorias *Visual C++* o tipo de ficheiro que pretendemos criar, neste caso, um ficheiro em branco C++ para a composição do código (Figura 5-2).

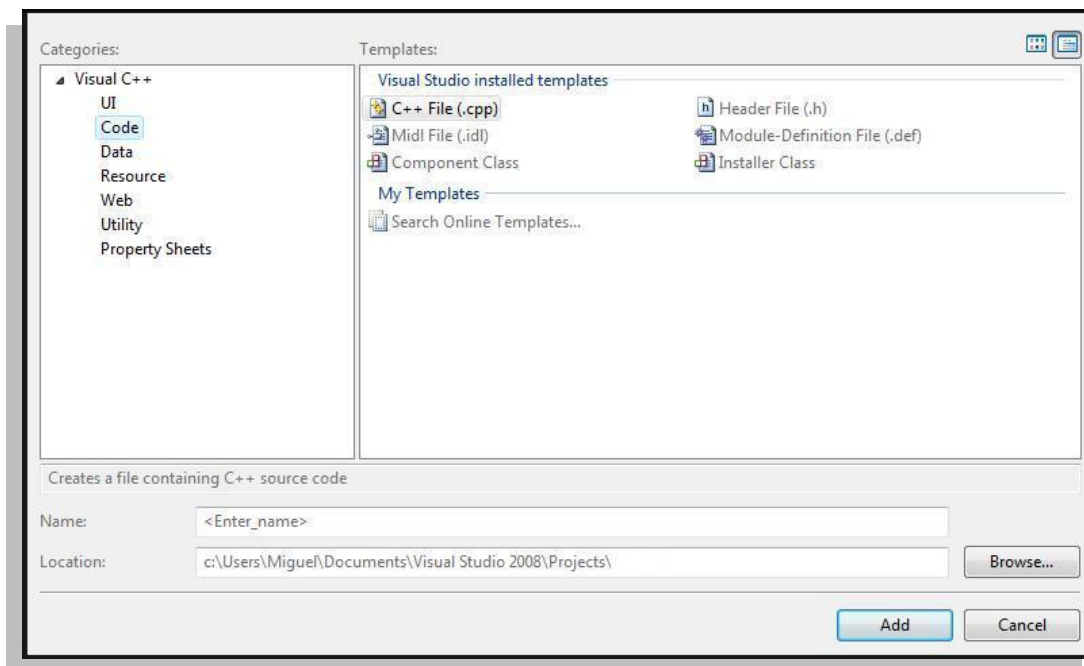


Figura 5-2 – Criação Ficheiro C++

Embora já esteja tudo pronto para iniciar a programação, é necessário ligar o projecto às bibliotecas de *OpenCV*¹⁵ através da selecção do projecto com o botão direito do rato e no menu escolher *Properties*. Após a selecção, surgirá um menu onde se opta por *Configuration Properties*, *Linker* e *Input* no lado esquerdo do mesmo. Esta selecção coloca-nos na parte direita do menu na área onde se une as bibliotecas ao projecto. Para o fazer, basta colocar o nome de cada uma das bibliotecas na caixa de texto *Additional Dependencies*, desta forma o projecto já pode recorrer à utilização das bibliotecas disponibilizadas pelo *OpenCV* (Figura 5-3).

¹⁵ A informação sobre como instalar devidamente a ferramenta *OpenCV* encontra-se disponível em: “http://nashruddin.com/OpenCV_2_0_Installation_on_Windows_XP_and_Visual_Studio_2008”.

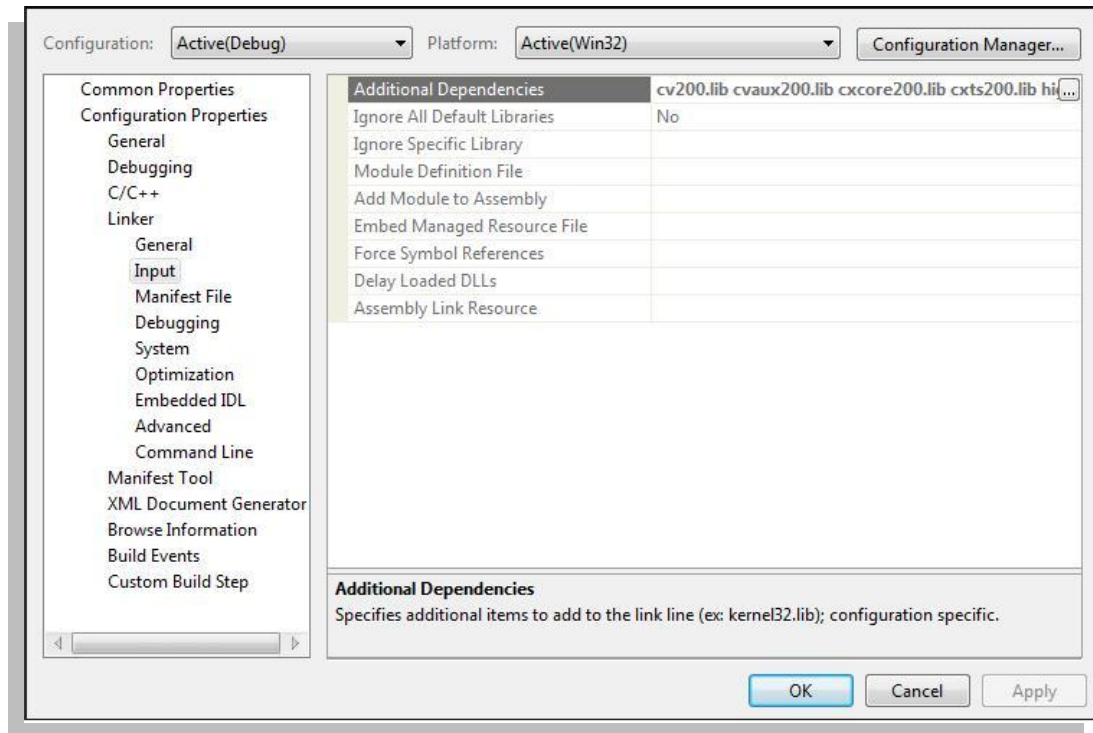


Figura 5-3 – Ligação do Projecto às bibliotecas *OpenCV*

Apresenta-se a seguir a concepção detalhada destas aplicações, descrevendo-se, para cada uma delas, a forma como foram implementadas desde estruturas, funções e variáveis.

5.1.1. Inicialização e Fases das Aplicações

De modo a simplificar a utilização das aplicações, o processamento das imagens para identificar as faces é iniciado logo após o arranque de cada uma das aplicações. Este processamento não requer nenhuma actividade por parte do utilizador, ou seja, as detecções são feitas automaticamente e o utilizador limita-se apenas a visualizar partes das informações adquiridas pelas aplicações. A inicialização da aplicação FaceDetectRT é feita através do ambiente do ambiente DOS onde é necessário especificar o local onde se encontra o ficheiro executável seguido do nome da aplicação (FaceDetectRT.exe). No caso da aplicação FaceDetectVC a sua inicialização é feita de forma idêntica à aplicação anterior mas necessita de receber o nome e a localização do vídeo a analisar (Figura 5-4).

Detecção e seguimento automáticos de faces humanas em vídeos capturados em ambientes não controlados

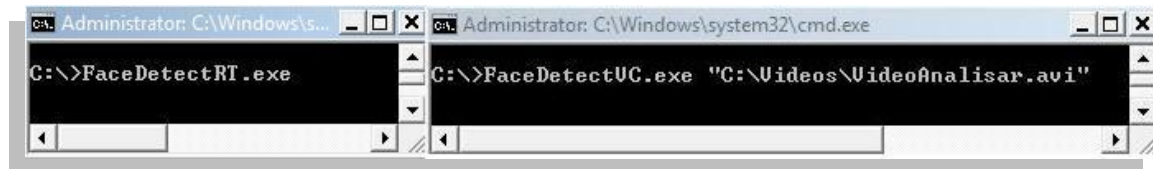


Figura 5-4 – Exemplo da Execução das Aplicações

Para ambas as aplicações a descrição que se segue detalha o trabalho desenvolvido em cada uma das seguintes fases: análise de *frames*, visualização de *frames*, eliminação de faces, incrementação e menu de resultados adquiridos após a execução das aplicações.

Na tabela 5-1 encontra-se mais detalhadamente a função de cada uma delas.

Fase	Especificação do Funcionamento
Análise de <i>Frames</i>	Fase onde cada <i>frame</i> recebida por recolha directa de uma câmara de vídeo, ou adquirida a partir de um ficheiro de vídeo, é analisada de forma a identificar a face, os olhos e o nariz de uma pessoa. Cada identificação será assinalada através da inscrição de: um rectângulo sobre a <i>frame</i> de vídeo no caso da face, um círculo para identificar a localização de ambos os olhos, e uma linha recta desde o centro da margem superior do rectângulo até ao centro do nariz para identificar a orientação da face. Ainda nesta fase, faz-se a inserção dos dados recolhidos na lista ligada de acordo com a informação que é adquirida durante o reconhecimento, determinando-se igualmente a orientação da face.
Visualização de <i>Frames</i>	Tal como o nome indica esta fase efectua somente a visualização da <i>frame</i> que foi analisada com as identificações assinaladas pela fase anterior.
Eliminação de Faces	Efectua a eliminação de faces ou falsas detecções que se encontram guardadas na lista ligada de acordo com um conjunto de regras.
Incrementação	Permite registar o número de <i>frames</i> recolhidas após a detecção de cada face, bem como, incrementar o valor para definir quando essa face deixa de existir ou obter mais informação.
Menu de Resultados	Dá a possibilidade ao utilizador de visualizar os resultados adquiridos no processamento do vídeo ou da realidade.

Tabela 5-1 – Fases das Aplicações

5.1.2. Análise de *Frames*

Esta secção apresenta os vários passos que foram necessários tomar para efectuar a análise de uma *frame* no sentido de detectar faces tanto em tempo real, como também, em vídeos já existentes. Cada passo descrito abaixo não só irá corresponder à aplicação *FaceDetectRT* mas também, corresponde à aplicação *FaceDetectVC* tendo apenas uma única diferença, esta é, a forma como as aplicações obtêm a *frame* para análise.

Inicialmente foi necessário associar variáveis para cada tipo de ficheiro de classificadores *Haar* a utilizar. Estes classificadores constituem a base principal da detecção facial pois cada um deles efectua os cálculos que foram definidos no método sugerido em Viola & Jones (2001). Os dados contidos nos classificadores são representações da “imagem integral” de amostras positivas e negativas de um objecto a detectar. Neste caso, os objectos de interesse são as faces, os narizes e os olhos das pessoas. Sabendo da importância da obtenção de um classificador bem construído, optou-se por utilizar os classificadores *Haar* identificados na tabela 5-2 que são fornecidos pela *Intel* através da instalação da biblioteca *OpenCV*. Estes foram:

Ficheiro de Classificadores	Descrição
haarcascade_frontalface_alt2.xml	Ficheiro que contém classificadores em cascata referentes às representações faciais numa orientação frontal.
haarcascade_mcs_lefteye.xml	Ficheiro que contém classificadores em cascata referentes às representações do olho esquerdo de uma pessoa.
haarcascade_mcs_righteye.xml	Ficheiro que contém classificadores em cascata referentes às representações do olho direito de uma pessoa.
haarcascade_mcs_nose.xml	Ficheiro que contém classificadores em cascata referentes às representações nasais de uma pessoa.

Tabela 5-2 – Ficheiros dos Classificadores *Haar* Utilizados

Após especificados os ficheiros que são necessários para as detecções, o próximo passo é a captura do ficheiro que queremos analisar. Essa captura é feita de forma muito simples, ou seja, basta utilizar a função “`cvCreateFileCapture(argv[1])`” no caso

de vídeos guardados, onde esta recebe como argumento o nome do vídeo e retorna um ponteiro para uma estrutura “CvCapture”. Essa estrutura contém todas as informações acerca do vídeo. Desta forma, já é possível obter uma *frame* para análise. No caso da recolha directa da realidade, utilizou-se a função “cvCreateCameraCapture(0)” onde apesar de retornar também um ponteiro “CvCapture” esta não necessita de receber argumentos, pois, o argumento necessário é o Id da câmara mas só é utilizado em casos de múltiplas câmaras.

No decorrer da captura, cada *frame* recebida é analisada de forma individual e reescrita no sentido de indicar ao utilizador as detecções que foram realizadas. Nesta fase de análise, as aplicações começam por criar uma cópia da *frame* original com um tamanho reduzido, sendo as cores convertidas para uma escala de tons de cinzento após aplicar o grau de prioridade BGR¹⁶. De forma a aumentar a eficácia da detecção, utilizou-se uma função para equalizar o histograma¹⁷ que permite normalizar as escalas de brilho. Dado que o método proposto em Viola & Jones (2001) possui informações de brilho acerca de um determinado objecto a detectar, isto é necessário para ajudar a reconhecer com mais facilidade esse objecto que, neste caso, se trata de uma face humana. No que diz respeito à necessidade de aplicar o redimensionamento da imagem, o facto de a imagem ser mais pequena contribui para aumentar o desempenho das detecções devido ao facto de os classificadores terem deste modo menos áreas a percorrer do que na imagem original.



Figura 5-5 – Tons de Cinzento/ Tons de Cinzento com equalização do histograma

¹⁶ BGR(Blue, Green, Red) corresponde às cores primárias azul, verde e vermelho de aparelhos electrónicos onde as áreas mais significativas correspondem às áreas azuis e a menos significativas às vermelhas.

¹⁷ Histograma é um diagrama que representa a frequência de ocorrência dos brilhos de uma imagem. (Pinho & Manuel 2005)

A este respeito, a figura 5-5 mostra claramente a diferença de claridade entre uma imagem sem equalização (à esquerda) e outra com equalização (à direita).

5.1.2.1. Detecção Facial/ Nasal/Ocular

Tomadas as medidas descritas acima, inicia-se a pesquisa de faces através da função “`cvHaarDetectObjects()`” que aplica o método descrito em Viola & Jones (2001). Esta função permite descobrir áreas rectangulares na imagem que são idênticas aos objectos guardados nos classificadores *Haar*. Assim, compara a área da imagem com cada etapa dos classificadores; caso esta passe em todas as etapas, e se os objectos dos classificadores corresponderem a faces humanas, então o objecto analisado tem grande probabilidade de ser uma face humana. A análise da imagem é efectuada várias vezes com diferentes escalas. Em casos de sobreposição de imagem, tais como a análise de áreas onde já foi efectuada uma análise pelo classificador, a função aplica heurísticas¹⁸ de forma a reduzir o número de regiões analisadas. No final da pesquisa, a função retorna uma sequência de rectângulos que são susceptíveis de conter o objecto para o qual o classificador está treinado. Os parâmetros necessários para o seu funcionamento indicam-se na tabela 5-3.

Parâmetros	Descrição
Image	<i>Frame</i> na qual se pretende efectuar a detecção.
Cascade	Variável que representa os classificadores <i>Haar</i> que se pretende usar para identificar o objecto desejado.
Storage	Zona de memória onde é guardada a informação sobre as sequências de rectângulos aspirantes ao objecto.
Scale_factor	Factor que determina o aumento da janela de pesquisa após analisar a imagem completamente. Quanto maior for esse valor, menor será o tempo de processamento mas com o custo de uma maior probabilidade de ocorrência de falhas nas detecções em casos em que a escala falha objectos de determinados tamanhos.
Min_Neighbors	Evita falsas detecções através da identificação de uma única face em áreas onde existem várias sobreposições, isto é, evita a identificação da mesma face várias

¹⁸ Heurística é a utilização de informações baseadas na experiência que ajudam a solucionar os problemas de forma rápida.

	<p>vezes entre pixéis adjacentes ou através das pesquisas com diferentes escalas.</p> <p>Tendo várias detecções feitas numa determinada área, este agrupa todas as detecções numa só detecção.</p>
Flags	<p>Permite definir o modo de funcionamento da pesquisa.</p> <p>Existem quatro configurações possíveis:</p> <ul style="list-style-type: none"> • CV_HAAR_DO_CANNY_PRUNING: ignora regiões planas (sem linhas); • CV_HAAR_SCALE_IMAGE: algoritmo passa a fazer o escalonamento da imagem em vez do detector; • CV_HAAR_FIND_BIGGEST_OBJECT: retorna apenas o maior objecto encontrado durante a pesquisa; • CV_HAAR_DO_ROUGH_SEARCH: termina imediatamente a pesquisa após obter uma única detecção.
Min_Size	<p>Tamanho mínimo do objecto a encontrar.</p> <p>O tamanho escolhido deverá ser de acordo com o tamanho mínimo do objecto que se pretende detectar.</p> <p>Quanto maior for este valor, menor será o tempo de processamento mas inclui perdas nas detecções de objectos com tamanhos inferiores ao tamanho definido.</p>

Tabela 5-3 – Parâmetros da Função cvHaarDetectObjects

A configuração utilizada para as detecções faciais foi a seguinte: Scale_factor = 1.1, o que significa que o aumento da janela ao fim de cada pesquisa será de 10%; Min_neighbors ficou definido como 3, sendo este o valor padrão. No que diz respeito à configuração da pesquisa, foi definida para ignorar as regiões sem linhas, e o tamanho mínimo do objecto foi colocado como 20×20, sendo estes os valores mínimos que os classificadores possuem para a identificação de uma face, levando a que seja necessário mais tempo de processamento, mas apresentando a vantagem de identificar faces de qualquer tipo de tamanho.

De forma a melhor explicar os procedimentos que conduzem à detecção de uma face, recorreu-se à informação fornecida pela aplicação descrita em Harvey (2010) que permite visualizar o funcionamento das detecções (consultar figura 5-6).



Figura 5-6 – Funcionamento da Detecção (adaptado de: Harvey, 2010)

A figura 5-6 ilustra os passos necessários que resultam no reconhecimento facial através do cálculo da imagem integral. Assim, partindo dos valores da imagem integral, inicia-se a procura de áreas da imagem com características semelhantes às características dos classificadores. A procura está representada na figura pelos rectângulos com cores pretas e brancas que estão contidas no rectângulo vermelho (janela de procura). Quando são encontradas características, a função deixa uma cópia da janela de pesquisa no local onde esta foi encontrada, passando de seguida para a posição seguinte. A janela de procura só avança quando são analisadas todas as etapas possíveis do classificador. Chegando ao final da imagem, é feito um redimensionamento da janela de procura e o ciclo reinicia-se. Na fase final do processamento, todos os rectângulos assinalados como uma potencial face são verificados de modo a não ocorrerem sobreposições. Se existirem dois ou mais rectângulos sobrepostos, estes contribuem para confirmar a

existência de uma face naquele local, sendo então substituídos por apenas um rectângulo. Os falsos positivos são todos aqueles rectângulos que se encontram sozinhos (sem sobreposições), sendo então excluídos.

De modo a que o utilizador possa visualizar as faces encontradas, as aplicações descritas nesta dissertação utilizam as coordenadas dos rectângulos reconhecidos pela função anterior para desenhar na imagem original o rectângulo que representa a face. Estas coordenadas são necessárias não só para a representação da face, mas também para diminuir a área de procura relativamente aos reconhecimentos oculares e nasais (ver figura 5-7). Isto acontece porque ao determinarmos que o rectângulo contém uma face, então só é necessário efectuar a análise ocular e nasal naquela área, fazendo com que as aplicações não desperdicem tempo de processamento desnecessariamente em áreas da imagem que não contêm faces.

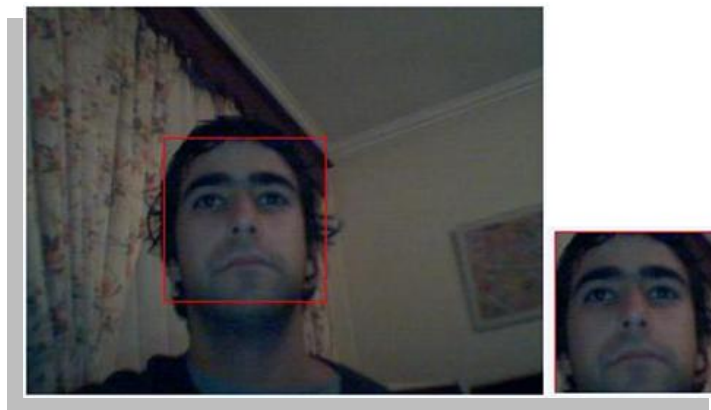


Figura 5-7 – Imagem Original com Reconhecimento Facial (esquerda) e Área da pesquisa Nasal e Ocular (direita).

O reconhecimento dos olhos e do nariz é efectuado em cada face encontrada, da mesma forma que o reconhecimento de faces. Contudo, requer outro tipo de configuração para melhorar a obtenção de resultados. Assim, o *Scale_factor* para o nariz ficou definido como 1.2, e para ambos os olhos como 1.15. No que diz respeito ao valor *min_neighbors*, para o nariz ficou definido que não ocorrerão agrupamentos (valor 0) e no caso dos olhos está definido como 1. Para finalizar as configurações, ambas as *flags* da configuração de pesquisa estão definidas para excluir áreas planas e o tamanho dos objectos estão definidos como 18×15 para o nariz e como 18×12 para ambos os olhos.

Tal como no reconhecimento facial, ambos os olhos e o nariz recebem coordenadas rectangulares no caso de reconhecimento, sendo estas utilizadas para a criação de

círculos no caso dos olhos e uma linha que se prolonga desde o centro do rectângulo da detecção do nariz até ao topo da área de pesquisa.



Figura 5-8 – Reconhecimento Facial/Ocular/Nasal

Na figura 5-8 apresenta-se um exemplo de uma detecção facial juntamente com o reconhecimento de ambos os olhos e do nariz. Esta imagem é visualizada imediatamente após a fase de análise terminar.

5.1.2.2. Armazenamento de Dados/Informações

O armazenamento dos dados nas listas é efectuado assim que todas as detecções faciais, oculares e nasais estejam concluídas. Cada face identificada é analisada através das coordenadas x e y do canto superior esquerdo do rectângulo que representa a detecção (cf. figura 5-9).



Figura 5-9 – Ponto da Detecção Facial Analisado

A análise efectuada pelas aplicações a tal ponto (canto superior esquerdo) permite determinar se a face encontrada na *frame* actual já foi reconhecida numa *frame* anterior. Esta análise é feita através do factor proximidade. De facto, as coordenadas da detecção actual são comparadas com as coordenadas das detecções anteriores, efectuando-se a respectiva subtracção. Esta subtracção é efectuada apenas nas últimas coordenadas de cada face existente na lista, determinando-se se o valor da diferença é ou não muito

elevado. Caso o valor seja superior a 20, nas coordenadas x ou y, relativamente a todas a faces contidas na lista, isto significa que a detecção actual está afastada das detecções encontradas nas *frames* anteriores, fazendo com que as aplicações criem um novo nó no final da lista de faces (detectou uma nova face) e criem também uma lista de coordenadas associadas à nova face. Se o valor for inferior a 20, trata-se de uma face já detectada anteriormente, pelo que as coordenadas serão guardadas nessa face, sendo criado somente um novo nó na lista de coordenadas e fazendo-se o incremento de um valor ao número de pontos pertencentes à face.

Ainda no sentido de otimizar o armazenamento de faces para que as coordenadas de uma face não sejam colocadas numa face diferente, criou-se uma variável do tipo contador que permite controlar se uma pessoa deixou de aparecer nas *frames* seguintes ou não. Cada face, após ser inserida na lista de faces, herda um contador designado por “fim”. Este contador é incrementado se não existirem novas coordenadas para adicionar à face após a análise de uma *frame*. Se o valor do “fim” for igual ou superior a 10 *frames*, a face associada deixa de obter novas coordenadas, já que é muito provável que essa pessoa tenha abandonado a área abrangida pela câmara. Por exemplo, supondo que uma câmara de videovigilância capta parte de um corredor de um supermercado no qual diversas pessoas são detectadas e guardadas na lista de faces. Após algum tempo, essas pessoas deixam de ser captadas pela câmara (movimentam-se para outro local) e novas pessoas irão naturalmente surgir. Se o sistema de finalização de faces não existisse, as novas pessoas, ao serem reconhecidas em locais semelhantes às detecções das pessoas anteriores, seriam colocadas nas listas como uma coordenada de uma face existente e não como uma nova face. Caso o contador de uma face não tenha atingido o limite, e essa face obteve uma nova coordenada, o contador reinicia então o seu valor.

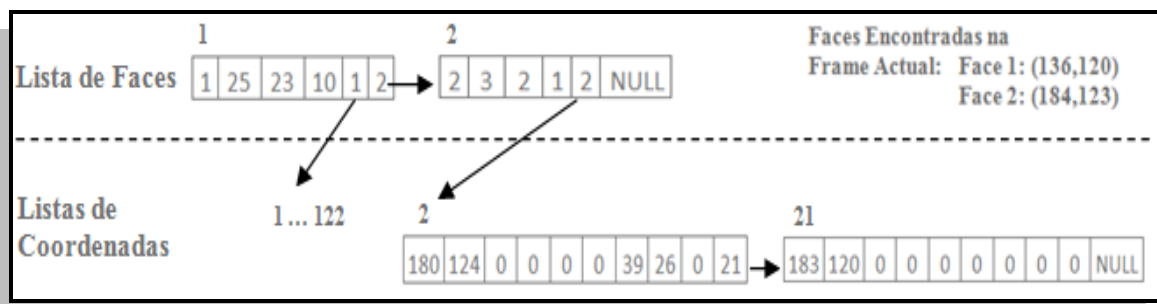


Figura 5-10 – Inserção de Valores nas Listas

Como se pode verificar no exemplo ilustrado na figura 5-, aqui a lista de faces já possui armazenadas duas faces: para a primeira face existem vinte e três coordenadas dessa face; para a segunda face existem apenas duas. Neste exemplo, as aplicações detectaram duas coordenadas correspondentes a faces (Face 1, Face 2). Assim, cada uma delas será analisada de forma a determinar se essa face já foi encontrada nas *frames* antecedentes ou não. Para cada face, começa-se por verificar se a variável “Fim” já atingiu o valor que define a finalização dessa face. Em caso afirmativo, verifica-se que a face cujo ID é 1 já contém o seu contador de finalização a 10, fazendo com que somente as coordenadas da face com ID 2 sejam analisadas. Sendo assim, verifica-se que a diferença de valores da Face 1 relativamente à face com ID 2 é de 47 (183-136) no eixo das abcissas x e 0 (120-120) no eixo das ordenadas y, concluindo-se pois que não pertence à mesma face. Em seguida cria-se um novo nó (ID 3) e uma lista de coordenadas associadas à nova face. No que diz respeito à Face 2, verifica-se que a diferença no eixo x é de apenas 1 e no eixo y é 3, concluindo-se assim que as coordenadas obtidas correspondem a uma face já existente na lista, criando apenas um novo nó na lista de coordenadas cujo ID será 22.

A informação que é armazenada relativamente aos olhos e ao nariz não requer uma procura pela face correcta, já que, dado que a respectiva detecção apenas é feita após a face ser detectada, as coordenadas dos olhos e do nariz ficam com a sua informação associada a essa face. Apesar desta associação, a informação recolhida acerca dos olhos só é guardada após ser analisada. Isto deve-se à necessidade de determinar a que olho pertence a detecção feita pelos classificadores (esquerdo ou direito). Apesar da utilização de dois ficheiros com classificadores, um destinado à detecção do olho esquerdo e o outro ao olho direito, pode acontecer que por vezes detectem ambos os olhos. Posto isto, a determinação do olho esquerdo ou direito é feita de forma simples através da obtenção da linha central da janela de pesquisa, sendo classificadas como olho esquerdo todas as detecções obtidas antes do centro e como olho direito todas as detecções obtidas após o centro (como se ilustra na figura 5-11).



Figura 5-11 – Centro da Janela de Pesquisa

De forma a otimizar as detecções feitas relativamente aos olhos foi necessário estabelecer prioridades de armazenamento dos dados, pois, nos casos em que um ficheiro de classificadores obtém a detecção de ambos os olhos foi necessário estabelecer quais é que seriam guardados. Sendo assim, como o reconhecimento do olho esquerdo se inicia em primeiro lugar, antes do olho direito, as detecções assim obtidas serão armazenadas na lista de coordenadas correspondentes à face. Contudo, as coordenadas que foram recolhidas relativamente ao olho direito podem vir a ser substituídas pelas detecções dos classificadores do olho direito. De facto, dado que os classificadores estão dedicados à obtenção do olho direito, estes têm necessariamente maior prioridade do que a detecção anterior. Desta forma, em casos para os quais os classificadores não identificam o olho a que se destinam, estes podem ser auxiliados pelos outros classificadores. A tabela 5-4 ajuda a entender melhor o que foi explicado anteriormente.

Classificadores do Olho	X olho esquerdo	Y olho esquerdo	X olho direito	Y olho direito
Esquerdo	18	23	41	24
Direito	16	22	43	26
Resultados Armazenados	<u>18</u>	<u>23</u>	<u>43</u>	<u>26</u>

Tabela 5-4 – Armazenamento de Dados Oculares

Na tabela acima verifica-se que os classificadores tanto do olho esquerdo como o do olho direito detectaram ambos os olhos, neste caso, os dados que são armazenados são os que correspondem ao seu próprio classificador, ou seja, os dados para o olho esquerdo são os obtidos pelos classificadores do olho esquerdo e os dados do olho direito são os obtidos pelos classificadores do olho direito.

Para além disso, para que os dados obtidos possam corresponder realmente ao olho de uma pessoa, foi feita uma filtragem pela localização dos olhos. Assim, todos os valores obtidos que não se encontram dentro da área estabelecida que contém os olhos serão eliminados.



Figura 5-12 – Limitação das Áreas Oculares

Essa área está representada na figura 5-12 (em cor amarela), sendo somente aceites valores que se localizam dentro do rectângulo representado.

Para finalizar o armazenamento dos dados, os valores obtidos para a localização do nariz são guardados após a obtenção das coordenadas fornecidas pelos classificadores, pois os dados referentes ao nariz não necessitam de qualquer tipo de análise adicional.

5.1.2.3. Determinação das Orientações

A determinação das orientações faciais em imagens de duas dimensões (2D) necessita de pelo menos três pontos de referência para a correcta determinação da orientação de uma face. Para este efeito, optou-se pela escolha dos pontos recebidos nas detecções de ambos os olhos e do nariz. A obtenção das orientações é feita através do cálculo da distância das coordenadas no eixo x de ambos os olhos relativamente ao valor x do nariz.

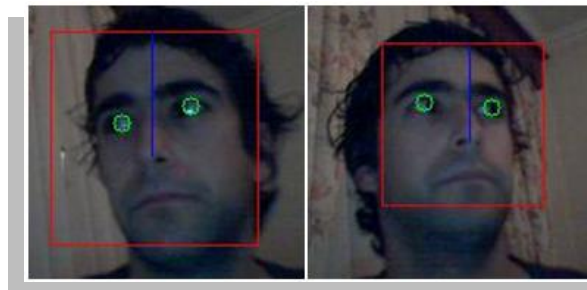


Figura 5-13 – Orientação à Esquerda/Orientação à Direita

A figura 5-13 permite ilustrar a detecção dos dois tipos de orientação da face do seguinte modo: quando uma pessoa possui uma determinada orientação, a identificação

do nariz surge cada vez mais próxima do olho correspondente à orientação. Isto significa que nas orientações à esquerda a distância entre o nariz e o olho esquerdo é menor que a distância entre o nariz e o olho direito, tal como é possível visualizar no primeiro caso da figura 5-13 (imagem da esquerda). Para o segundo caso ilustrado na figura 5-13 (imagem da direita), verifica-se precisamente o inverso do caso anterior: verificando-se um valor inferior para a diferença entre o olho direito e o nariz, determinamos que a orientação da face de uma dada pessoa é para a direita. Por exemplo, assumindo que o valor de x correspondente ao olho esquerdo é de 20 e o do olho direito é de 42, sendo o valor do x do nariz igual a 35, então a orientação da cabeça da pessoa será para o lado direito, pois a distância entre o valor x do olho direito e o valor x do nariz é menor que o valor da diferença entre o olho esquerdo e o nariz. Adquiridas estas informações, as aplicações guardam as orientações no campo “DirEsq” do nó da lista de coordenadas, sendo 1 o valor representante de uma orientação à esquerda e 2 o valor da orientação à direita.

No caso da falha na detecção de um dos três pontos de referência, este facto torna impossível determinar-se a orientação da cabeça, ou face, dessa pessoa, levando a que as aplicações guardem apenas os dados obtidos nas outras detecções positivas.

5.1.3. Visualização de *Frames*/Dados

Uma vez processados todos os dados sobre as detecções obtidas, segue-se a fase de visualização da *frame*, juntamente com alguns resultados que as aplicações obtiveram durante a análise dessa *frame*, permitindo que o utilizador possa, caso pretenda, retirar notas que lhe possam ser úteis durante a execução. Para cada pessoa detectada, ficam visíveis as áreas que representam a face, os olhos e o nariz respectivos. No sentido de identificar unicamente uma face, as aplicações apresentam ainda o ID correspondente a cada face detectada. Para além disso, também é possível visualizar as orientações que foram tomadas pelas pessoas contidas na imagem, onde cada orientação identificada é associada à pessoa a que pertence através do respectivo ID de face. Na figura 5-14 é possível verificar a apresentação destes dados que são disponibilizados durante a visualização de uma *frame*.



Figura 5-14 – Visualização de uma *Frame/Dados* (1)

A figura 5-14 ilustra ainda a detecção facial de cinco pessoas, sendo que somente em quatro delas foi possível obter a orientação da respectiva cabeça. Assim, houve uma pessoa (ID3) para a qual não se conseguiu obter as coordenadas dos três pontos de referência, o que não permitiu a obtenção da sua orientação. Embora na figura acima não se apresente o olho esquerdo (do ponto de vista da visualização) da pessoa cujo ID é 6 (olho direito da pessoa), e no caso da pessoa com o ID 5 não se identificar o olho direito, estes olhos foram contudo encontrados e armazenados na lista. Porém, não foram colocados na visualização devido ao facto de terem sido detectados pelos classificadores que estão destinados a outro efeito, isto é, os classificadores destinados à detecção do olho esquerdo detectaram igualmente o olho direito, mas os classificadores do olho direito não conseguiram encontrar o olho direito destas duas pessoas. Desta forma, ambos os olhos foram obtidos através dos classificadores do olho esquerdo. Contudo, apesar de existirem, o círculo correspondente ao olho direito não é desenhado na imagem para evitar que existam sobreposições de círculos, pois nos casos em que o olho direito é detectado pelos respectivos classificadores, ir-se-ia colocar outro círculo resultante dessa detecção, o que faria com que passassem a existir dois círculos para o mesmo olho.



Figura 5-15 – Visualização de uma *Frame/Dados* (2)

Na figura 5-15 já só foi possível detectar três faces e as suas respectivas orientações, isto porque, as duas faces que não foram detectadas é devido ao facto de no primeiro caso, só existe metade da face para analisar e no segundo caso, a face encontra-se com uma elevada rotação, dificultando assim a sua identificação.



Figura 5-16 – Visualização de uma *Frame/Dados* (3)

Nesta última figura (5-16), é feita a detecção das três pessoas da imagem com as suas respectivas orientações mas, como é possível verificar as mesmas pessoas já possuem Ids diferentes relativamente às imagens anteriores. Isto acontece ou porque as pessoas deixaram de aparecer durante dez *frames* ou porque quando surgiram, surgiram longe da posição que correspondiam à sua face.

5.1.4. Eliminação de Faces

A fase da eliminação de faces é importante para eliminar as falsas detecções que forem obtidas e armazenadas na lista durante a análise da *frame*. De facto, após se ter realizado uma análise exaustiva em vários vídeos guardados, verificou-se que as falsas detecções surgiram somente em duas ou três *frames* de forma consecutiva. Tendo em conta esta observação, determinou-se que as faces armazenadas na lista seriam eliminadas ao fim de cinco *frames* após a sua inserção na lista de faces caso não obtenham pelo menos quatro novos nós de coordenadas. Por exemplo, quando uma falsa detecção ocorre é criado um novo nó na lista de faces com a respectiva ligação à lista de coordenadas. Esta nova “suposta” face irá ser inicializada com o valor do número de pontos a um e o número de *frames* também a um. Supondo que na *frame* seguinte houve mais uma vez uma falsa detecção na mesma área, isso conduziu a que o número de pontos fosse incrementado para dois juntamente com o número de *frames*. Portanto, se nas próximas três *frames* a alegada face não obtiver pelo menos mais dois pontos, então será eliminada da lista. Após a sua eliminação, todas as faces que foram armazenadas depois da face eliminada verão os respectivos IDs decrementados de um para corresponderem correctamente ao número de faces guardadas nas listas de faces.

Para além disso, este procedimento ocorre não só para as falsas detecções como também para todas as faces que se encontram com menos de quatro pontos em cada grupo de cinco *frames* analisadas.



Figura 5-17 – Exemplo de uma Falsa Detecção/Falso Positivo

Na figura 5-17, apesar de estar representada uma falsa detecção (ID 4), esta não constará nos resultados finais após a análise completa das aplicações, pois assim que surgir a verificação da função de eliminação, verificar-se-á que a face correspondente

não obtém os pontos suficientes para continuar armazenada, pelo que a função efectuará a respectiva eliminação.

5.1.5. Incrementação

A função de incrementação, apesar de ser uma função simples, é muito importante para efectuar o controlo dos dados que são guardados nas listas. De facto, esta função é responsável por manter a contagem do número de *frames* que passaram de cada uma das faces após o primeiro armazenamento na lista, executando igualmente a incrementação da variável contador “fim”, disponibilizando com precisão os dados que são utilizados por outras funções.

5.1.6. Menu de Resultados

Finalmente, o menu de resultados que surge no final da visualização do vídeo permite ao utilizador a consulta de todas as detecções feitas durante o decurso do vídeo, ou durante a recolha de vídeo em tempo real. Este menu, para o caso da aplicação FaceDetectVC surge não só quando terminam as *frames* do vídeo, mas também, por opção do utilizador através da tecla *Esc* do teclado. Relativamente ao FaceDetectRT este só termina por ordem do utilizador, pois, como está ser feita uma recolha da realidade a aplicação não sabe quando há-de terminar, sendo assim, o menu só aparecerá quando o utilizador pressionar a tecla *Esc*.

Assim que surge o menu de resultados, o utilizador recebe informação sobre o número total de faces classificadas como diferentes que foram detectadas ao longo de todo o processamento, juntamente com três escolhas:

1. Visualizar o trajecto de uma face por ID;
2. Visualizar o número de Orientações Reconhecidas;
0. Saída.

5.1.6.1. Trajectos de uma Face

Neste caso, o utilizador tem à sua disposição a visualização de todas as informações sobre uma determinada face à sua escolha através da introdução do respectivo ID de face. A informação completa sobre as faces fica disponível assim que o utilizador escolha a face pretendida, apresentando-se de imediato toda a informação que foi

guardada nas listas durante o processamento e análise das *frames*. Tal informação diz respeito ao número de *frames* que passaram após ter sido feita a primeira detecção, o número de detecções obtidas para a respectiva face e, finalmente, todas as orientações que foi possível identificar ao longo de todo o percurso, juntamente com o intervalo de tempo tomado pela face em cada orientação. A obtenção deste intervalo de tempo é possível através do cálculo do número de *frames* consecutivas com obtenção de uma certa orientação e a informação acerca do número de *frames* que são processadas num segundo (fps) no vídeo. Por exemplo, se o *frame rate* do vídeo for de 20 fps, e supondo que foram identificadas 7 imagens consecutivas com orientação à esquerda, então o intervalo de tempo que a pessoa detectada esteve a olhar para a esquerda foi de $7/20 = 0,35$ segundos.

Para além destes dados, é ainda possível verificar qual foi o trajecto tomado pela pessoa na área abrangida pela câmara, sendo esse trajecto assinalado por uma linha vermelha juntamente com os pontos onde ocorreu a primeira identificação de uma orientação e as mudanças de orientação que foram sendo tomadas ao longo do tempo. As orientações para a esquerda são assinaladas por um círculo azul e as orientações para a direita são assinaladas por um círculo amarelo.

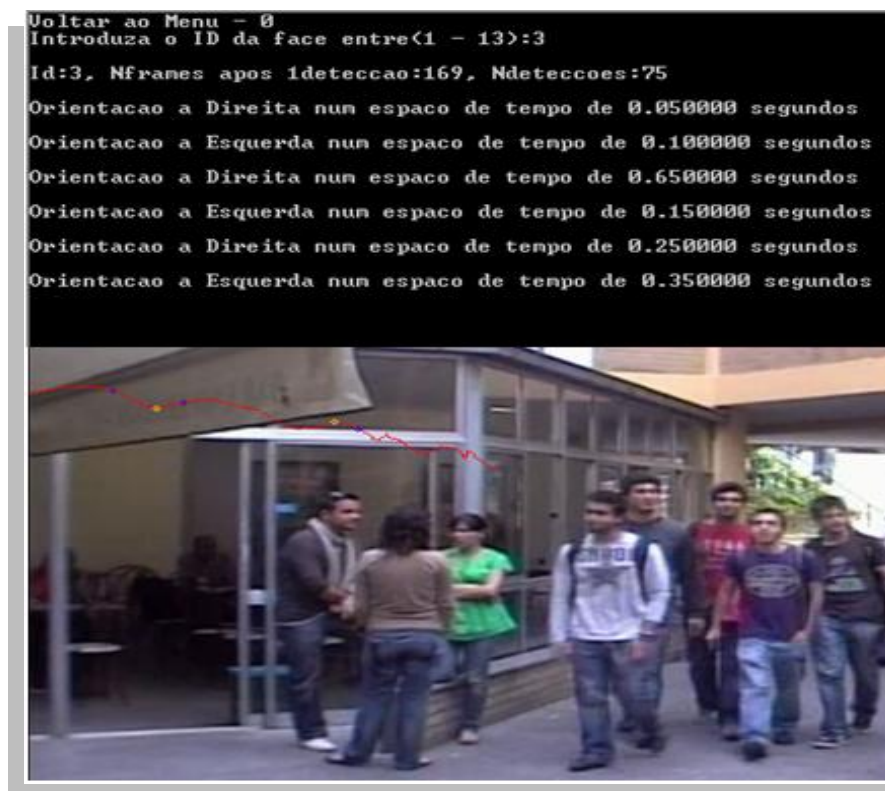
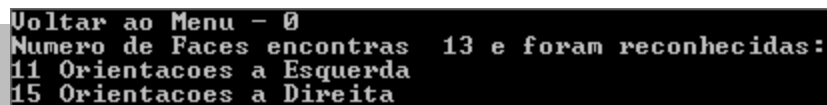


Figura 5-18 – Exemplo da Informação/Trajecto de uma Face

Por exemplo, a construção do trajecto de uma pessoa que é visível na figura 5-18 é obtido a partir de todas as coordenadas obtidas nas detecções faciais, isto é, todas as coordenadas do canto superior esquerdo da detecção facial ligadas entre si.

5.1.6.2. Número de Orientações Reconhecidas

Existe ainda um submenu (ver figura 5-19) que permite visualizar o número total de orientações que foram obtidas durante a análise e processamento das *frames*. A contagem é feita apenas se ocorrer uma mudança de orientação durante a pesquisa feita na lista de coordenadas, já que cada nó da lista pode ou não possuir informação sobre as orientações da face, tendo por vezes vários nós consecutivos com a mesma orientação. Sendo assim, seria desnecessário repetir a informação sobre essa orientação. O registo de outra orientação para a mesma face só é efectuado se a orientação obtida no nó anterior for diferente da actual. Desta forma, somente as mudanças de orientação serão contabilizadas para o resultado final.



```
Voltar ao Menu - 0
Numero de Faces encontras 13 e foram reconhecidas:
11 Orientacoes a Esquerda
15 Orientacoes a Direita
```

Figura 5-19 – Exemplo do Número de Orientações Reconhecidas

Na figura 5-19 é possível visualizar o resultado adquirido ao fim de todo o processamento e análise das *frames*.

5.2. Diagrama de Fluxo

Na figura 5-20 representa-se o diagrama de fluxo de ambas as aplicações, sendo possível verificar, de uma forma resumida, todos os passos percorridos até à obtenção dos resultados finais.

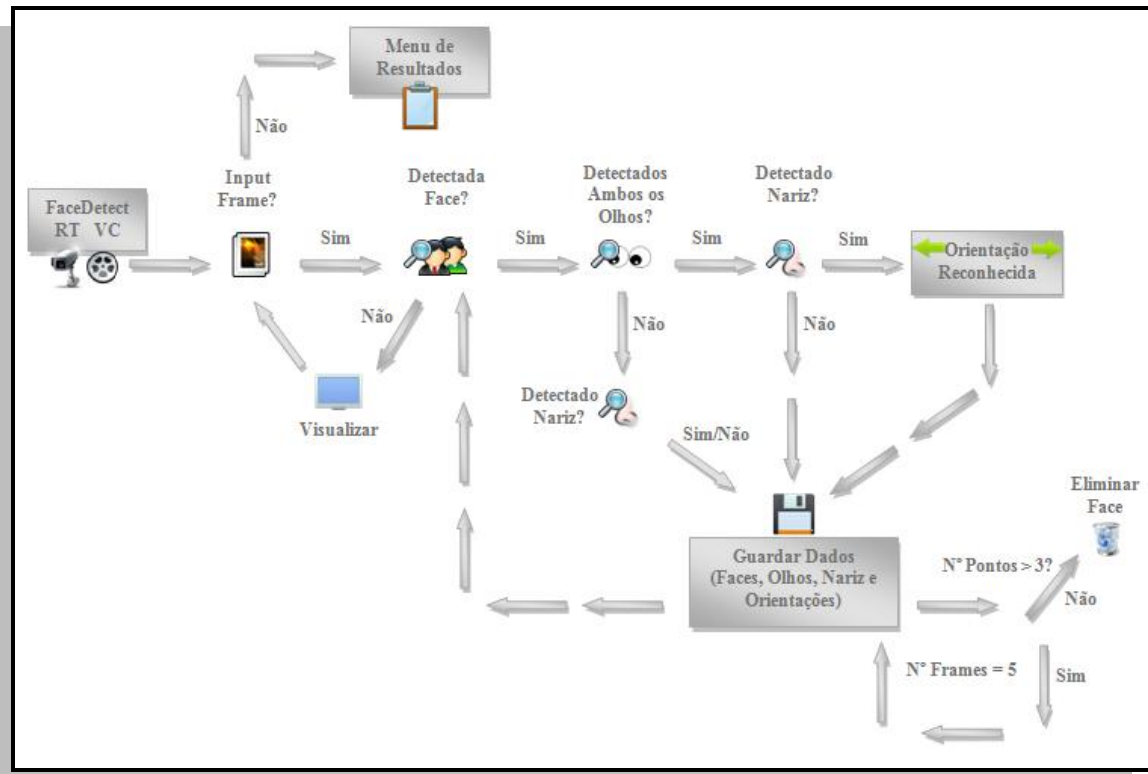


Figura 5-20 – Diagrama de Fluxo

Ambas as aplicações *FaceDetectRT* e *FaceDetectVC* iniciam com o envio de uma *frame* para análise. Essa *frame* começa por ser analisada de forma a determinar se existe alguma face presente. Em caso de sucesso, para cada face que é encontrada realiza-se uma pesquisa a fim de detectar a região dos olhos, seguida da pesquisa pela área do nariz. Se as detecções adquirirem os três pontos de referência necessários (ambos os olhos e nariz), então as aplicações iniciam a determinação da orientação da cabeça, ou face, dessa pessoa, terminando com o armazenamento de todos os dados adquiridos sobre a face, os olhos, o nariz e a respectiva orientação. Feito o armazenamento dos dados, segue-se a função de eliminação que vai verificar se a face adicionada cumpre os requisitos mínimos, isto é, se já foi adicionada há, pelo menos, mais de cinco *frames*, e se possui no mínimo 3 pontos ou nós de coordenadas. Após a execução da função de eliminação, o ciclo repete-se para a face seguinte até não conter mais faces. Quando isto acontece, a aplicação irá apresentar ao utilizador as detecções adquiridas na *frame* actual e volta a adquirir mais uma *frame* para análise, caso exista. Na ausência de mais *frames* para análise, a aplicação termina o processamento e apresenta o menu de resultados adquiridos no decorrer das detecções.

6. Teste e Avaliação

O teste e a avaliação das aplicações é uma acção muito importante, visto que procura eliminar as lacunas que possam estar presentes nas aplicações. Com a sua realização pretende-se também averiguar se as aplicações cumprem os objectivos principais que foram recolhidos inicialmente. Trata-se da fase do desenvolvimento em que são efectuados testes às detecções e ao desempenho de ambas as aplicações desenvolvidas.

Segundo Faulkner (1998), a avaliação deve ser encarada como um processo contínuo de Implementação – Avaliação – Implementação e deve ser feita, se possível, nas três fases seguintes:

- **Fase de análise do sistema** – se possível o programador deve analisar os trabalhos desenvolvidos em áreas semelhantes de forma a obter informação que facilite a sua criação.
- **Fase de implementação do sistema** – criação de um protótipo, funcional ou não, de forma a simular o sistema. Fazer correcções nesta fase é muito mais fácil do que fazer correcções com o sistema totalmente implementado.
- **Fase de pré-produção** – quando o protótipo está concluído, a avaliação pode ser feita em maior escala e concentrar-se nos detalhes do sistema.

6.1. Técnicas de avaliação

De acordo com Faulkner (1998), existem dois métodos de avaliação de um sistema. O primeiro método, designado por **método de avaliação analítica**, consiste numa avaliação formal com lápis e papel dos objectivos e tarefas a executar. O segundo método, conhecido como **método empírico**, consiste numa análise de desempenho do utilizador em relação à utilização do sistema, por meio de entrevistas, experiências, questionários ou mesmo através da observação dos utilizadores a realizar as tarefas.

No que diz respeito às **entrevistas**, estas podem ser realizadas em qualquer um dos métodos de avaliação descritos. Podem inclusivamente ser utilizadas como forma de recolher a informação inicial para o desenvolvimento do sistema e também durante o

seu desenvolvimento, com a participação activa dos utilizadores finais, de forma a obter o sistema desejado e não o sistema que os programadores pensam ser o melhor (Faulkner 1998).

As **experiências** podem ser de dois tipos, experiências comparativas, em que se compara o desempenho do novo sistema com um sistema já existente e experiências absolutas, onde o novo sistema é testado isoladamente.

Já os **questionários** podem fornecer informação que permite identificar problemas na concepção do sistema. Devem ser efectuados durante a fase de Implementação – Avaliação – Implementação, até os avaliadores terem a certeza que as respostas são as acertadas (Faulkner 1998).

Finalmente, a técnica de **observação** possibilita à equipa de desenvolvimento uma observação pormenorizada de um utilizador a efectuar uma determinada tarefa, mas é necessário ter cuidado visto que, muitas vezes, as pessoas ao serem observadas têm comportamentos diferentes daqueles que evidenciam na realidade.

6.2. Avaliação do Sistema

Para realizar a avaliação das aplicações desenvolvidas no âmbito do trabalho descrito nesta dissertação foi escolhido o método empírico, mediante a realização de uma experiência absoluta em que não se compara o novo sistema em relação a outro, mas se analisa e avalia o desempenho dos classificadores para as respectivas detecções, o armazenamento dos dados e as orientações reconhecidas correctamente.

Sendo a análise do desempenho dos classificadores *Haar* um processo complexo devido ao facto de ser necessário avaliar, entre outros parâmetros, as várias etapas dos classificadores, e as várias dimensões dos objectos a serem detectados, neste trabalho optou-se por recorrer a estudos mencionados na literatura que permitem aferir o grau de desempenho dos classificadores que foram utilizados no desenvolvimento das aplicações.

No que diz respeito à avaliação do armazenamento das faces e das orientações que foram correctamente obtidas, optou-se pela utilização de um ficheiro de vídeo guardado em disco cujas características principais se identificam na tabela 6-1.

Espaço Ocupado	Resolução	Taxa de <i>frames</i>	Duração do Vídeo	Número de <i>frames</i>
2.26 MB	768x576	20 FPS	12 Seg.	244 <i>frames</i>

Tabela 6-1 – Características Principais do Vídeo de Testes

O vídeo de teste apresenta um ambiente não controlado no qual é possível identificar as faces de pessoas, bem como as respectivas orientações de face durante a passagem num espaço físico.

6.3. Análise e discussão dos Resultados

Esta secção inicia-se com a análise dos dados referentes ao desempenho. Como foi referido anteriormente, o uso de câmaras de videovigilância podem disponibilizar aos seus proprietários informações que podem ser úteis para diferentes finalidades. De forma a verificar se as aplicações permitem obter dados precisos, foram estudados os classificadores para as detecções faciais, oculares e nasais, tendo-se igualmente submetido a aplicação *FaceDetectVC* a testes de armazenamento e determinação das orientações obtidas. Tendo sido feita a avaliação da aplicação *FaceDetectVC*, os resultados obtidos aplicam-se também para a avaliação da aplicação *FaceDetectRT* devido ao facto de ambas funcionarem da mesma forma, sendo a única diferença o facto da segunda funcionar em tempo real e a primeira em ficheiros de vídeo previamente armazenados em disco.

Assim, utilizando um vídeo com 244 *frames* tomou-se nota de todos os dados armazenados, orientações obtidas e o tempo de processamento de todas as *frames* contidas em tal vídeo de teste.

6.3.1. Classificadores *Haar*

De acordo com Castrillón-Santana et al. (2008), vários autores optaram por criar e tornar os seus próprios classificadores públicos desde a publicação do mais recente trabalho de Viola & Jones (2004) onde se descreve uma abordagem rápida de múltiplas fases para a detecção de objectos de forma geral, e também desde a publicação do trabalho de Lienhart & Maydt (2002) ter sido disponibilizado. Sendo assim, os vários classificadores necessitavam de ser testados ao nível do desempenho para que, no futuro, os criadores de *software* pudessem optar pela escolha dos classificadores mais

eficazes para as respectivas aplicações, tendo como base os resultados dos seus desempenhos. De forma a compreender e analisar o desempenho dos vários classificadores, Castrillón-Santana et al. (2008) optaram por utilizar o conjunto de dados da CMU¹⁹ (Schneiderman & Kanade 2000) para utilização experimental. Estes dados estão divididos em quatro categorias diferentes: *test-low*, *test*, *rotated* e *newtest*, sendo cada categoria uma combinação dos dados fornecidos em Sung & Tomaso Poggio (1998) e Rowley et al. (1998). O conjunto de dados é composto por várias imagens representadas em tons de cinzento que contêm faces diferentes com diversas orientações.

Assim, Castrillón-Santana et al. (2008) submeteram o ficheiro de classificadores *haarcascade_frontalface_alt2.xml* às quatro categorias diferentes de forma a testar o desempenho dos classificadores. A forma utilizada para testar o desempenho consistiu no cálculo da taxa de detecções obtidas por número de falsos positivos, tendo obtido os resultados ilustrados na tabela 6-2:

Categoria	Desempenho
<i>test-low</i>	89.07%
<i>test</i>	86.98%
<i>rotated</i>	19.28%
<i>newtest</i>	83.56%

Tabela 6-2 – Desempenho do *haarcascade_frontalface_alt2*

Conforme se pode observar na tabela 6-2, exceptuando a categoria *rotated*, que obteve um fraco desempenho devido ao facto dos classificadores não estarem devidamente treinados para a obtenção de faces com algumas rotações, os resultados obtidos nas restantes categorias foram bastante positivos no que diz respeito ao desempenho da detecção.

No que diz respeito às detecções oculares e nasais, os resultados referidos em Castrillón-Santana et al. (2008) nas detecções efectuadas após a pesquisa das faces encontram-se resumidos na tabela 6-3:

¹⁹ CMU – “Carnegie Mellon University”

Olho Esquerdo		Olho Direito		Nariz	
TD	FD	TD	FD	TD	FD
17.2%	18	15.1%	18	1.7%	0

Tabela 6-3 – Desempenhos Oculares e Nasais
(TD – Taxa de Detecções, FD – Falsas Detecções)

Os resultados ilustrados na tabela 6-3 dizem respeito aos classificadores que foram utilizados no desenvolvimento de ambas as aplicações referidas nesta dissertação. Assim, analisando os resultados das obtenções dos olhos verifica-se que, apesar de no olho esquerdo ter sido obtida um valor de 17.2% e no direito de 15.1%, estes valores são um pouco mais elevados nas aplicações aqui descritas devido ao facto de ambas as aplicações recorrerem aos dois ficheiros de classificadores, podendo um classificador colmatar as falhas do outro, conforme foi explicado acima.

Note-se ainda que, no caso do nariz, apesar de ter-se obtido uma taxa muito baixa de detecções, não ocorreram falsas detecções, o que permite fornecer uma informação precisa da localização do nariz sempre que é detectado.

6.3.2. Armazenamento das Detecções

Para determinar o grau de armazenamento das aplicações desenvolvidas optou-se por registar todas as falsas detecções que ocorreram e todas as eliminações que sucederam ao longo do processamento do vídeo escolhido para o efeito, através da análise visual do vídeo e dos dados armazenados nas listas, tendo-se obtido os resultados indicados na tabela 6-4.

Falsas Detecções	Falsas Detecções Eliminadas	Faces Eliminadas com Menos de 3 Frames
11	11	19

Tabela 6-4 – Resultados do Armazenamento dos Dados

Ao longo de todo o processamento identificaram-se e guardaram-se treze faces, sendo cada uma delas considerada como pertencente a uma pessoa diferente. Contudo, no vídeo de teste estavam apenas presentes sete pessoas. Isto sucede nos casos em que, apesar de já ter sido detectada e armazenada uma pessoa nas *frames* anteriores, a mesma

pessoa volta a ser armazenada como uma nova face. Esta situação ocorre sempre que uma pessoa deixa de ser detectada e só volta a ser detectada mais tarde, mas também pode ocorrer devido ao posicionamento da câmara, nos casos e que a câmara de videovigilância se encontra muito perto do nível do solo ou numa posição que permite uma grande aproximação de uma pessoa à câmara. Nestes casos, o armazenamento da mesma pessoa como uma nova face ocorre mais frequentemente, pois quanto mais próxima da câmara estiver a pessoa, maior será a diferença entre as coordenadas do ponto que foi armazenado dessa face nas *frames* seguintes.

6.3.3. Orientações Reconhecidas Correctamente

Na obtenção dos resultados relativamente às orientações foi necessário identificar todos os casos para os quais foi possível obter os três pontos de referência (dois olhos e nariz) e determinar se as orientações obtidas pelas aplicações correspondem de facto às orientações tomadas pelas pessoas no vídeo de teste. Obtiveram-se os resultados da tabela 6-5.

Orientações Reconhecidas	Orientações Correctas	Orientações Incorrectas
95	74	21

Tabela 6-5 – Resultados das Orientações

Os resultados da tabela 6-5 demonstraram que, tendo detectado os três pontos de referência, as aplicações conseguem determinar as orientações de uma pessoa com uma taxa de sucesso de 77.9%. Esta taxa de sucesso poderá vir a aumentar se as detecções efectuadas pelos classificadores para os olhos e para o nariz também aumentarem, já que, nos casos onde ocorreram orientações incorrectas, verificou-se que sucederam devido a uma falha na detecção do olho esquerdo ou do olho direito. A referida falha diz respeito a situações em que um dos olhos fica um pouco fora da zona correcta do olho, fazendo com que o respectivo olho tenha uma maior proximidade com o nariz do que o olho oposto, dando assim uma orientação incorrecta para essa pessoa, tal como se ilustra na figura 6-1.



Figura 6-1 – Exemplo de Uma Orientação Incorrecta

Na figura 6-1 é possível visualizar que, embora todas as detecções tenham sido feitas com sucesso, por vezes essas detecções obtêm ligeiras falhas nas suas posições relativamente à posição real dos olhos da pessoa. Neste caso, verifica-se que a face correspondente se encontra com uma orientação para a esquerda. Contudo, as aplicações determinam que ocorreu uma orientação para a direita devido ao facto da análise dos dados ter indicado que o olho esquerdo se encontra a uma distância maior do nariz do que o olho direito, resultando pois uma orientação detectada incorrectamente.

6.3.4. Tempo de Processamento

Para averiguar o tempo do processamento que a aplicação *FaceDetectVC* toma a efectuar a análise do vídeo até à última fase em que apresenta o menu de resultados, submeteu-se, em primeiro lugar, o vídeo de teste referido anteriormente a vários *frame rates*. Para a segunda avaliação manteve-se o *frame rate* mas alterou-se a resolução do vídeo. Deste modo, os resultados obtidos ilustram-se na tabela 6-6:

FPS	15 FPS	20FPS	23FPS	25 FPS	30 FPS
Tempo de Execução	10 Seg.	12 Seg.	14 Seg.	16 Seg.	19 Seg.
Faces Encontradas	12 Faces	13 Faces	15 Faces	17 Faces	18 Faces
Nº Total de Detecções Guardadas	194	274	359	367	450

Tabela 6-6 – Resultados do Vídeo de Testes com Diferentes Taxas de *Frames* Por Segundo

Como seria de esperar, os resultados da tabela 6-6 permitem verificar que à medida que aumenta o *frame rate*, aumenta igualmente o tempo tomado para analisar o vídeo completo. Em contrapartida, obtém-se um maior número de detecções. O valor do tempo de execução indicado na tabela 6-6 diz respeito ao tempo que a aplicação demora

a analisar o número de *frames* igual ao valor do *frame rate*. Para o caso de um vídeo de 15 fps, a aplicação demora a cerca de 10 segundos analisar essas 15 *frames*.

No sentido de determinar se a resolução de um vídeo iria afectar as detecções efectuou-se ainda uma análise com o vídeo de teste modificado para várias resoluções diferentes. O vídeo contém 20 *frames* por segundo e a sua resolução original é de 768x576. Os resultados obtidos para diferentes resoluções estão apresentados na tabela 6-7.

Resolução	352x240	352x288	720x480	720x576	768x576 (original)
Tempo de Execução	2.6 Seg.	3.1 Seg.	10 Seg.	11 Seg.	12 Seg.
Faces Encontradas	5 Faces	5 Faces	12 Faces	10 Faces	13 Faces
Nº Total de Detecções Guardadas	73	99	271	235	274

Tabela 6-7 – Resultados do Vídeo de Testes com Diferentes Resoluções

Os resultados ilustrados na tabela 6-7 confirmam que a resolução de um vídeo afecta a velocidade do processamento das *frames* e também as respectivas detecções. Neste caso, verificou-se que resolução de 720x480 foi a que menos afectou o desempenho das detecções e obteve um menor tempo de processamento em relação ao vídeo original. Ainda na tabela 6-7 é possível identificar que as detecções são mais afectadas se a resolução na vertical e na horizontal da imagem não forem proporcionais. De facto, os valores da resolução na vertical e na horizontal devem manter, tanto quanto possível, a imagem próxima da realidade. Por exemplo no caso da resolução 720x576 a detecção foi inferior às detecções com a resolução 720x480 devido ao facto da imagem ter ficado ligeiramente “esticada”, afastando-se bastante da imagem real.

Finalmente, a aplicação *FaceDetectRT* foi testada com uma simples *webcam* com aproximadamente 1.3 megapixéis, uma resolução de 352x240 e um *frame rate* de 15 fps. Tendo sido já analisados os aspectos relativos ao armazenamento dos dados e às orientações correctas e incorrectas da primeira aplicação, optou-se por medir apenas a velocidade, ou o atraso, que a aplicação obtém durante as detecções feitas em tempo real. Esses resultados foram bastante positivos tendo-se obtido um atraso que vaiou apenas entre 0.9 e 1.4 segundos.

7. Conclusão

Neste capítulo são apresentadas as conclusões a que se chegou com o desenvolvimento desta dissertação e os possíveis melhoramentos futuros.

O trabalho aqui descrito procurou demonstrar o interesse e a utilidade que as tecnologias de reconhecimento humano podem ter ao serem utilizadas sobre vídeo capturado a partir de câmaras de vídeo de baixo custo, no sentido de possibilitar a obtenção de informações úteis referentes a locais designados como ambientes não controlados, de forma a facilitar futuras tomadas de decisão. Esta perspectiva implicou a criação de duas aplicações e a realização de um estudo que pretendeu identificar o método de reconhecimento de faces humanas mais adequado.

7.1. Considerações Finais

O objectivo inicial de tentar criar aplicações para efectuar a detecção e o seguimento automático de pessoas em ambientes não controlados foi atingido. A recolha e análise dos resultados dos testes efectuados possibilitaram a recolha de informação importante, que contribui para determinar a precisão e o desempenho das aplicações na obtenção dos dados sobre as pessoas presentes no ambiente que as câmaras de vídeo abrangem.

Verificou-se que a introdução das aplicações veio contribuir para uma recolha adicional de informação por parte de câmaras de vídeo de baixo custo, sem requerer a interacção de um ser humano. Essa recolha de dados contribui com vários aspectos, desde a verificação das orientações mais tomadas pelas pessoas até às respectivas deslocações no ambiente físico. Permite ainda determinar com um certo grau de precisão quantas pessoas foram identificadas ao longo de todo o processo de análise, permitindo assim compreender as áreas físicas que despertam maior interesse nas pessoas.

No que diz respeito à precisão das aplicações, efectuamos um balanço igualmente positivo pois, apesar do tempo de processamento ser um pouco elevado, as aplicações detectaram deslocações correctas das pessoas nos ambientes não controlados, o que significa que essas pessoas foram detectadas na face correcta, evitando as falsas

detecções que ocorreram ao longo do processamento, obtendo-se uma taxa de orientações correctas de aproximadamente 78%.

Este trabalho contribuiu ainda para rever os métodos e as bibliotecas de Visão Computacional mais utilizados actualmente no reconhecimento humano, e especificamente no reconhecimento de faces humanas, juntamente com as respectivas características. Permitiu ainda seleccionar o método mais indicado para os objectivos das aplicações desenvolvidas.

7.2. Limitações

A maior limitação deste trabalho consistiu no facto de não ter sido possível testar as aplicações numa experiência realizada num ambiente não controlado real, tendo-se apenas testado um ficheiro de vídeo que simula um ambiente não controlado. No entanto, os resultados obtidos podem ser utilizados como indicações úteis sobre a utilidade destas tecnologias, pois o vídeo de teste contém todos os requisitos necessários para o efeito, desde um espaço físico onde várias pessoas se deslocam, contendo igualmente áreas que despertam mais interesse a essas pessoas de forma a analisar as respectivas orientações.

Uma das limitações das aplicações descritas neste trabalho tem a ver com o facto de nos casos para os quais não é possível obter os três pontos de referência (olhos e nariz) de uma face não ser possível determinar a orientação de uma dada pessoa. Destaca-se igualmente a dependência que as orientações possuem dos resultados obtidos pelos classificadores na obtenção das posições de ambos os olhos e do nariz: caso as posições estejam incorrectas, provocam a detecção de uma orientação incorrecta.

Outra das limitações diz respeito ao facto de algumas faces serem eliminadas por falta de coordenadas e, mais tarde, obterem novas coordenadas, acabando por se perder no máximo três coordenadas dessa face. Destaca-se ainda o facto de uma face poder ser armazenada com IDs diferentes. Isto sucede porque as pessoas deixaram de aparecer durante dez *frames* consecutivas ou porque, quando surgiram, fizeram-no longe da posição que correspondia à sua face.

Uma última limitação está relacionada com o tempo que demora a processar toda a informação nos casos para os quais os vídeos têm durações muito prolongadas e

também a aplicação *FaceDetectVC* possui apenas compatibilidade para vídeos cujo formato é AVI²⁰.

7.3. Trabalho Futuro

Como trabalho futuro, prevê-se para já a criação de uma variável de ambiente de forma a facilitar a execução ou a selecção de um vídeo de ambas as aplicações.

Outras possibilidades de desenvolvimento futuro incluem a criação dos próprios classificadores de forma a aumentar a precisão das detecções. Para além disso, pretendemos ainda explorar outros tipos de situações como por exemplo utilizar as aplicações para ambientes distintos. É ainda necessário conjugar os classificadores do olho esquerdo com os classificadores do olho direito e tentar aumentar a rapidez de processamento das *frames* sem comprometer o desempenho das detecções.

²⁰ AVI (Audio Video Interleave) é um formato de vídeo que possui áudio e vídeo sincronizados.

Bibliografia

- Benezeth, Y. et al., 2009. Vision-Based System for Human Detection and Tracking in Indoor Environment. *International Journal of Social Robotics*, 2(1), pp.41-52.
- Bradski, G. & Kaehler, A., 2008. *Learning OpenCV: Computer vision with the OpenCV library*, O'Reilly Media.
- Castrillón-Santana, M. et al., 2008. Face and facial feature detection evaluation. *Third International Conference on Computer Vision Theory and Applications, VISAPP08*, (1).
- Faulkner, C., 1998. *The essence of human-computer interaction*, Prentice Hall.
- Forsyth, D.A. & Ponce, J., 2002. *Computer vision: a modern approach*, Prentice Hall Professional Technical Reference.
- Freund, Y. & Robert, E.S., 1997. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *Journal of Computer and System Sciences*, 55(1), pp.119-139.
- Harvey, A., 2010. OpenCV Face Detection: Visualized. Disponível em: <http://ahprojects.com/c/itp/thesis> [Consultado em 14 Setembro 2010].
- Kollreider, K. et al., 2007. Real-Time Face Detection and Motion Analysis With Application in "Liveness" Assessment. *IEEE Transactions on Information Forensics and Security*, 2(3), pp.548-558.
- Lienhart, R. & Maydt, J., 2002. An extended set of Haar-like features for rapid object detection. *Proceedings. International Conference on Image Processing*, p.I-900-I-903.
- Papageorgiou, C.P., Oren, M. & Poggio, T., 1998. A general framework for object detection. *International Conference on Computer Vision*, pp.555-562.
- Pinho, R. & Manuel, R., 2005. Equalização e modificação de histogramas.
- Rowley, H.A., Baluja, S. & Kanade, T., 1998. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1), pp.23-38.
- Schneiderman, H. & Kanade, T., 2000. A statistical method for 3D object detection applied to faces and cars. *Proceedings IEEE Conference on Computer Vision and Pattern Recognition CVPR 2000 Cat NoPR00662*, 1, pp.746-751.

Sung, K.-K. & Poggio, Tomaso, 1998. Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1), pp.39-51.

Vasconcelos, J. & Carvalho, J., 2005. *Algoritmia e Estruturas de Dados*, Centro Atlântico.

Viola, P. & Jones, M., 2001. Rapid object detection using a boosted cascade of simple features. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

Viola, P. & Jones, M., 2004. Robust Real-Time Face Detection. *International Journal of Computer Vision*, 57(2), pp.137-154.