Luís Felipe Moreira da Silva Cassales
Rafael Leme Costa

Reconhecimento de Face

UNIVERSIDADE DE SÃO PAULO
Escola Politécnica
2013

Luís Felipe Moreira da Silva Cassales
Rafael Leme Costa

Reconhecimento de Face

Orientador: Prof. Dr. Marcos Ribeiro Pereira Barretto

UNIVERSIDADE DE SÃO PAULO
Escola Politécnica
2013

FICHA CATALOGRÁFICA

Cassales, Luís Felipe da Silva
Reconhecimento de face / L.F.S. Cassales, R.L. Costa. -- São Paulo, 2013.
58 p.

Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos.

1.Autofaces 2.JADE 3.OpenCV 4.Reconhecimento de face 5.Robô sociável I.Costa, Rafael Leme II.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos III.t.

A nossos familiares e amigos.

Agradecimentos

Em primeiro lugar, gostaríamos de agradecer a nosso orientador Marcos Barretto, que nos guiou e nos auxiliou a alcançar nossos objetivos.

Gostaríamos de agradecer também a nossos professores de PSI-2651 Processamento, Síntese e Análise de Imagens I: Hae Yong Kim e Roseli de Deus Lopes, que nos auxiliaram no aprendizado da biblioteca OpenCV e no desenvolvimento do trabalho.

Por último, mas tão importante quanto, gostaríamos de agradecer a nossos familiares e amigos, que nos apoiaram e nos compreenderam ao longo dessa longa e dura jornada.

"A percepção do desconhecido
é a mais fascinante das experiências.
O homem que não tem os olhos abertos
para o misterioso passará pela vida sem ver nada."
(Albert Einstein)

Resumo

Este trabalho consiste no desenvolvimento de um sistema de reconhecimento de faces por câmera, utilizando o conceito multi-agentes, com aplicação no robô "Minerva". O trabalho se insere no contexto de robôs sociáveis, que devem ser capazes de interagir com os seres humanos de maneira natural, comunicando-se com eles e compreendendo-os. O reconhecimento de faces é importante por se tratar de uma técnica segura e confiável que permite ao robô o reconhecimento do interlocutor com quem interage. A tarefa de reconhecimento de faces é realizada em duas etapas, sendo a primeira a da detecção da face e a segunda a do reconhecimento propriamente dito. Para a primeira etapa, utiliza-se o método proposto por Viola e Jones (2004), que consiste em utilizar grupos de pixels em formatos retangulares para detectar padrões claros e escuros com base na variação do valor do contraste desses grupos, visando detectar características presentes numa face humana, como por exemplo, a simetria entre os olhos e a região mais escura ao redor deles se comparada com as demais regiões da face. Já para o reconhecimento, utiliza-se a aplicação em cascata do algoritmo Discrete Cosine Transform (DCT) e o algoritmo Principal Component Analysis (PCA), que gera um subespaço com menor covariância através do cálculo da distância euclidiana entre a imagem do rosto da pessoa que se deseja verificar e a imagem média das faces de treinamento obtida pelo algoritmo de aprendizagem de máquina na fase de treinamento. Para a implementação dos algoritmos, utiliza-se o JavaCV, que é uma implementação na linguagem Java da biblioteca de visão computacional OpenCV, originalmente implementada na linguagem C++. É proposta uma arquitetura baseada em Java Agent Development Framework (JADE), que introduz as vantagens da modularização em agentes e da possibilidade de intercambiabilidade. O sistema conta com quatro agentes: *ImageAgent*, *DetectionAgent*, *RecognitionAgent* e *MonitorAgent*. O *ImageAgent* captura uma frame da câmera e avisa o *DetectionAgent*, que realiza a detecção e a envia ao *MonitorAgent* e ao *RecognitionAgent*. Este último realiza o reconhecimento e envia o resultado ao *MonitorAgent*, que é responsável por montar a imagem final e imprimir na tela. Esses agentes estão integrados a três tipos de memórias: a Memória Sensorial, que é responsável pelo contato do sistema com o ambiente externo, a Memória Episódica, que é responsável por manter uma coleção dos eventos ocorridos e a Memória Semântica, que armazena os dados de sujeitos conhecidos. São apresentados testes com fotos, que utilizam The Database of Faces - AT&T Laboratories Cambridge, testes com vídeos, que utilizam o Honda/UCSD Video Database e testes de execução, que utilizam a arquitetura proposta baseada em agentes. Conclui-se que o sistema se mostra bastante eficiente nos testes com fotos e nas detecções em vídeo e se mostra satisfatório no reconhecimento em vídeo, dada em conta sua aplicação. Demonstra-se também que o sistema pode ser utilizado em tempo real, para aplicação em robôs sociáveis.

Palavras-chaves: autofaces. JADE. OpenCV. reconhecimento de face. robô sociável.

Abstract

This work consists in developing a face recognition system by video camera, using the multi-agent concept, with application on "Minerva" robot. The work is inset in the social robots context, which should be capable of interacting with the human beings in a natural way, communicating with them and understanding them. The face recognition is important because it is a safe and reliable technique that allows the robot to recognize the interlocutor with whom it interacts. The task of face recognition is made in two stages, the first one being the face detection and the second one the recognition itself. For the first stage, it is used the method proposed by Viola and Jones (2004), which consists in using pixel groups in rectangular shapes to detect light and dark patterns based on the contrast value variation, to detect features present in a human face, such as, the symmetry between the eyes and the darker region around them if compared to the other regions of the face. For the recognition, it is used the cascade application of the algorithm Discrete Cosine Transform (DCT) and the algorithm Principal Component Analysis (PCA), that generates a subspace with less covariance through the calculus of the euclidian distance between the image of the person's face to be verified and the average image of the training faces obtained by the machine learning algorithm in the training phase. To the algorithm implementation, it is used the JavaCV, that is an implementation in Java language of the computer vision library OpenCV, originally written in C++ language. It is proposed an architecture based in Java Agent Development Framework (JADE), that introduces the advantages of the modularization in agents and the possibility of interchangeability. The system has four agents: *ImageAgent*, *DetectionAgent*, *RecognitionAgent* e *MonitorAgent*. The *ImageAgent* captures a frame from the camera and warns the *DetectionAgent*, that performs the detection and sends it to *MonitorAgent* and to *RecognitionAgent*. The last one performs the recognition and sends the result to *MonitorAgent*, that is responsible to mount the final image and to print it on the screen. These agents are integrated to three kinds of memories: the Sensory Memory, that is responsible to make contact between the system and the external environment, the Episodic Memory, that is responsible to keep a collection of the occurred events and the Semantic Memory, that stores data of the known subjects. It is presented photo tests, that use The Database of Faces - AT&T Laboratories Cambridge, video tests, that use the Honda/UCSD Video Database and execution testes, that use the proposed architecture based in agents. It is concluded that the system proves to be very efficient in the photo tests and in the video detections and proves to be satisfactory in the video recognition, given its application. It is also shown that the system can be used for real time application in social robots.

Keywords: eigenfaces. JADE. OpenCV. face recognition. social robot.

Lista de ilustrações

Figura 1 – Aplicações das características de Haar na detecção de face de um dos autores 18
Figura 2 – Imagem de uma face (Extraído de: The Database of Faces - AT&T Laboratories Cambridge) antes (a) e após a aplicação do DCT (b) . . . 19
Figura 3 – Imagem da face da Figura 2a após reconstrução com blocos 8x8 (a), 16x16 (b) e 32x32 (c) 20

Figura 4 – Visão geral da arquitetura do sistema 31
Figura 5 – Visão detalhada da estrutura de Memória do sistema 32
Figura 6 – As quatro principais características de Haar 33

Figura 7 – Sujeitos 1, 2 e 4 (Extraído de: The Database of Faces - AT&T Laboratories Cambridge) utilizados em T1 41
Figura 8 – Sujeitos 7, 9, 11 e 23 (Extraído de: The Database of Faces - AT&T Laboratories Cambridge) utilizados em SH 41
Figura 9 – Sujeitos 19, 20 e 38 (Extraído de: The Database of Faces - AT&T Laboratories Cambridge) utilizados em SH 42
Figura 10 –– Sujeitos 4, 19 e 20 e sujeitos 7, 13 e 17 (Extraído de: The Database of Faces - AT&T Laboratories Cambridge) utilizados em CI-OC 42
Figura 11 –Sujeitos 8, 11 e 24 (Extraído de: The Database of Faces - AT&T Laboratories Cambridge) utilizados em CI-OF 42
Figura 12 –Sujeitos 12, 38, 40, 35, 25, 29, 4, 9 e 23 (Extraído de: The Database of Faces - AT&T Laboratories Cambridge) utilizados em CI-CR 42
Figura 13 –Imagem média das três faces, obtida após aplicação do PCA (a) e Imagens projetadas no subespaço após aplicação do PCA – eigenfaces (b) 43
Figura 14 –Imagem média das três faces, obtida após aplicação do DCT e do PCA (a) e Imagens projetadas no subespaço após aplicação do DCT e do PCA – eigenfaces (b) 43
Figura 15 –Imagens (Extraído de: The Database of Faces - AT&T Laboratories Cambridge) identificadas erroneamente pela solução implementada . . 44
Figura 16 –Imagens dos sujeitos 1, 2, 3 e 4 do Honda/UCSD Video Database, após detecção e redimensionamento, no formato padrão do sistema 46
Figura 17 –Imagens de falsos positivos detectados em vídeos de treino do Honda/UCSD Video Database 47

Figura 18 – Detector de faces através de webcam detectando o rosto dos autores, presença de óculos 48
Figura 19 – Detector de faces através de webcam detectando o rosto de um dos autores, cabeça inclinada 48
Figura 20 – Detector de faces através de webcam detectando o rosto de um dos autores, cabeça rotacionada (a) e olhos fechados (b) 49
Figura 21 – Reconhecimento Correto do Sujeito 1 - Fantine no teste com vídeos do Honda/UCSD Video Database 51
Figura 22 – Reconhecimento Correto do Sujeito 2 - Geoff no teste com vídeos do Honda/UCSD Video Database 51
Figura 23 – Reconhecimento Incorreto do Sujeito 13 - Satya no teste com vídeos do Honda/UCSD Video Database 52
Figura 24 – Reconhecimento Incorreto do Sujeito 14 - Vincent no teste com vídeos do Honda/UCSD Video Database 52
Figura 25 – Detecção Incorreta em frame do Sujeito 9 - Melody no teste com vídeos do Honda/UCSD Video Database 53
Figura 26 – Detecção Incorreta em frame do Sujeito 1 - Fantine no teste com vídeos do Honda/UCSD Video Database 53

Lista de tabelas

Tabela 1 – Comparativo entre as principais referências bibliográficas 25
Tabela 2 – Diferenças entre banco de dados relacional e NoSQL, extraída de (BRITO, 2010) 29

Tabela 3 – Comparativo entre os testes realizados 44
Tabela 4 – Contagem de Detecções Corretas e Falsos Positivos no teste com vídeos do Honda/UCSD Video Database 47
Tabela 5 – Contagem de Reconhecimentos Corretos e Reconhecimentos Incorretos no teste com vídeos do Honda/UCSD Video Database 50

Lista de abreviaturas e siglas

ACID	Atomicidade, Consistência, Isolamento e Durabilidade
AdaBoost	Adaptive Boosting
DCT	Discrete Cosine Transform
FIPA	Foundation for Intelligent Physical Agents
IDCT	Inverse Discrete Cosine Transform
JADE	Java Agent Development Framework
PCA	Principal Component Analysis
SIFT	Scale Invariant Feature Transform

Sumário

Lista de ilustrações **8**

Lista de tabelas **10**

1 Introdução **14**
1.1 Motivação 14
1.2 Objetivos 15
1.3 Estrutura 16

2 Revisão Bibliográfica **17**
2.1 Conceitos teóricos 17
2.2 Estado da Arte em detecção e reconhecimento de faces 23
2.3 Java Agent Development Framework (JADE) 26
2.4 Banco de Dados Orientado a Grafos 27

3 Sistema Proposto **31**
3.1 Visão Geral 31
3.2 Algoritmos Utilizados 33
3.2.1 Detecção de Face 33
3.2.2 Tracking de Face 33
3.2.3 Reconhecimento de Face 33
3.3 Agentes 34
3.3.1 Agente Imagem (*ImageAgent*) 34
3.3.2 Agente Detecção (*DetectionAgent*) 34
3.3.3 Agente Reconhecimento (*RecognitionAgent*) 35
3.3.4 Agente Monitor (*MonitorAgent*) 35
3.3.5 Regras FIPA no contexto do sistema 35
3.4 Memória 36
3.4.1 Memória Sensorial 36
3.4.2 Memória Episódica 36
3.4.3 Memória Semântica 37
3.5 Aspectos de Implementação 38

4 Resultados **40**
4.1 Testes com Fotos 40
4.1.1 Reconhecimento de face 40
4.1.1.1 Procedimentos experimentais 40

4.1.1.2 Resultados obtidos 43
4.2 Testes com Vídeo 45
4.2.1 Detecção de face 45
4.2.1.1 Procedimentos experimentais 45
4.2.1.2 Resultados obtidos 46
4.2.1.3 Testes Próprios 48
4.2.2 Reconhecimento de face 49
4.2.2.1 Procedimentos experimentais 49
4.2.2.2 Resultados obtidos 49
4.3 Execução em Tempo Real 54
4.3.1 Procedimentos experimentais 54
4.3.2 Resultados obtidos 54

5 Conclusão 55

Referências 56

1 Introdução

1.1 Motivação

Uma das principais motivações do trabalho é desenvolver um software para utilização no robô sociável "Minerva", com o intuito de fazer a identificação de seus usuários.

Robôs sociáveis devem ser capazes de interagir com o interlocutor, identificando-o e até mesmo sendo capaz de compreendê-lo, podendo assim interagir de forma diferenciada, específica para cada um de seus usuários. Existem diversos métodos que podem ser utilizadas para identificação, verificação e validação de usuários, sendo os mais utilizados:

• Reconhecimento por voz: método intrusivo, em que muitas vezes obriga o usuário a repetir uma palavra ou frase específica. Torna-se falho quando a voz do usuário apresenta rouquidão, situação bastante comum quando uma pessoa está gripada, por exemplo. Além disso, é bastante suscetível a ruídos provenientes do meio externo;

• Reconhecimento de íris: método também intrusivo. Torna-se falho em situações em que o usuário está utilizando óculos escuros e/ou lentes de contato;

• Reconhecimento de impressões digitais: método intrusivo. É fácil de ser fraudado com a utilização de próteses de silicone e torna-se falho se a pessoa estiver com o dedo sujo ou ferido. Além disso, grande parte dos leitores de impressão digital são produzidos baseando-se na utilização de sensores capacitivos, que são muito suscetíveis à variação do campo elétrico produzida pela umidade. Assim sendo, se o dedo pessoa estiver muito molhado, a imagem obtida da impressão digital estará bastante escura, e no caso de o dedo estar muito seco, a imagem obtida será muito clara ou opaca (MORAES, 2006).

• Reconhecimento de face: método não intrusivo. A face reúne uma grande quantidade de características únicas de cada pessoa (distância entre o centro dos olhos, tons mais escuros ao redor dos olhos, comparados com a região do nariz, simetria da face, etc), assim torna-se o método mais confiável, pois torna-se muito difícil reproduzir com grande semelhança todas as características reunidas na face. É, porém, um método de implementação mais complexa.

No contexto de robôs sociáveis, os métodos que poderiam ser utilizados são o reconhecimento por voz e o reconhecimento de face. Os outros métodos citados não tem sentido nessa aplicação por serem dependentes de sensores específicos, que tornariam o contato com o robô muito artificial. O reconhecimento por voz poderia ser utilizado, porém é muito mais suscetível a ruídos. Logo, o método mais adequado no contexto de robôs sociáveis é o de reconhecimento facial, pois trata-se de uma técnica segura e confiável, permitindo ao robô fazer o reconhecimento de cada usuário. Além disso, no caso de reconhecimento facial a partir de câmeras, pode-se implementar um sistema complemen-

tar de tracking, que permitiria ao robô olhar diretamente para a face de seu interlocutor, possibilitando assim uma conversa face a face. Tal sistema permitiria ao robô realizar um diálogo mais semelhante ao observado entre dois seres humanos, o que configuraria uma nova evolução no tema.

Algumas dificuldades fazem-se presentes quando do desenvolvimento de um sistema de reconhecimento facial. A face pode apresentar características instáveis, ou seja que não estão presentes o tempo todo nos humanos, como por exemplo, a presença de barba e bigode, ou ainda, a utilização de óculos (FAN et al., 2012) ou a presença de novas cicatrizes que não existiam quando as primeiras imagens foram obtidas e armazenadas pelo sistema. Pode-se estabelecer um intervalo de confiança para considerar que a pessoa a ser reconhecida é a mesma da imagem de face cadastrada no banco de dados. Por exemplo, se a imagem da face a ser reconhecida possuir 85% de identificação com uma imagem de face contida no banco de dados, então ela é aceita como a pessoa cadastrada. Contudo, nem sempre esse método funciona e, em determinadas situações, o reconhecimento poderá obter um falso positivo (quando o sistema reconhece uma pessoa como sendo outra, ou ainda quando uma pessoa que não está cadastrada no banco de dados é reconhecida como uma pessoa cadastrada) ou um falso negativo (quando o sistema não reconhece uma pessoa cadastrada no banco de dados).

Outro tipo de dificuldade encontrada é a pouca iluminação (FAN et al., 2012). Ambientes mal iluminados tornam difícil o reconhecimento de face, uma vez que a qualidade da imagem obtida torna-se pior e a variação de cores da face torna-se imperceptível. O uso do sistema em ambientes com iluminação controlada, junto a métodos de processamento de imagens para aumento de brilho e contraste resolvem o problema. Para a aplicação em robôs sociáveis, é fácil eliminar esse problema: confinando-se o robô a ser utilizado em um ambiente específico, com iluminação controlada.

Poses ou imagens da face obtidas lateralmente também geram problemas para se realizar o reconhecimento de face (XUAN; NITSUWAT, 2007). Problemas como esses só podem ser resolvidos com a utilização de múltiplas câmeras posicionadas adequadamente ou ainda utilizando-se diferentes tipos de imagem, como imagens térmicas, por exemplo (WU et al., 2012). Assim sendo, essas dificuldades serão as mais presentes para a aplicação em robôs sociáveis e dificilmente serão tratadas.

1.2 Objetivos

O objetivo do trabalho é implementar uma solução com a aplicação em robôs sociáveis, que sejam capazes de reconhecer seus usuários. Como objetivos secundários, pode-se estabelecer o desenvolvimento dessa solução a partir de módulos que possam ser distribuídos em diferentes máquinas, através de sistemas multi-agentes. Com isso, torna-se fácil a intercambiabilidade das funções existentes e a adição de novas funções, pela criação

de novos agentes, não sendo necessário alterar o código das funções já implementadas.

Trabalhos mais desafiadores poderiam ser traçados com a implementação de uma solução que consiga detectar relações que possam existir entre os indivíduos identificados, bem como prever possíveis atitudes desses indivíduos. Tal implementação consistiria no desenvolvimento de funções da memória semântica que visem armazenar o contexto em que a pessoa foi reconhecida (lugar, data, pessoas presentes, situação em que as pessoas se encontravam). Com isso, poderia ser previsto com quem a pessoa se reúne em determinado horário, em que lugar, em que ocasião, permitindo ao sistema prever possíveis atitudes das pessoas reconhecidas e identificar quando ela está agindo diferente do usual.

1.3 Estrutura

As seções seguintes estão divididas em 4 capítulos: Revisão Bibliográfica, Sistema, Resultados e Conclusão.

No capítulo Revisão Bibliográfica, apresenta-se o Estado da Arte do trabalho e os Conceitos Teóricos envolvidos.

No capítulo Sistema, apresenta-se uma descrição geral do sistema, dos agentes e das memórias e a proposta implementada.

No capítulo Resultados, descreve-se os procedimentos experimentais de cada teste e os resultados obtidos.

No capítulo Conclusão, apresenta-se a conclusão final do trabalho.

2 Revisão Bibliográfica

2.1 Conceitos teóricos

A seguir, são apresentados os conceitos teóricos mais relevantes ao trabalho, importantes para a compreensão do estado da arte. Alguns conceitos também visam esclarecer as regras adotadas para que seja possível o bom funcionamento da plataforma multi-agentes utilizada, uma vez que tais sistemas ainda não foram muito difundidos.

a) Classificadores de Haar em Cascata

O núcleo base para a detecção de objetos baseada em Classificadores de Haar são as características de Haar. Essas características são definidas por grupos de dois ou três retângulos adjacentes de pixels que se utilizam da mudança de valores de contraste verificada em cada retângulo para determinar regiões claras e escuras.(WILSON; FERNANDEZ, 2006)

Características retangulares simples de uma imagem podem ser obtidas utilizando-se uma representação chamada Imagem Integral. Uma Imagem Integral consiste em um vetor contendo a soma dos valores da intensidade dos pixels contidos na região à esquerda e superior à posição do pixel de posição (x,y) para o qual se deseja calcular a Imagem Integral. Assim, seja uma imagem descrita por I[x,y], sua Imagem Integral definida por II[x,y] é obtida computacionalmente pela equação 2.1:

$$II[x,y] = \sum_{x' \leq x, y' \leq y} I(x', y') \tag{2.1}$$

Algumas características de Haar também podem ser utilizadas rotacionadas de ângulos de $45^{\underline{o}}$. Nesses caso, o cálculo de sua Imagem Integral definida por IIR[x,y] pode ser obtido por meio da equação 2.2:

$$IIR[x,y] = \sum_{x' \leq x, x' \leq x-|y-y'|} I(x', y') \tag{2.2}$$

Computacionalmente, o cálculo da Imagem Integral é feito de maneira bastante rápida, com o uso de pouca memória, por isso tem sido bastante utilizada para o cálculo das características Haar para detecção de faces em vídeo, em tempo real. Pode-se utilizar diversas características de Haar em cascata para maximizar o processo de detecção de uma face. Essas características visam detectar a simetria entre os olhos, o maior contraste da região dos olhos em comparação com a região das bochechas, detectar a região mais

clara no nariz em comparação com a região das bochechas, etc. A Figura 1 demonstra a utilização das características de Haar para detectar uma face.

Figura 1 – Aplicações das características de Haar na detecção de face de um dos autores

O processo de detecção de faces utilizando características de Haar pode ser maximizado, utilizando-se uma associação desses classificadores em cascata, processo conhecido como classificadores de Haar em cascata. O processo consiste em obter o valor da primeira característica de Haar por meio de do cálculo da Imagem Integral. Se o resultado for acima ou abaixo de um determinado valor estabelecido para a presença de face, calcula-se o valor da próxima característica de Haar e assim sucessivamente até o fim das características de Haar. Ao passar por todas as características, tem-se então que existe uma face no trecho de imagem analisado. Caso o valor calculado não satisfaça aos critérios estabelecidos para a presença de face, já descarta-se a imagem, não sendo necessário o cálculo da característica de Haar seguinte, com isso o processo torna-se mais rápido. Esse processo pode ainda ser otimizado com a implementação de aprendizagem de máquina com o algoritmo Adaptive Boosting (AdaBoost)(VIOLA; JONES, 2004), explicado na seção seguinte.

b) Adaptive Boosting (AdaBoost)

Segundo Rojas (2009), o algoritmo Adaptive Boosting (AdaBoost) é usado para gerar classificadores fortes a partir de classificadores fracos. Segundo Athitsos (2012), um classificador é definido pela escolha de características (features), seguida pela escolha de uma função de pontuação e de um processo de decisão. As características de Haar, explicadas na subseção anterior, são as características utilizadas no presente trabalho. Segundo Hewitt (2007), um classificador fraco apenas acerta a resposta certa um pouco mais frequentemente do que uma suposição aleatória iria. Porém, a combinação de vários classificadores fracos, com diversos pesos atribuídos a eles, constituiria um classificador forte que seria capaz de melhor alcançar a solução desejada. Segundo Rojas (2009), a equação a seguir exemplifica a geração de um classificador fraco através de, por exemplo, 11 classificadores fracos:

$$C(x_i) = \alpha_1 k_1(x_1) + \alpha_2 k_2(x_2) + \ldots + \alpha_{11} k_{11}(x_{11}) \tag{2.3}$$

Em que, dado um padrão x_i, cada classificador k_i pode emitir uma opinião $k_j(x_i) \in \{-1, 1\}$e a decisão final será $C(x_i)$, sendo que $\alpha_1, \alpha_2, \ldots, \alpha_{11}$são os pesos atribuídos aos classificadores.

Segundo Rojas (2009), o algoritmo AdaBoost trabalha em processo iterativo, em que, após receber os classificadores fracos, faz o ranking deles e atribui os pesos correspondentes, de acordo com a relevância do elemento a cada iteração. No início, atribui-se o mesmo peso a todos os elementos e, a seguir, utiliza-se um exemplo mais difícil a cada iteração e assim são construídos os pesos. No fim, espera-se obter um classificador forte, mais acurado que cada classificador fraco.(ATHITSOS, 2012)

c) Discrete Cosine Transform (DCT)

Segundo Britanak (2001), a transformada discreta cosseno (discrete cosine transform - DCT) é membro da família de transformadas unitárias senoidais. Elas tem diversas aplicações em processamento de imagens, em particular em compressão e descompressão de imagens.

Segundo (Sharkas, 2005), aplicar o DCT em uma imagem a decompõe em uma soma ponderada de sequências básicas de cossenos. A DCT em 2 dimensões é dada por:

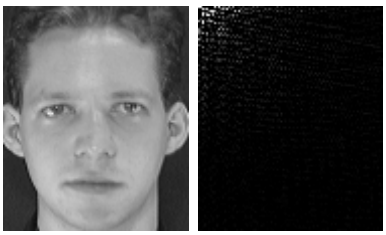
$$C(u,v) = \frac{2}{\sqrt{MN}}\alpha(u)\alpha(v)\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y)\cos\left[\frac{(2x+1)u\pi}{2M}\right]\cos\left[\frac{(2y+1)v\pi}{2N}\right] \quad (2.4)$$

para u=0,1,2,...,W-1 e v=0,1,2,...,H-1.

A transformada inversa (IDCT) é dada por:

$$I(x,y) = \frac{2}{\sqrt{HW}}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1} \alpha(u)\alpha(v)C(x,y)\cos\left[\frac{(2x+1)u\pi}{2W}\right]\cos\left[\frac{(2y+1)v\pi}{2H}\right] \quad (2.5)$$

O DCT foi aplicado à imagem da face da Figura 2a através da utilização da função dct() presente na biblioteca OpenCV e obteve-se os coeficientes DCT mostrados na Figura 2b.



(a) (b)

Figura 2 – Imagem de uma face (Extraído de: The Database of Faces - AT&T Laboratories Cambridge) antes (a) e após a aplicação do DCT (b)

Sabendo que, em imagens, o eixo x e o eixo y começam no canto superior esquerdo e apontam respectivamente para frente e para baixo, nota-se que a maior parte das energias se concentra em baixas frequências. Em seguida, a face é reconstruída com blocos de tamanhos 8x8, 16x16 e 32x32, ou seja, descartando-se as frequências mais altas, para em seguida ser enviada ao PCA, que será explicado com mais detalhes na próxima seção. A reconstrução é feita através da utilização da função idct() presente na biblioteca OpenCV.

(a) (b) (c)

Figura 3 – Imagem da face da Figura 2a após reconstrução com blocos 8x8 (a), 16x16 (b) e 32x32 (c)

Observa-se que foi possível a obtenção de imagens que apresentam bordas mais suaves e contornos da face com menor variação nos tons de cinza. A face reconstruída com bloco de tamanho 32x32 na Figura 3c já se assemelha muito à imagem original da Figura 2a, enquanto a face reconstruída com bloco de tamanho 16x16 na Figura 3b visualmente não se assemelha muito à imagem original por estar um pouco borrada, porém já é suficiente para ser identificada alguma melhora em comparação com o reconhecimento que se utiliza apenas de PCA, em alguns casos, como será demonstrado mais adiante.

d) Auto-Faces (Eigenfaces)

O método de reconhecimento facial mais utilizado, e também o primeiro ser utilizado com sucesso denomina-se auto-faces (eigenfaces). Esse método baseia-se em um método matemático conhecido por análise de componente principal (Principal Component Analysis – PCA). A análise de componente principal parte das seguintes premissas para realizar o reconhecimento facial (PENTLAND; TURK, 1991):

• uma imagem que contém uma face, de dimensões N x N, se representada por um vetor no espaço dimensional N^2 ocupará um único ponto;

• imagens de face sendo semelhantes em configurações gerais, não podem ser distribuídas aleatoriamente no espaço N^2;

• por isso, essas imagens podem ser descritas em um subespaço de pequenas dimensões. A partir dessas premissas a ideia principal que constitui a análise de componente principal é encontrar os vetores que melhor contabilizam a variação de faces em todo o espaço constituído pelas diversas faces detectadas pelo sistema. Tais vetores recebem o

nome de auto-vetores, e a partir deles, pode-se construir um espaço de faces e projetar cada imagem de face nesse espaço. Cada uma dessas imagens de face projetadas recebe o nome de auto-face (eigenface). Partindo de um conjunto de m imagens de dimensões N x N, pode-se alimentar algum sistema de aprendizagem de máquina com a imagem média, definida pela equação 2.6:

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{2.6}$$

onde: µ é o vetor imagem média obtida como resultado do treino; m é o número total de imagens que compõem o treino; x_i é o o vetor que concatena os valores dos pixels que compõem cada linha constituinte da imagem i. Por exemplo, seja uma imagem A constituída por 9 pixels, dispostos em 3 linhas e 3 colunas, tal que:

$$A = \begin{bmatrix} 1 & 3 & 5 \\ 2 & -5 & 1 \\ 2 & 4 & 4 \end{bmatrix}$$

O vetor $x_i$que concatena essa imagem é o seguinte: $x_i = \begin{bmatrix} 1 \\ 3 \\ 5 \\ 2 \\ -5 \\ 1 \\ 2 \\ 4 \\ 4 \end{bmatrix}$.

Cada imagem difere da imagem média de acordo com r_i, de acordo com a equação 2.7:

$$r_i = x_i - \mu \tag{2.7}$$

Calculado o vetor r_i que define a diferença entre cada imagem e a idade média, pode-se obter a matriz de covariância C do sistema, através da equação 2.8:

$$C = AA^T \tag{2.8}$$

onde: $A = [r_1, r_2, ..., r_m]$ e A^T é sua matriz transposta. Após encontrar a matriz de covariância C, pode-se encontrar os auto-vetores v_i, definidos pela equação 2.9:

$$A^T A v_i = \mu_i v_i \tag{2.9}$$

onde: μ_ié cada linha que compõe o vetor imagem média obtida após o treino.

Calculados os auto-vetores v_i , podemos encontar os vetores que compõem o espaço de faces U, utilizando a equação 2.10:

$$u_i = Av_i \tag{2.10}$$

onde: u_i corresponde a cada vetor coluna da matriz espaço de faces U.

Agora, tendo definido o espaço de faces U, pode-se determinar a projeção de cada face no espaço de faces, a qual é definida pela equação 2.11:

$$p_k = U^T(x_k - \mu) \tag{2.11}$$

onde: $k = 1, 2, 3, ..., m$, sendo m o número de imagens que compõem o treino, e p_k é o vetor projeção de cada face k no espaço de faces.

Assim, torna-se fácil introduzir uma imagem de teste no espaço de faces, utilizando a equação 2.12:

$$p = U^T(x - \mu) \tag{2.12}$$

onde: p é o vetor projeção da imagem de teste no espaço de faces.

Após introduzir a imagem da face de teste no espaço de faces, é conveniente definir a distância entre ela e cada imagem de face de treino. Essa distância é calculada pela equação 2.13:

$$\varepsilon_k^2 = ||p - p_k||^2 \tag{2.13}$$

onde: $k = 1, 2, 3, ...,m$.

A etapa seguinte consiste em encontra o valor do limiar θ que determina a maior distância entre duas imagens. Esse valor é determinado pela equação 2.14, a seguir:

$$\theta = \frac{1}{2} \max_{j,k} \left\{ ||p_j - p_k||^2 \right\} \tag{2.14}$$

onde: $j, k = 1, 2, 3, ..., m$.

Finalmente, pode-se obter a distância ε entre a imagem original x e sua imagem reconstruída a partir do espaço de faces de x_f. Essa distância é fornecida pela equação 2.15:

$$\varepsilon^2 = ||x - x_f||^2 \tag{2.15}$$

onde: $x_f = U_x + \mu$.

O processo de reconhecimento segue as seguintes regras:

- Se $\varepsilon \geq \theta$, a imagem de entrada não é a imagem de uma face;
- Se $\varepsilon < \theta$ e $\varepsilon_k \geq \theta$, para todo k, então a imagem de entrada contém uma face desconhecida;

- Se $\varepsilon < \theta$ e $\varepsilon_k^* = min\{\varepsilon_k\} < \theta$, a imagem de entrada contém a face do indivíduo k^*.

2.2 Estado da Arte em detecção e reconhecimento de faces

A tarefa de reconhecimento de face é realizada usualmente em duas etapas, sendo a primeira a de detecção da face e a segunda o seu reconhecimento propriamente dito. Para a primeira etapa, diversos algoritmos são aplicáveis, tais como o Adaptive Boosting (AdaBoost) (FAN et al., 2012; WU et al., 2012; ZHAO; CHUA, 2008) que gera classificadores fortes a partir da combinação linear de classificadores mais fracos, e o Cascata (FAN et al., 2012; ZHANG, 2008; FADZIL; ABU, 1994), que pode ser realizado, por exemplo através do localizador da cabeça através dos níveis de cinza e do localizador de olhos (FADZIL; ABU, 1994) ou pela verificação da cor da pele, da simetria da face e novamente dos olhos (ZHANG, 2008). Já para a segunda etapa, verifica-se que os melhores resultados são gerados com o uso de métodos como o Principal Component Analysis (PCA) (XUAN; NITSUWAT, 2007; LONE; ZAKARIYA; ALI, 2011; KAR et al., 2006), que gera um subespaço com menor covariância através do cálculo da distância euclidiana entre a imagem obtida do rosto da pessoa que se deseja verificar e compara com subespaços já armazenados, e o Discrete Cosine Transform (DCT) (EKENEL et al., 2007; LONE; ZAKARIYA; ALI, 2011; KAR et al., 2006), permite filtrar os componentes de alta frequência da imagem, permitindo-se assim obter uma imagem com bordas mais suaves e com contornos da face com menor variação na tonalidade das cores, além de eliminar possíveis ruídos existentes na imagem adquirida. Atualmente, muito se utiliza da abordagem multi-algorítmica (LONE; ZAKARIYA; ALI, 2011; KAR et al., 2006), obtendo-se resultados muito satisfatórios no reconhecimento correto de face, com taxa de acurácia da ordem de 90% (EKENEL et al., 2007; XUAN; NITSUWAT, 2007; FADZIL; ABU, 1994; ZHANG, 2008; LONE; ZAKARIYA; ALI, 2011).

Entre outros métodos utilizados, pode-se citar o método de redes neurais artificiais (FADZIL; ABU, 1994), que é uma alternativa ao AdaBoost, porém para a tarefa específica de reconhecimento de face se mostrou menos eficiente que este, em comparação com os outros trabalhos. Também pode-se citar o método das Cadeias de Markov, que é uma alternativa ao PCA e se diferencia deste por separar as pessoas a serem reconhecidas em um espaço de estados e em seguida calcular uma distribuição de probabilidades associada a pesos definidos para cada pessoa, além de ser um processo iterativo (ZHAO; CHUA, 2008; WU et al., 2012), mostra-se um novo método, em que o uso combinado de imagens visuais e imagens térmicas obtidas por meio de câmeras infravermelho aumenta a performance do sistema de reconhecimento facial, principalmente no que diz respeito a texturas. As maiores dificuldades encontradas na detecção e no reconhecimento de face estão na presença de características instáveis, como a utilização de óculos ou presença de barba e bigode, por exemplo, que influenciam a efetividade do sistema. Além disso,

a iluminação ambiente também produz um impacto nessa efetividade, já que podem ser observadas diferentes áreas claras e escuras. Outro ponto a ser observado, em sistemas baseados em captura de imagens por vídeo com sua utilização em tempo real, é a utilização da menor quantidade de algoritmos possível, aumentando a velocidade e reduzindo o consumo de memória (WU; WU; MENG, 2010).

Os melhores métodos que existem hoje utilizam uma método particular de PCA, denominado autofaces (eigenfaces) que, com o uso de uma medida de interpenetração de superfícies calculada em quatro regiões (áreas circular e elíptica ao redor do nariz, parte superior da face e face inteira), consegue obter uma taxa de acurácia de 100% em reconhecimento automático (JENKINS; BURTON, 2008). O método de autofaces consiste de um PCA que sobrepõe diversos sub-espaços gerados e obtém uma imagem que será sobreposta às imagens armazenadas em um banco de dados para, assim, utilizar algum tipo de medida de similaridade e comprovar se a face possui uma equivalente armazenada.

A Tabela 1 apresenta um comparativo entre as principais referências utilizadas, os algoritmos utilizados e os resultados apresentados.

Referência	*Redes Neurais*	*AdaBoost*	*PCA*	*DCT*	*Resultados*
(FAN et al., 2012)		X			Dual-thread - vídeo fluente Timer - vídeo não fluente
(EKENEL et al., 2007)			X	X	taxa de reconhecimento correto (testes próprios) 80,6% - DCT 68,7% - PCA
(JENKINS; BURTON, 2008)			X		taxa de reconhecimento correto PCA - 100%
(XUAN; NITSUWAT, 2007)			X		taxa de reconhecimento correto 85,6% - média PCA 5 bases de dados
(FADZIL; ABU, 1994)	X				taxa de reconhecimento correto 87% - testes próprios
(LONE; ZAKARIYA; ALI, 2011)			X	X	taxa de reconhecimento correto Database de Cambridge 90,56% - PCA 93,24% - DCT 94,24% - PCA-DCT
(KAR et al., 2006)			X	X	taxa de reconhecimento correto 81% - DCT 83% - PCA
(ZHANG, 2008)	X				taxa de reconhecimento correto 95% - cascata

Tabela 1 – Comparativo entre as principais referências bibliográficas

Revisitando a literatura, é possível encontrar alguns métodos que divergem entre si para realização de experimentos de reconhecimento de face em vídeos.

Um dos métodos encontrados estabelece modelos dinâmicos para o treino das faces detectadas. Com isso, pode-se obter equações que descrevem o movimento de cada uma das frames adquiridas. Conhecendo-se essas equações, torna-se fácil obter a distância

entre as faces detectadas, utilizando-se, por exemplo, a equação da distância de Frobenius, conforme experimentos descritos em Hadid e Pietikainen (2009) . Utilizando-se, por exemplo, as 20 imagens de face mais próximas entre si, estabelece-se uma sequência mais confiável de imagens a serem projetadas no espaço de faces, uma vez que, imagens borradas, ou ainda que contenham apenas uma parcela da face, tendem a ter distâncias maiores entre si. Assim, a taxa de reconhecimentos corretos tende a melhorar. Vale ressaltar que as frames podem ser adquiridas por diversos métodos, entre eles o DCT, e que diferentes métodos de obtenção das frames podem alterar a taxa de reconhecimentos corretos.

Outro método encontrado é baseado em algoritmos de classificação adaptativa, como o Fuzzy ARTMAP, por exemplo. Nesse método, a cada sequência de um determinado números de frames adquiridas é realizada uma classificação das mesmas de acordo com as diferentes características que interessam ao sistema. A essas características são atribuídos pesos específicos e as frames com maior valor médio ponderado dessas características obtêm melhor classificação. Somente as primeiras frames são utilizadas para o teste(CONNOLLY; GRANGER; SABOURIN, 2012). Com isso, determina-se que apenas as imagens com maior probabilidade de reconhecimento correto sejam introduzidas no sistema.

Outro método, utilizado é o descrito em Mian (2011). Um conjunto com 10 imagens é obtido de cada indivíduo na fase de treino. As características das faces das imagens que compõem esse grupo são extraídas utilizando-se SIFT (Scale Invariant Feature Transform). Após isso, as imagens são classificadas de acordo com essas características, sendo que as duas imagens com menores erros são utilizadas para determinar o indíce de similaridade do grupo. Na fase de reconhecimento, o índice de similaridade de cada grupo de frames adquiridos é comparado com os índices dos grupos que compõem o banco de dados, sendo que o grupo é atualizado a cada 900 ms.

2.3 Java Agent Development Framework (JADE)

Segundo (BELLIFEMINE; POGGI; RIMASSA, 1999), o crescimento em quantidade de informações compartilhadas em rede requer que os sistemas de informação sejam distribuídos em uma rede e interoperem com outros sistemas. A implementação de sistemas multi-agentes ainda encontra-se em estágio de desenvolvimento. Tais sistemas são de extrema importância na atualidade, onde surge a necessidade de unificar as informações adquiridadas pelos diversos agentes constituintes da rede. Para facilitar essa unificação, surgiu o protocolo FIPA, que visa facilitar a interoperabilidade entre esse agentes.

O protocolo FIPA (Foundation for Intelligent Physical Agents) visa especificar as mensagens-chave e os agentes-chave de forma a tornar padronizável e mais simplificada a implementação de sistemas multi-agentes. Esse protocolo baseia-se em duas principais regras: a primeira é que o tempo para se alcançar o consenso e completar o novo padrão

deve ser o mais curto possível e, principalmente, ele deve agir não de modo a impedir o progresso, mas sim como um facilitador, permitindo que as indústrias façam acordos de padronização. A segunda regra é que apenas o comportamento externo dos componentes deve ser especificado, deixando detalhes de implementação e arquiteturas internas a cargo dos agentes desenvolvedores do sistema. É nesse contexto que se instaura a plataforma JADE (Java Agent Development Framework), que é um sistema proprietário em sua arquitetura interna, porém permite estabelecer a comunicação entre os diferentes agentes do sistema através do protocolo FIPA.

A grande vantagem da utilização do JADE é a redução de tempo necessário para a unificação dos diversos agentes constituintes pelo sistema, uma vez que basta especificar as mensagens-chave que serão transmitidas e recebidas pelos agentes-chave e deixar toda a arquitetura interna a cargo da plataforma.

O protocolo FIPA estabelece três funções mandatórias principais:

• gerenciador de agentes do sistema (AMS) - é o agente quem controla e supervisiona o acesso e o uso à plataforma;

• canal de comunicação do agente (ACC) - é o agente quem estabelece o padrão para contato entre agentes internos e externos à plataforma;

• diretório facilitador (DF) - é o agente quem provê o serviço de busca à plataforma.

2.4 Banco de Dados Orientado a Grafos

Os bancos de dados orientados a grafos fazem parte de um grupo maior de banco de dados, os bancos de dados NoSQL. Tais bancos de dados caracterizam-se por não utilizarem o Modelo Relacional e serem mais flexíveis quanto às propriedades ACID (atomicidade, consistência, isolamento e durabilidade). Os banco de dados relacionais baseiam-se no Teorema CAP (consistência, disponibilidade e particiomento), o qual afirma ser possível garantir apenas duas dessas três características simultaneamente. Esse conceito estende-se ao paradigma BASE (Basically Available, Soft State, Eventual consistency) , ou seja, os bancos de dados NoSQL funcionam continuamente, porém garantem a consistência apenas eventualmente (BRITO, 2010). Com isso, nos momentos em que não há a necessidade de se garantir a consistência, o sistema tem a sua performance melhorada, uma vez que reduz-se o número de operações.

Os bancos de dados NoSQL surgiram da necessidade de flexibilizar a forte estruturação imposta pelo Modelo Relacional, o que acaba culminando em uma maior performance e alta escalabilidade. Os bancos de dados relacionais exigem uma estrutura de distribuição vertical de servidores, ou seja, à medida em que se aumenta a quantidade de dados, a quantidade de memória física em disco exigida pelo sistema também aumenta, o que acaba impondo a necessidade de servidores de alta performance, com grande capaci-

dade de processamento e de armazenamento e, principalmente, alto custo. Já os bancos de dados NoSQL permitem uma estrutural horizontal de funcionamento: à medida em que a quantidade de dados vai aumentando, pode-se aumentar a quantidade de servidores para suprir a demanda, sendo que esses servidores não necessitam ser de alta performance. O sistema de buscas do Google, por exemplo, não seria possível hoje em dia sem a utilização de banco de dados NoSQL, pois, se fossem utilizados bancos de dados relacionais, não existiria computador, nem disco rígido, capazes de suportar a grande quantidade de dados armazenados, da ordem de petabytes. Essa estrutura horizontal dos banco de dados NoSQL também é a responsável por manter o tempo de busca constante, independente da quantidade de informações contidas no sistema, o que não ocorre com os bancos de dados relacionais, onde devido a sua estrutura verticalizada, o tempo de busca tende a aumentar uma vez que aumenta a quantidade de dados contido no sistema, pois há a necessidade de se percorrer uma maior quantidade de linhas da lista que armazena esses dados. Essa consistência do tempo de busca é verificada nos bancos de dados orientados a grafos, onde a estrutura horizontal dos nós permite que seja possível percorrer rapidamente todos os nós de um grafo através de uma busca binária, por exemplo. Soma-se a isso ainda o fato de raramente ter de se percorrer todos os nós de um grafo para se encontrar a informação desejada, o que é muito útil para aplicações online, como em reconhecimento de faces para robôs sociáveis.

As diferenças entre banco de dados relacionais e banco de dados NoSQL são explicitadas na tabela abaixo, extraída de Brito (2010):

	Relacional	NoSQL
Escalonamento	Possível, mas complexo. Devido à natureza não estruturada do modelo, a adição de forma dinâmica e transparente de novos nós no grid não é realizada de forma natural.	Uma das principais vantagens desse modelo. Por não possuir nenhum tipo de esquema pré-definido, o modelo possui maior flexibilidade o que favorece a inclusão transparente de outros modelos.
Consistência	Ponto mais forte do modelo relacional. As regras de consistência presentes propiciam um maior grau de rigor quanto à consistência das informações.	Realizada de modo eventual no modelo: só garante que , se nenhuma atualização for realizada sobre o item de dados, todos os acessos a esse item devolverão o último valor atualizado.
Disponibilidade	Dada a dificuldade de se conseguir trabalhar de forma eficiente com a distribuição dos dados, esse modelo pode não suportar a demanda muito grande de informações do banco.	Outro fator fundamental do sucesso desse modelo. O alto grau de distribuição dos dados propicia que um maior número de solicitações aos dados seja atendida por parte do sistema e que o sistema fique menos tempo não-disponível.

Tabela 2 – Diferenças entre banco de dados relacional e NoSQL, extraída de (BRITO, 2010)

Segundo (ROBINSON; WEBBER; EIFREM, 2013), um grafo é um conjunto de nós e o relacionamento que os conecta. Esta estrutura expressiva permite modelar diversos tipos de cenários, da construção de uma nave espacial, a sistemas de estradas; de cadeias logísticas, a históricos médicos de populações, etc.

Um banco de dados orientado a grafos é sistema de gerenciamento *online* com métodos de Criar, Ler, Atualizar e Deletar (CRUD) que expõem um modelo de dados de um grafo. Entre as vantagens da utilização de um banco de dados orientado a grafos, há uma melhora na performance, na flexibilidade e na agilidade do modelo.

Com relação à flexibilidade, os grafos são naturalmente aditivos, o que significa que é possível adicionar novos tipos de relacionamentos, novos nós, e novos subgrafos para uma estrutura já existente, sem comprometer as consultas existentes e a funcionalidade da aplicação. Isso tem um impacto positivo na produtividade do desenvolvedor e no gerenciamento de riscos do projeto, sendo possível adicionar facilmente mudanças ao

projeto a qualquer momento, sem que seja necessário modelar exaustivamente o domínio antes do tempo.

Finalmente, com relação à agilidade, permite-se que haja uma evolução no modelo de dados juntamente com o resto da aplicação, utilizando-se de uma tecnologia alinhada com as práticas atuais incrementais e iterativas de desenvolvimento de software.

Segundo Joshi (2013), a principal desvantagem da utilização de um banco de dados orientado a grafos é a forte conexão de cada vértice e aresta com os elementos adjacentes, o que torna difícil a distribuição da base de dados em servidores.

A utilização de um banco de dados orientado a grafos no presente trabalho visa a criação de uma memória semântica ao robô, com a finalidade de guardar a entidade "Pessoa", com suas informações relevantes, por ora "Nome" e "Imagens". Deseja-se valer do benefício da performance, já que a consulta será executada em tempo real. A falta de consistência originada pelo uso de um banco de dados orientado a grafos não é um problema para o sistema tratado neste trabalho, uma vez que são adquiridas diversas frames em intervalos muito reduzidos de tempo. Portanto, a perda de uma frame não traz nenhum malefício ao bom funcionamento do sistema. A impossibilidade de distribuição da base de dados não representa um impedimento no caso da aplicação a um robô sociável, que usualmente terá utilização apenas de uma máquina.

3 Sistema Proposto

3.1 Visão Geral

Já existem alguns robôs sociáveis desenvolvidos e diversos tipos de reconhecimento facial, que se utilizam das mais variadas técnicas, tanto para detecção (PORTER et al., 2013; NICPONSKI; RAY, 2001) quanto para reconhecimento (BIGIOI; STEINBERG; CORCORAN, 2013; FOLTA et al., 2013; LEE, 2013). Entretanto, a arquitetura da solução planejada para este trabalho apresenta algumas características particulates, tais como a sistematização de uma arquitetura multi-agente com módulos responsáveis pela realização de tarefas específicas, além da utilização de memórias distribuídas para o armazenamento dos diferentes tipos de informações armazenadas pelos diversos sistemas que compõem um robô sociável.

A figura 4 é um esboço da arquitetura planejada.

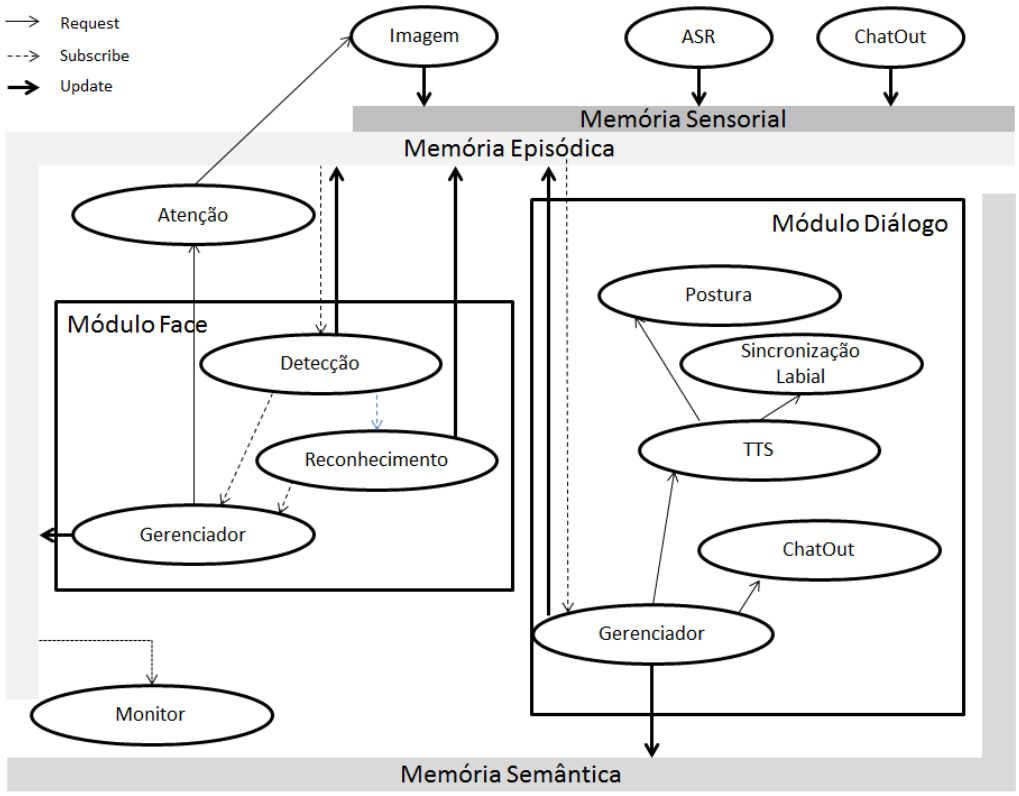


Figura 4 – Visão geral da arquitetura do sistema

Estão destacados dois dos módulos: Face e Diálogo. O módulo Face é responsável pelas atividades relativas ao reconhecimento de face em si, enquanto o módulo Diálogo é uma implementação de um gerenciador de diálogo. A Figura 4 também destaca os módulos de "Memória Semântica" e de "Memória Episódica"; esta última ainda conta

com uma estrutura particular, a "Memória Sensorial". O Módulo Face opera a partir de imagens, obtidas pelo agente "Imagem", armazenadas na Memória Sensorial". O agente "Detecção" é ativado a cada imagem (via envio de mensagens), responsável por detectar faces. Ao detectá-las, deve informar ao agente "Reconhecimento", responsável pela extração de características (*features*) para o reconhecimento. Ambos, "Detecção" e "Reconhecimento", armazenam o resultado de seu processamento na "Memória Episódica". O módulo "Gerenciador" é responsável pela consulta à "Memória Semântica", buscando ver se há identificação da pessoa reconhecida e atualizando a "Memória Episódica" quando isso ocorre.

O Módulo Diálogo opera a partir de trechos de diálogo, obtidos por canais de entrada "ASR" (Automatic Speech Recognition) ou "ChatIn". Estes trechos de diálogo são informados (via envio de mensagens) ao Gerenciador, que é responsável por determinar como será a resposta. A resposta propriamente dita é realizada através de voz (pelos agentes "TTS" (Text-To-Speech), "Sincronização Labial" e "Postura") ou texto (agente "ChatOut").

A figura 5 mostra mais detalhadamente o funcionamento da estrutura de memória do sistema.

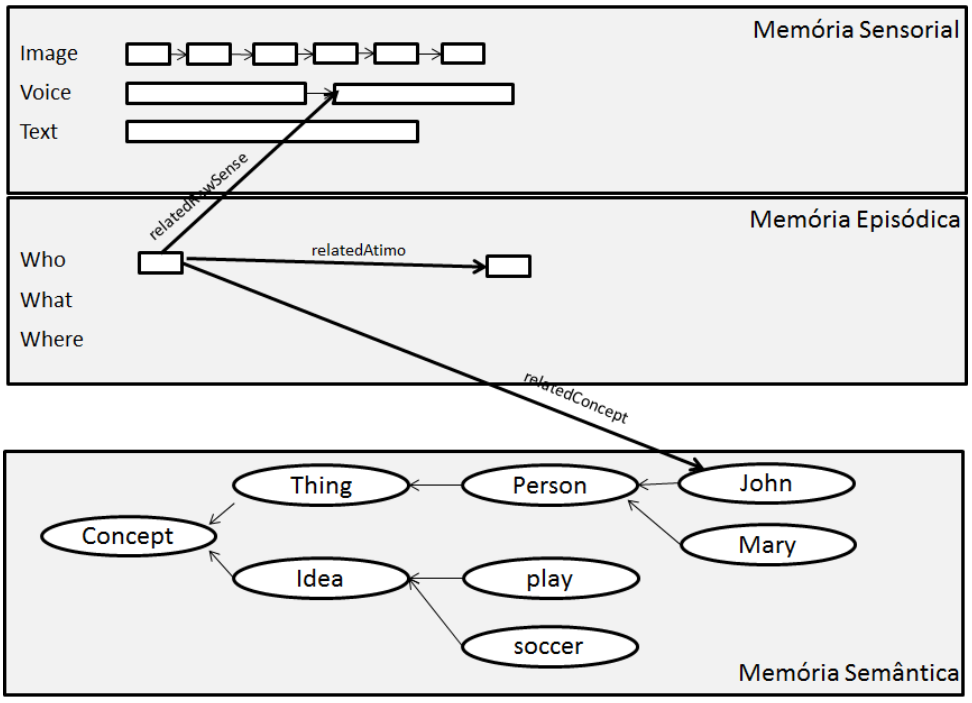


Figura 5 – Visão detalhada da estrutura de Memória do sistema

Todo o armazenamento baseia-se em bancos de dados orientados a grafos (GraphDB). A Memória Sensorial contém canais para cada tipo de sensor como, por exemplo, Imagem, Voz, Texto. Cada amostragem do sensor é armazenada, ligando-se à anterior, formando assim a sequência de amostras que pode permitir a reconstrução do sinal original.

A Memória Episódica contém conceituações: no exemplo da Figura 5, a caracterização de que "John" foi identificado em uma determinada imagem. Já a Memória Semântica contém todo o conhecimento tácito (*Concepts*, como serão aqui denominados), divididos em *Thing* ("coisas" que tem uma estória, podendo ser uma pessoa, uma árvore ou mesmo uma pedra) e *Idea* (ideias, conceitos abstratos).

O presente trabalho foca-se no estudo do Módulo Face, que será descrito com mais detalhes nas seções a seguir.

3.2 Algoritmos Utilizados

3.2.1 Detecção de Face

Para a detecção de face, foi implementado o método proposto por Viola e Jones (2004), o qual utiliza apenas as quatro principais características de Haar, as quais podem ser observadas na Figura 6.

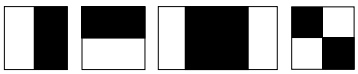


Figura 6 – As quatro principais características de Haar

A vantagem da utilização desse método é que dado o pequeno número de características de Haar utilizadas, o programa é executado mais rapidamente do que seria se fossem utilizadas todas as características, podendo ser implementado em câmera com um baixo nível de travamentos. Tal solução foi baseada na utilizada por Hewitt (2007).

3.2.2 Tracking de Face

O módulo de Tracking implementado no primeiro semestre foi baseado no algoritmo Continuously Adaptive Mean Shift (Camshift), já implementado na biblioteca OpenCV. Neste módulo, o vídeo é atualizado frame a frame e a face detectada é rastreada baseando-se na probabilidade de haver uma face naquele local, dado a existência de face verificada nos frames anteriores.

Devido aos cálculos estatísticos realizados o programa torna-se lento e, no caso de a face transladar muito rapidamente, o rastreamento da face falha, devido a grande mudança de posição verificada em poucos frames.

3.2.3 Reconhecimento de Face

O módulo de reconhecimento propriamente dito, primeiramente obtém a transformada DCT da imagem da face. Após isso, a imagem é reconstruída através da IDCT, utilizando-se 16 coeficientes, pois esse foi o menor número de coeficientes com o qual a

imagem apresentava a qualidade necessária para o reconhecimento, sem perdas de informação.

Após a reconstrução da imagem com auxílio da IDCT, a imagem passa para a função que realiza a PCA, utilizando-se do método de auto-faces. Ao final desse processo, teremos uma reconstrução da imagem inserida no espaço de faces. Vale ressaltar que para que isso seja possível, deve existir um treino prévio do sistema, baseado em aprendizagem de máquina utilizando AdaBoost.

O treino realizado deve conter diferentes tipos de faces em diversas condições, com boa iluminação, com baixa iluminação, de frente, rotacionadas de diferentes indivíduos dos sexo masculino e feminino, com e sem óculos, com e sem barba e/ou bigode.

Após inserir a imagem da face a ser reconhecida no espaço de estados, deve-se buscar a face que mais se assemelha a ela, caracterizando a fase de reconhecimento propriamente dito.

3.3 Agentes

Os agentes apresentados na Figura 4 e implementados em código Java, com a utilização do JADE são apresentados com mais detalhes a seguir.

3.3.1 Agente Imagem (*ImageAgent*)

O agente Imagem é o primeiro agente acionado durante o funcionamento do sistema. Ele é o responsável por adquirir as frames obtidas pela webcam ou por qualquer outro tipo de câmera, como uma câmera PTZ, por exemplo. O agente Imagem salva momentaneamente no banco de dados da memória sensorial a imagem da frame atual e informa ao agente Detecção para que o mesmo possa iniciar o processo de detecção. Em linguagem de programação do JADE, o agente Imagem envia ao Agente Detecção uma ACLMessage do tipo INFORM com o número da frame atual capturada.

O agente Imagem é um agente do tipo TickerBehaviour, ou seja, seu funcionamento é cíclico. Neste trabalho, foi determinado o período de 1000 ms como tick para seu funcionamento, ou seja, ele adquire uma frame a cada 1s.

3.3.2 Agente Detecção (*DetectionAgent*)

O agente Detecção possui comportamento cíclico, ou seja, ele é acionado cada vez que o agente Imagem adquire uma frame. Ele recebe essa informação e então carrega a frame atual. No contexto de controle, a comunicação entre os agentes configuraria um comportamento do tipo master-slave, em que o Agente Imagem envia um comando e o Agente Detecção executa o seu trabalho. Após isso, ele determina se há imagem de faces

na frame recebida, utilizando o algoritmo (VIOLA; JONES, 2004). Se detectadas faces na frame, ele repassa as posições na imagem da diagonal do quadrado que representa cada face ao agente Reconhecimento e ao agente Monitor e as salva na memória episódica.

3.3.3 Agente Reconhecimento (*RecognitionAgent*)

O agente Reconhecimento também é do tipo cíclico, pois seu acionamento depende de haver detecção de face. Ao receber uma mensagem contendo as informações de faces detectadas pelo agente Detecção, ele acessa a classe *detectedPerson* para obter as coordenadas da frame das faces, localizadas no banco de dados da memória episódica, que serão introduzidas no espaço de faces determinado pelo algoritmo eigenfaces. Após o cálculo da distância euclidiana da face detectada em relação à imagem média obtida nas etapas de treino, o agente Reconhecimento busca no banco de dados localizado na memória semântica uma pessoa cuja face tenha a maior similaridade com a face detectada, sendo que essa similaridade deve ser obrigatoriamente acima de 75% (limiar determinado experimentalmente) para reconhecimento como uma pessoa conhecida. Ao final do procedimento, o agente Reconhecimento envia uma mensagem ao agente Monitor informando o nome da pessoa ou se a pessoa é desconhecida e as coordenadas do local onde o nome deve ser impresso.

3.3.4 Agente Monitor (*MonitorAgent*)

O agente Monitor recebe informações do agente Detecção e do agente Reconhecimento, os quais são cíclicos, portanto, seu funcionamento também deve ser cíclico. Em termos gerais, ele é o agente responsável pelo gerenciamento de todas as informações que devem ser exibidas na frame.

Após receber uma mensagem do agente Detecção e também do agente Reconhecimento e confirmar que as informações são válidas, o agente Monitor acessa as classes necessárias para exibir uma moldura ao redor da face detectada e também exibe o nome da pessoa reconhecida, bem como outras informações que possam constar sobre a pessoa no banco de dados da memória semântica.

3.3.5 Regras FIPA no contexto do sistema

No presente trabalho, as regras FIPA mencionadas anteriormente se encaixam no seguinte contexto: o agente Imagem controla o acesso ao sistema, pois, o processo de detecção de faces somente inicia-se a partir do momento em que o agente Imagem envia a frame adquirida através da câmera para que o agente Detecção possa iniciar a detecção da face, atentendo à função AMS.

Também é o agente Imagem quem determina o padrão de acesso ao sistema, uma vez que somente imagens ou vídeos podem ser transmitidos ao sistema. Já o agente

Monitor é o responsável por controlar o padrão de contato com agentes externos, sendo o responsável por transmitir as informações sobre a pessoa reconhecida enviando strings com informações como o nome da pessoa por exemplo, atendendo à função ACC.

Por fim, o agente Reconhecimento é o responsável por prover o serviço de busca, acessando o banco de dados orientado a grafos para encontrar uma possível identificação da face detectada com a face de algum sujeito nele contido, atendendo, assim, à função DF.

3.4 Memória

3.4.1 Memória Sensorial

A Memória Sensorial é responsável pelo armazenamento dos sinais sensoriais, particularmente os sinais físicos, como imagens e voz. Na visão aqui apresentada, é como se fora parte da Memória Episódica, estando a ela fortemente relacionada. Entretanto, as características de armazenamento são distintas e, por isso, foi dela destacada neste detalhamento.

A Memória Sensorial deveria armazenar sinais contínuos, como impulsos visuais, auditivos e táteis. Entretanto, deve sempre armazená-los de forma discreta, por imposição da implementação computacional. A sequência das amostras deve permitir recuperar o sinal contínuo. A implementação dá-se então a partir do banco de dados orientado a grafos, usando-se cada nó como uma amostra, pertencente à classe RawSense, exemplificada a seguir:

```
class RawSense {
id: rawId
type: string // image, voice, text
startTime: timestamp
endTime: timestamp
samplingFilename: string
samplingFormat: string
}
```

3.4.2 Memória Episódica

A Memória Episódica é a memória de todos os eventos (tempo, lugar, emoções e todo o contexto). É a coleção das experiências pessoais que ocorreram. Atua como "memória global", no sentido de ser acessível a todos os agentes do sistema. Seu papel é passivo, em geral; o mecanismo de subscrição de eventos que implementa não deve ser confundido como um papel ativo mas simplesmente uma facilidade arquitetural para extensibilidade do sistema. Na visão aqui adotada, a EM (Episodic Memory) tem tamanho infinito, ou

seja, não há o "esquecimento" que ocorre na memória episódica dos humanos. Entretanto, sustenta-se que um robô deve manter a memória de todos os eventos que presenciou, não sendo adequado que "esqueça-se", como um humano. E que a evolução tecnológica tornará menos crítica a limitação da capacidade de armazenamento. É função da EM armazenar "quem fez o que e quando o fez"; assim, as palavras-chave são "WHO", "WHAT" e "WHEN". Além disso, planos de ação, emoções e outros tipos de eventos (externos ou interiores) também tem seu lugar nesta estrutura. Sua conexão com a Memória Semântica é estreita: na realidade, cada elemento que armazena refere-se a um *Concept* (*Thing* ou *Idea*), armazenado na Memória Semântica (SM, Semantic Memory). A EM estrutura-se em grafo, usando banco de dados orientado a grafos. Cada nó do grafo pertence à classe de exemplo ATIMO:

```
class Atimo {
id: atimoId
startTime: timestamp
endTime: timestamp
generalChannel: channelOntologyId //who,what,where
specificChannel: channelOntologyId
}
```

Cada conexão (edge) do grafo pertence à classe AtimoConnection:

```
class AtimoConnection {
id: connectionId
type: string //relatedAtimo, relatedRawSense, relatedConcept
}
```

Note-se que um mesmo átimo pode aplicar-se a um período de tempo (startTime, endTime) e não somente a um instante. Além disso, cada átimo pode relacionar-se a vários outros átimos (relatedAtimos) assim como a um RawSense da Memória Sensorial. Finalmente, cada átimo está relacionado a um *Concept* na SM (relatedConcept). Estas conexões estão representadas na fig.2.

3.4.3 Memória Semântica

A Memória Semântica (SM) armazena o significado de tudo que é conhecido. Na visão presente, a SM é um grafo, representado como na Figura 5. A raiz é *Concept*, do qual derivam duas categorias essenciais: *Thing* e *Idea*. A partir de *Thing*, tem-se a representação acerca de todas as coisas do mundo concreto, dos reinos animal, vegetal e mineral. Nota-se que tudo, no mundo concreto, tem instâncias e cada uma delas tem a sua estória: a estória de uma pessoa, de uma planta ou de uma pedra. Já *Idea* representa apenas conceitos abstratos, como os verbos e adjetivos. Ou conceitos que não possuem instâncias, como os esportes, cores, etc. São diversas as representações das ontologias que, no final, representam o conteúdo da Memória Semântica.

3.5 Aspectos de Implementação

No presente trabalho, propõe-se uma solução para reconhecer faces em ambiente controlado, implementada inicialmente utilizando-se a linguagem C++ e a biblioteca OpenCV, que inclui centenas de algoritmos de visão computacional. A detecção de faces utiliza um tipo de detector proposto por Viola e Jones (2004) e pode ser aplicada tanto a fotos quanto a vídeos. O reconhecimento de faces utiliza o algoritmo PCA somente e o algoritmo DCT juntamente com o algoritmo PCA. A solução implementada utilizou as funções dos algoritmos mencionados já implementadas no OpenCV e foi baseada na solução implementada por Hewitt (2007) e por Shervin Emami (www.shervinemami.info) em 30 de dezembro de 2011. É possível dividir a solução incial em dois módulos principais: Detecção da Face e Reconhecimento da Face. Também foi implementada uma solução de tracking da face, baseada no algoritmo Camshift. Mais informações são fornecidas nas subseções a seguir.

A solução final foi elaborada com o objetivo de ser utilizada no robô sociável, visando a modularização em agentes e a possibilidade de intercambiabilidade proporcionada pela utilização do JADE. Utilizou-se a linguagem Java e o empacotador JavaCV. O JavaCV foi concebido por uma equipe para um programa de pesquisa de doutorado no Laboratório Okutomi & Tanaka, Tokyo Institute of Technology e fornece empacotadores para bibliotecas utilizadas por pesquisadores no campo de visão computacional, entre elas OpenCV. Possui suporte especialmente para as funções de aplicação de autofaces e de DCT, que foram utilizadas no presente trabalho. Em Java, o sistema é dividido em dois pacotes: *agent* e *facerec*. O pacote agent contém os agentes citados anteriormente: *ImageAgent*, *DetectionAgent*, *RecognitionAgent* e *MonitorAgent*. O pacote *facerec* contém a classe *javacv*, que contém as funções de detecção e reconhecimento utilizadas pelos agentes, além de outras funções que, por exemplo, fazem a impressão de molduras e nomes nas faces reconhecidas. Esse pacote também contém as classes auxiliares *detectedPerson* e *recognizedPerson*, que contém respectivamente informações para a reconstrução da moldura da face detectada e sobre o nome e o ponto da imagem onde ele deve ser impresso. O agente Imagem realiza a amostragem da câmera a uma taxa de 1 frame/s.

O objetivo do trabalho é a implementação em um computador comum, disponível comercialmente (SONY VAIO FIT, 8GB RAM, Processador i7-3537U Quad-Core CPU @ 2.00GHz, Placa de Vídeo Intel HD Graphics 4000). Não há intenção de utilizar computadores com altíssimas velocidades de processamento e grande capacidade de memória.

O conceito relacionado a memórias e a banco de dados orientado a grafos não foi implementado por fugir do escopo do trabalho, porém foi adicionado ao trabalho por ser importante ao sistema em uma implementação futura. O módulo de Tracking foi descontinuado na solução final, pois, como a amostragem realizada compreende de 1 frame/s, é

possível realizar a detecção e o reconhecimento a cada frame, sendo desnecessária para a aplicação um módulo que faça o rastreamento da face.

4 Resultados

4.1 Testes com Fotos

4.1.1 Reconhecimento de face

4.1.1.1 Procedimentos experimentais

Foram realizados diversos testes para verificar a acurácia da solução implementada, utilizando-se a base de dados de faces publicada pela Universidade de Cambridge (The Database of Faces - AT&T Laboratories Cambridge), que consiste em dez imagens de quarenta pessoas distintas, obtidas no laboratório entre Abril de 1992 e Abril de 1994. Para alguns sujeitos, as imagens foram obtidas em ocasiões distintas, variando a iluminação, expressão facial (olhos fechados e abertos, com sorriso e sério) e presença de características instáveis (óculos por exemplo). A utilização desses exemplos foi explorada nos testes. Todas as imagens foram obtidas com um plano de fundo escuro e homogêneo com os sujeitos em posição frontal, com certa tolerância para movimentos laterais. O computador usado para teste foi SONY VAIO VPCF1, 6GB RAM, Processador i7 Quad-Core CPU @ 1.73GHz, Placa de Vídeo NVIDIA GeForce GT 425M.

O primeiro teste (1T), proposto por Hewitt (2007), foi realizado com os sujeitos 1, 2 e 4, utilizando-se a primeira de cada um deles para fazer o aprendizado. Os testes subsequentes podem ser subdivididos em cinco grupos: Sujeitos homogêneos (SH), Presença e ausência de características instáveis – óculos (CI-OC), Presença e ausência de características instáveis – olhos fechados (CI-OF), Presença e ausência de características instáveis – cabeça rotacionada (CI-CR) e Database Completo (DB-CP). Como sugerido pelo nome correspondente, o grupo SH tem como objetivo comparar sujeitos que se parecem, o grupo CI-OC tem como objetivo comparar a presença e a ausência de óculos, o grupo CI-OF tem como objetivo comparar a presença de olhos fechados e o grupo CI-CR tem como objetivo comparar a presença de rotação na cabeça. O grupo DB-CP tem como objetivo verificar o funcionamento do sistema com a utilização de todos os sujeitos disponíveis.

O teste 1T foi realizado em três formatos: no formato original proposto por Hewitt (2007), que utiliza 3 imagens de treino e 18 imagens de teste, em um segundo formato, utilizando-se as 30 imagens de teste e em um terceiro formato, utilizando-se 6 imagens de treino e 30 imagens de teste.

O grupo SH se utilizou de quatro sujeitos no primeiro teste (sujeitos 7, 9, 11 e 23) e de três sujeitos no segundo teste (sujeitos 19, 20 e 38).

O grupo CI-OC se utilizou de três sujeitos nos dois primeiros testes (sujeitos 4, 19 e 20) e também de três sujeitos nos outros dois testes (sujeitos 7, 13 e 17). O primeiro

teste procurou utilizar imagens de sujeitos com óculos para o treino e sem óculos para o teste, enquanto o segundo teste procurou utilizar imagens de sujeitos sem óculos para o treino e com óculos para o teste, em ambos os casos.

O grupo CI-OF se utilizou de três sujeitos no primeiro teste (sujeitos 8, 11 e 24) e utilizou fotos com olhos abertos para o treino e algumas fotos de olhos fechados para o teste, conforme o sujeito em questão.

O grupo CI-CR se utilizou de quatro sujeitos nos três primeiros testes (sujeitos 12, 38, 40 e 35 e 12, 38, 40 e 25) e de três sujeitos nos outros dois testes (sujeitos 29, 40 e 25). O primeiro e o segundo teste utilizaram imagens frontais de sujeitos para o treino e imagens rotacionadas para o teste, modificando um dos sujeitos utilizados. O terceiro teste utiliza os mesmos sujeitos do segundo teste, com a diferença de que este tenta fazer o reconhecimento correto da foto rotacionada do segundo teste, adicionando uma foto também rotacionada ao treino, porém ainda falha. No quarto teste, mantém-se os sujeitos que tiveram fotos não reconhecidas corretamente nos testes anteriores e adiciona-se outro sujeito diferente para teste. No quinto teste, faz-se uma revisão de todas as fotos que não foram reconhecidas corretamente e tenta-se fazer a compensação pela utilização de 80% das imagens dos três sujeitos anteriores no treino. O sexto teste é análogo, com a utilização de imagens dos cinco sujeitos que obtiveram erros nos testes anteriores e no atual.

As imagens dos sujeitos utilizados, separadas por grupos são apresentadas a seguir:

Figura 7 – Sujeitos 1, 2 e 4 (Extraído de: The Database of Faces - AT&T Laboratories Cambridge) utilizados em T1

Figura 8 – Sujeitos 7, 9, 11 e 23 (Extraído de: The Database of Faces - AT&T Laboratories Cambridge) utilizados em SH

Figura 9 – Sujeitos 19, 20 e 38 (Extraído de: The Database of Faces - AT&T Laboratories Cambridge) utilizados em SH

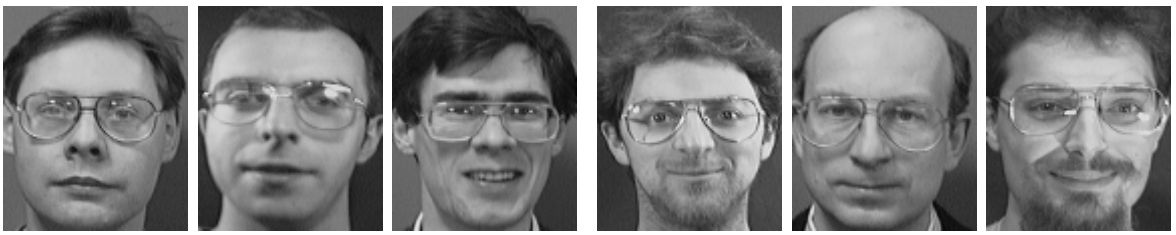


Figura 10 – – Sujeitos 4, 19 e 20 e sujeitos 7, 13 e 17 (Extraído de: The Database of Faces - AT&T Laboratories Cambridge) utilizados em CI-OC

Figura 11 – Sujeitos 8, 11 e 24 (Extraído de: The Database of Faces - AT&T Laboratories Cambridge) utilizados em CI-OF

Figura 12 – Sujeitos 12, 38, 40, 35, 25, 29, 4, 9 e 23 (Extraído de: The Database of Faces - AT&T Laboratories Cambridge) utilizados em CI-CR

As imagens citadas na seção de Conceitos Teóricos, obtidas pela solução implementada em 1T são apresentadas a seguir:

(a) (b)

Figura 13 – Imagem média das três faces, obtida após aplicação do PCA (a) e Imagens projetadas no subespaço após aplicação do PCA – eigenfaces (b)

(a) (b)

Figura 14 – Imagem média das três faces, obtida após aplicação do DCT e do PCA (a) e Imagens projetadas no subespaço após aplicação do DCT e do PCA – eigenfaces (b)

Nota-se novamente bordas mais suaves e contornos da face com menor variação nos tons de cinza, como mencionado na seção DCT.

4.1.1.2 Resultados obtidos

Um resumo de resultados é apresentado na tabela a seguir, que indica o código numérico dos sujeitos utilizados, a quantidade de imagens de treino / imagens de teste, a taxa de reconhecimento correto utilizando somente PCA, o tempo de realização do teste correspondente, a taxa de reconhecimento correto utilizando DCT e PCA, o tempo de realização do teste correspondente e, caso houve erro no reconhecimento, o código do sujeito reconhecido erroneamente / o código da imagem correspondente:

Grupo	*Sujeitos*	*Imagens*	*PCA*	*Tempo*	*DCT+PCA*	*Tempo*	*Erro*
1T	1,2,4	3 / 18	94%	1.3 ms	100%	1.3 ms	s4 / 6
	1,2,4	3 / 30	96%	1.4 ms	100%	1.4 ms	s4 / 6
	1,2,4	6 / 30	96%	0.9 ms	100%	0.9 ms	s4 / 6
SH	7,9,11,23	4 / 24	91%	0.7 ms	91%	0.7 ms	s9 / 6 - s23 / 5
	19,20,38	3 / 18	100%	0.6 ms	100%	0.6 ms	-
CI-OC	4,19,20	15 / 15	100%	1.9 ms	100%	1.6 ms	-
	4,19,20	15 / 15	100%	1.5 ms	100%	1.6 ms	-
	7,13,17	15 / 15	100%	1.4 ms	100%	1.0 ms	-
	7,13,17	15 / 15	100%	1.3 ms	100%	1.2 ms	-
CI-OF	8,11,24	12 / 9	100%	1.0 ms	100%	1.7 ms	-
CI-CR	12,38,40,35	16 / 16	93%	1.2 ms	93%	1.1 ms	s40 / 4
	12,38,40,35	16 / 16	93%	1.4 ms	93%	1.2 ms	s25 / 8
	12,38,40,35	16 / 16	93%	1.1 ms	93%	1.2 ms	s40 / 4
	29,40,25	12 / 12	91%	0.8 ms	91%	1.0 ms	s40 / 4
	29,40,25	24 / 6	83%	1.6 ms	83%	2.0 ms	s40 / 4
	4,9,23,25,40	40 / 10	100%	3.9 ms	100%	3.7 ms	-
DB-CP	Todos	120 / 120	90%	3.9 ms	90%	4.0 ms	-
	Todos	200 / 200	93%	6.4 ms	94%	6.3 ms	-

Tabela 3 – Comparativo entre os testes realizados

Com relação aos testes descritos, notou-se que o sistema funcionou perfeitamente na presença de características instáveis (grupo CI-OC) e na presença de sujeitos com olhos fechados (grupo CI-OF). O sistema porém, enfrentou dificuldades na presença de sujeitos com a cabeça rotacionada (grupo CI-CR). As imagens que foram erroneamente identificadas são apresentadas a seguir:

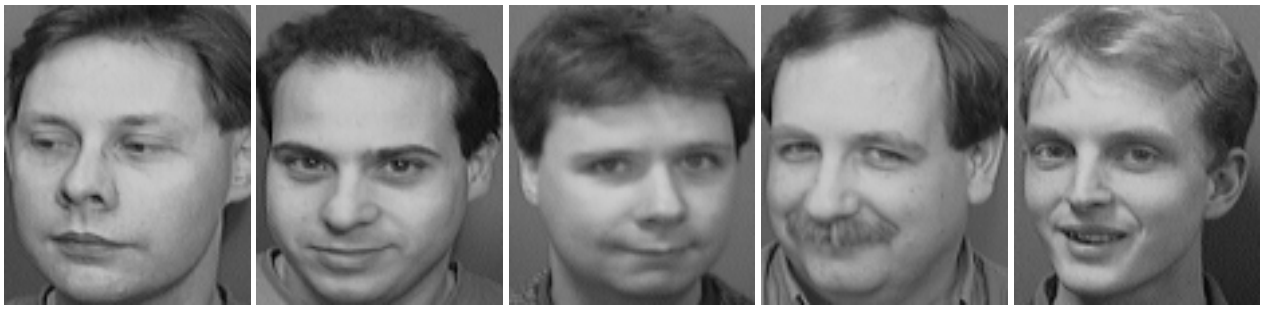


Figura 15 – Imagens (Extraído de: The Database of Faces - AT&T Laboratories Cambridge) identificadas erroneamente pela solução implementada

O último teste do grupo CI-CR que utilizou 80% das faces disponíveis para treino desses cinco sujeitos conseguir obter 100% de sucesso, utilizando-se inclusive as imagens da Figura 15 para teste.

A utilização do DCT antes do PCA apenas ajudou no reconhecimento nos testes do grupo 1T e não fez diferença nos outros grupos. Logo, não é possível concluir que o DCT tem influência positiva no reconhecimento de face.

Por fim, observou-se que o tempo de realização do teste aumentou com a utilização de uma base de dados maior no treinamento, o que se configura em uma limitação ao sistema: a base de dados utilizada no treino não pode ser tão grande quanto se queira, ou seja, é limitada, pois, como deseja-se utilizar a solução para vídeo, a fluidez dele estaria comprometida pela aplicação de um algoritmo lento no processo.

4.2 Testes com Vídeo

4.2.1 Detecção de face

4.2.1.1 Procedimentos experimentais

O Honda/UCSD Video Database foi concebido com o objetivo de fornecer um padrão de avaliação de algoritmos de tracking e reconhecimento de face. Cada sequência de vídeo foi gravada em um ambiente interno a 15 frames por segundo, com resolução de 640x480. Como demonstrado no estado da arte, verifica-se que a variação de pose é um grande desafio para o reconhecimento de face, logo, todos os vídeos contem rotações significativas da cabeça. Além disso, também há a aproximação e afastamento da tela, desejando-se assim testar variações de escala.

Os testes a seguir foram realizados com os vídeos do segundo conjunto do Honda/UCSD Video Database, seguindo padrões encontrados em (MIAN, 2011; HADID; PIETIKAINEN, 2009; CEVIKALP; TRIGGS, 2010). Esse conjunto foi gravado no Laboratório de Visão Computacional, Universidade da Califórnia, São Diego em 2004 e é composto de 30 vídeos de 15 sujeitos, divididos em vídeos de treino e de teste, tendo ambos os subconjuntos a duração aproximada de 50 segundos por vídeo. Em cada vídeo, a pessoa roda e vira sua cabeça na ordem e velocidade de preferência. Na Figura 16, são mostradas imagens de exemplo retiradas de vídeos de treino de 4 sujeitos, após detecção e redimensionamento pelo sistema.

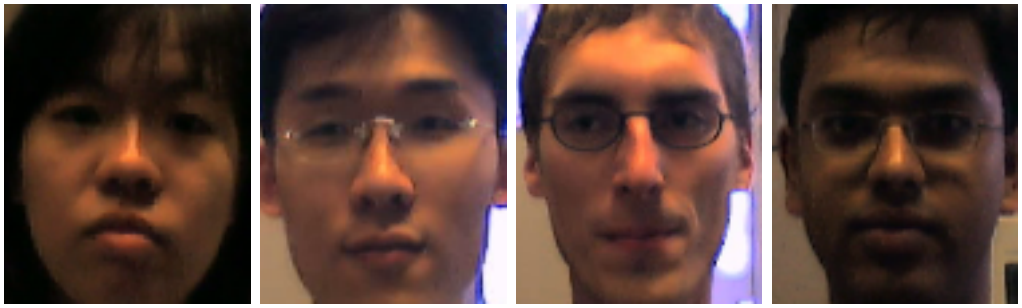


Figura 16 – Imagens dos sujeitos 1, 2, 3 e 4 do Honda/UCSD Video Database, após detecção e redimensionamento, no formato padrão do sistema

As detecções foram obtidas dos vídeos de treino, após a utilização do algoritmo de detecção de face do sistema (VIOLA; JONES, 2004) em cada frame. As faces detectadas pelo sistema vão de 40x40 a 300x300, com aumento de escala de 10% a cada etapa. O computador usado para teste foi SONY VAIO FIT, 8GB RAM, Processador i7-3537U Quad-Core CPU @ 2.00GHz, Placa de Vídeo Intel HD Graphics 4000.

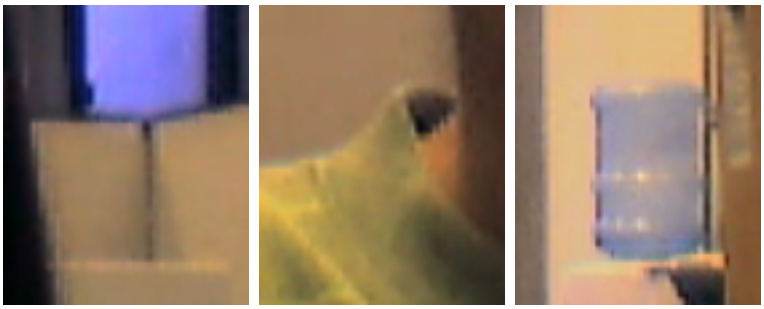
4.2.1.2 Resultados obtidos

A seguir, na Tabela 4, demonstra-se o número total de detecções obtidas no vídeo de treino de cada sujeito, o número de detecções corretas (DC) e o número de falsos positivos (FP), além da taxa de detecções corretas do sistema, em oposição a taxa de detecções de falsos positivos, e também a taxa de detecções corretas do sistema no total. A contagem foi feita por inspeção manual, sendo que todas as detecções obtidas pelo sistema nos vídeos foram salvas em imagem.

Sujeito	*Total*	*DC*	*FP*	*RC(%)*
1	110	110	0	100.00
2	1189	851	338	71.57
3	1145	1126	19	98.34
4	667	664	3	99.55
5	1071	1013	58	94.58
6	1314	1028	286	78.23
7	1021	1009	12	98.82
8	2228	2207	21	99.06
9	1121	1078	43	96.16
10	660	602	58	91.21
11	837	812	25	97.01
12	496	495	1	99.80
13	1222	1196	26	97.87
14	792	709	83	89.52
15	966	936	30	96.89
Total	**14839**	**13836**	**1003**	**93.24**

Tabela 4 – Contagem de Detecções Corretas e Falsos Positivos no teste com vídeos do Honda/UCSD Video Database

Dentre os falsos positivos, verifica-se tanto a detecção de padrões do tipo olhos e nariz (Figura 17a), como de padrões não identificados, como partes de uma gola de camiseta (Figura 17b) ou um filtro de água presente na cena (Figura 17c). Há casos em que esses falsos positivos são objetos de fundo e, como se encontram longe da câmera, poderiam não ser mais detectados caso a escala mínima utilizada fosse maior. Isso comprometeria a distância de detecções e, como o número final de falsos positivos não se mostrou significativa, escolheu-se optar por manter a possibilidade de detecção a maiores distâncias.



(a) (b) (c)

Figura 17 – Imagens de falsos positivos detectados em vídeos de treino do Honda/UCSD Video Database

Através da análise de resultados da Tabela 4, verifica-se que o algoritmo de detecção utilizado pelo sistema (VIOLA; JONES, 2004) se mostra muito eficiente na detecção de face, sendo que a taxa total de detecções corretas é de 93,24%.

4.2.1.3 Testes Próprios

A seguir, demonstra-se o funcionamento do sistema com testes próprios, realizados com os autores.

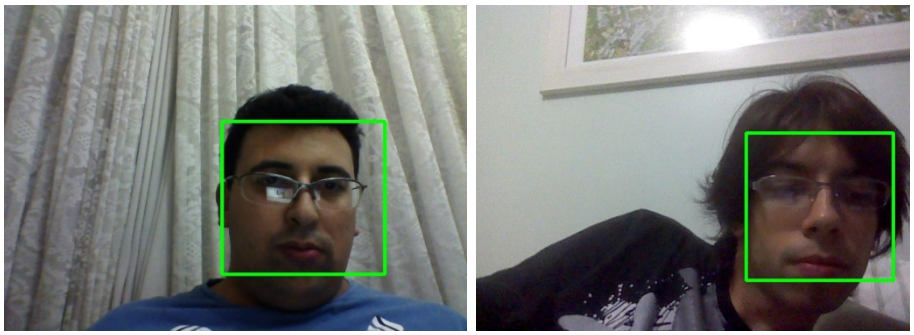


Figura 18 – Detector de faces através de webcam detectando o rosto dos autores, presença de óculos

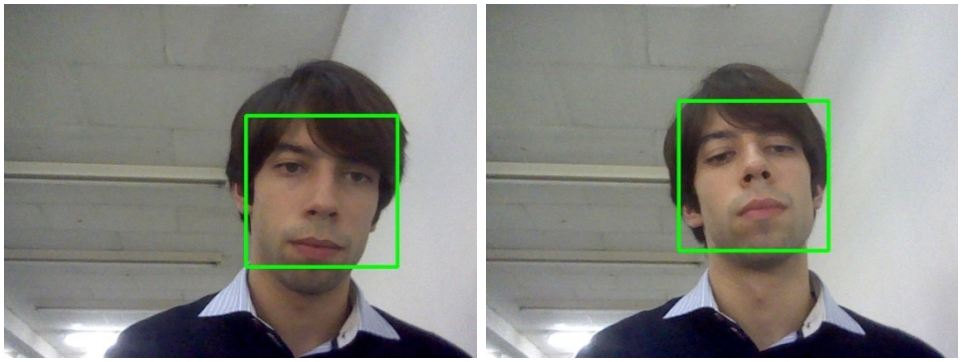
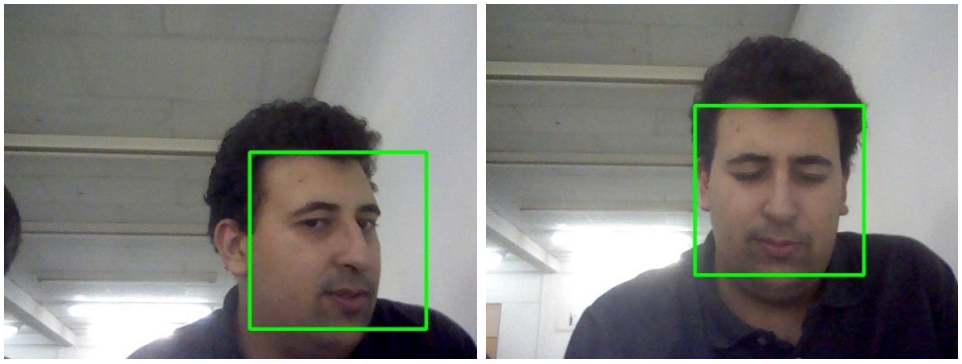


Figura 19 – Detector de faces através de webcam detectando o rosto de um dos autores, cabeça inclinada



(a) (b)

Figura 20 – Detector de faces através de webcam detectando o rosto de um dos autores, cabeça rotacionada (a) e olhos fechados (b)

O módulo de detecção de faces funcionou bem na presença de características instáveis (ex: óculos), como pode ser visto pela Figura 18, na presença de certa angulação da cabeça, como pode ser visto pela Figura 19 e na presença de rotação da cabeça e olhos fechados, como pode ser visto pela Figura 20.

4.2.2 Reconhecimento de face

4.2.2.1 Procedimentos experimentais

Neste trabalho, optou-se não utilizar nenhum modelo dinâmico ou índice de similaridade no treino, como mencionado no Estado da Arte, reduzindo assim a quantidade de cálculos realizados. O reconhecimento é realizado diretamente sempre que uma nova face é detectada. O sistema é inicializado sempre que uma nova imagem é adquirida pela câmera e, se detectada uma face em uma imagem, inicia-se o processo de reconhecimento.

Os testes a seguir são complementares aos testes anteriores de detecção, neles utiliza-se os vídeos do segundo conjunto da base de dados do Honda/UCSD Video Database. A fase de treino para reconhecimento foi feita pela seleção manual de 20 faces de cada sujeito, escolhidas dentre as faces detectadas no teste anterior. O conjunto de faces de treino final é constituído de 300 faces, sendo 20 faces de cada um dos 15 sujeitos. O algoritmo de reconhecimento utilizado pelo sistema é o conjunto DCT+PCA. Como nos vídeos de treino, em cada vídeo de teste a pessoa roda e vira sua cabeça na ordem e velocidade de preferência.

4.2.2.2 Resultados obtidos

A seguir, na Tabela 5, demonstra-se o número total de detecções obtidas no vídeo de teste de cada sujeito, o número de reconhecimentos corretos (RC) e o número de

reconhecimentos incorretos (RI), além da taxa de reconhecimentos corretos do sistema em cada caso. A contagem de reconhecimentos foi feita automaticamente pelo sistema, através da introdução de uma variável do tipo ArrayMap, que contém uma *String* com o nome da pessoa como chave e o número de vezes que foi reconhecida como informação adicional. Vale ressaltar que em alguns casos, houveram detecções incorretas, que foram classificadas pelo sistema como algum dos sujeitos e esse número está presente na contagem, pois, em certos casos, ele não fazia diferença significativa no total (Figura 25) e, em outros casos, o nome da pessoa reconhecida era impresso fora da área do vídeo (Figura 26), não sendo possível identificar por inspeção posterior do vídeo.

Sujeito	*Total*	*RC*	*RI*	*RC (%)*
1	190	170	20	89.47
2	251	223	28	88.84
3	490	295	195	60.20
4	163	141	22	86.50
5	310	91	219	29.35
6	903	521	382	57.70
7	575	449	126	78.09
8	713	683	30	95.79
9	521	390	131	74.86
10	334	274	60	82.04
11	425	364	61	85.65
12	260	116	144	44.62
13	773	5	768	0.65
14	259	87	172	33.59
15	408	365	43	89.46

Tabela 5 – Contagem de Reconhecimentos Corretos e Reconhecimentos Incorretos no teste com vídeos do Honda/UCSD Video Database

Exemplificados nas figuras a seguir estão frames com reconhecimentos corretos do sistema (Figura 21, Figura 22) e frames com reconhecimentos corretos do sistema (Figura 23, Figura 24), além de frames com detecções incorretas e falsos positivos (Figura 25, Figura 26).

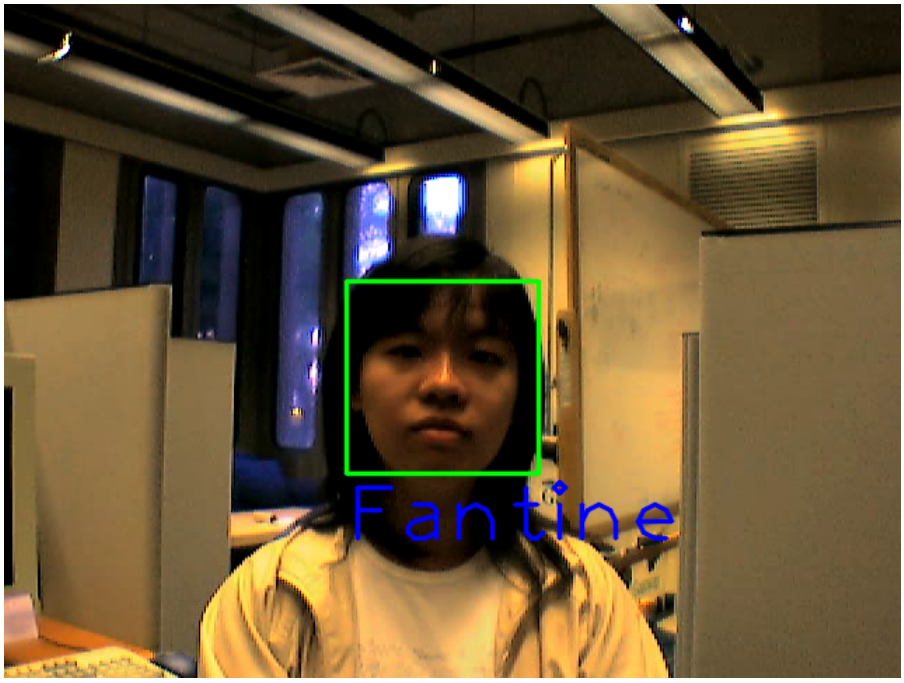


Figura 21 – Reconhecimento Correto do Sujeito 1 - Fantine no teste com vídeos do Honda/UCSD Video Database

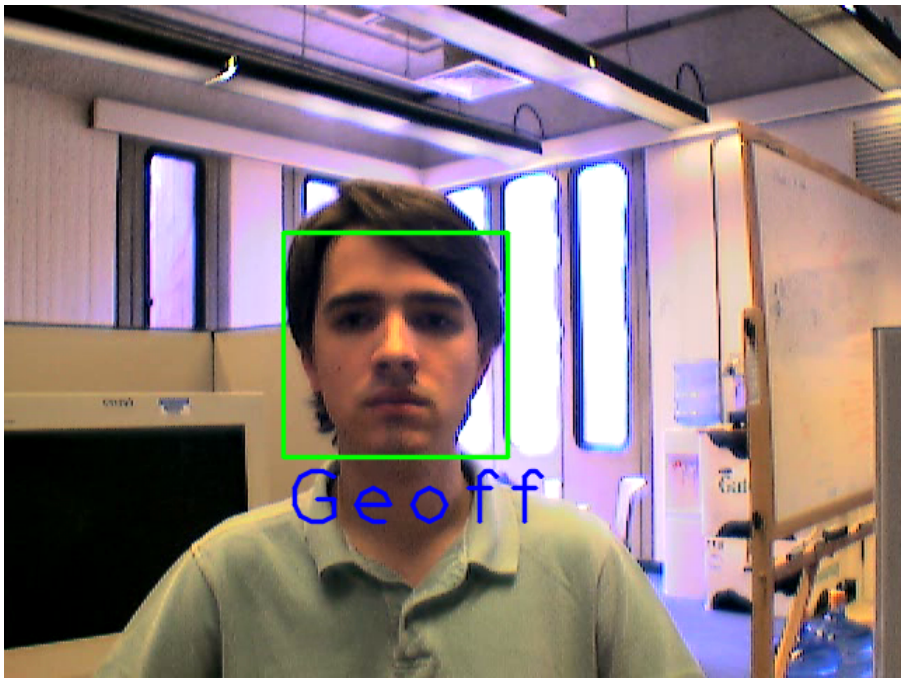


Figura 22 – Reconhecimento Correto do Sujeito 2 - Geoff no teste com vídeos do Honda/UCSD Video Database

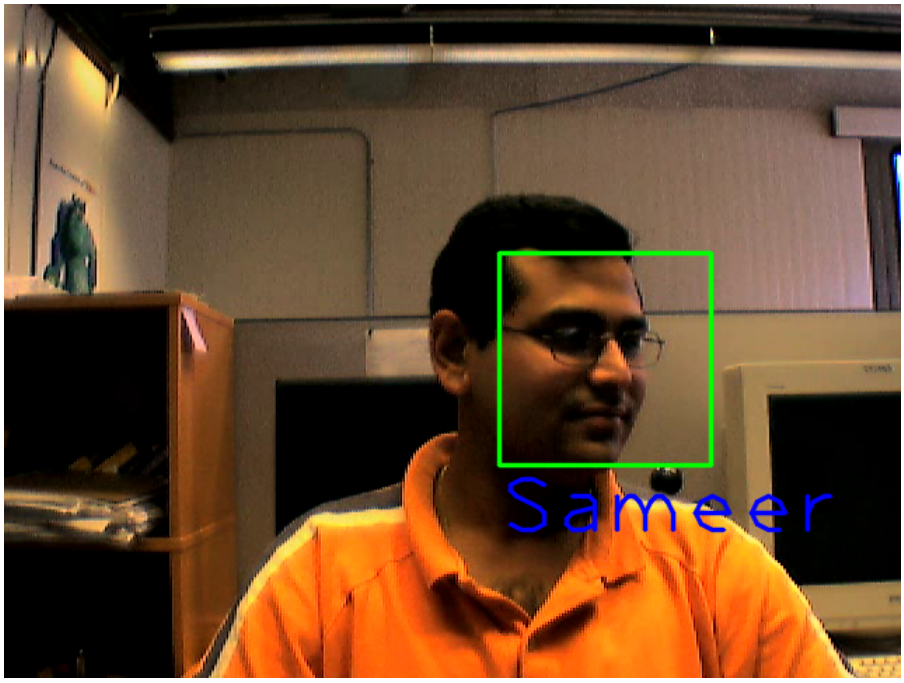


Figura 23 – Reconhecimento Incorreto do Sujeito 13 - Satya no teste com vídeos do Honda/UCSD Video Database

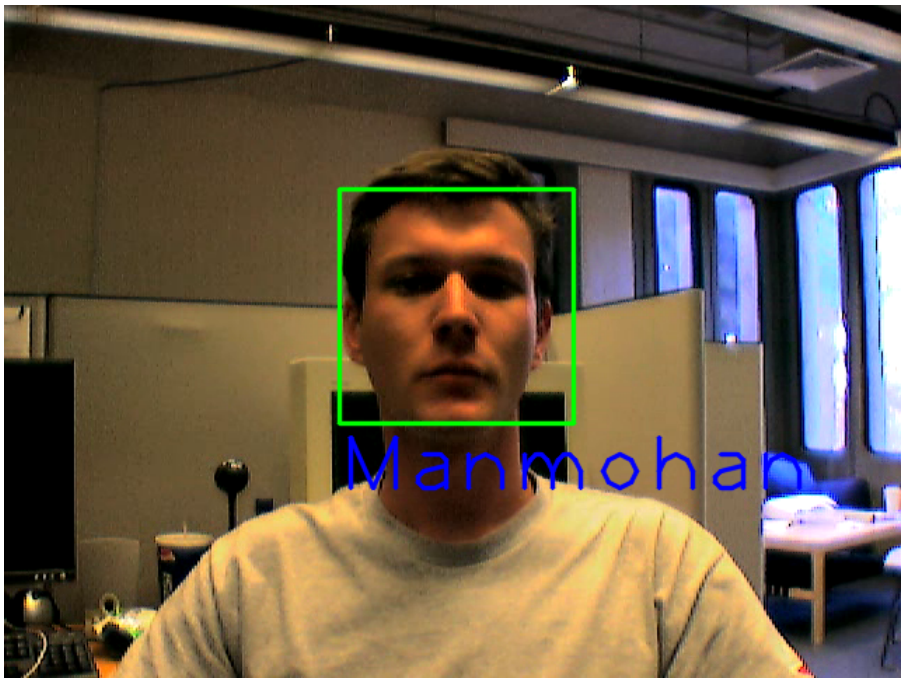


Figura 24 – Reconhecimento Incorreto do Sujeito 14 - Vincent no teste com vídeos do Honda/UCSD Video Database

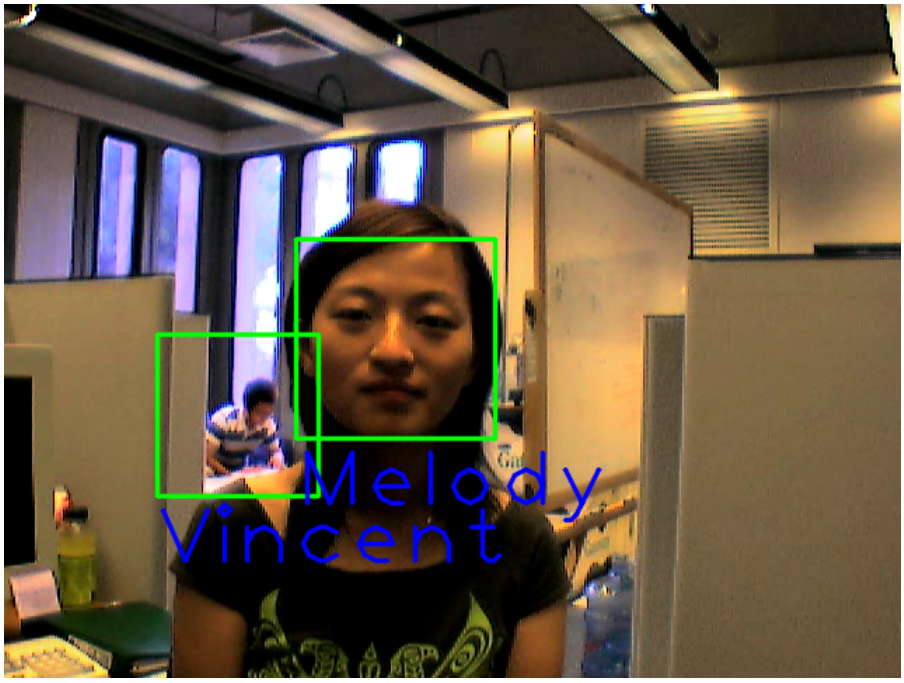


Figura 25 – Detecção Incorreta em frame do Sujeito 9 - Melody no teste com vídeos do Honda/UCSD Video Database

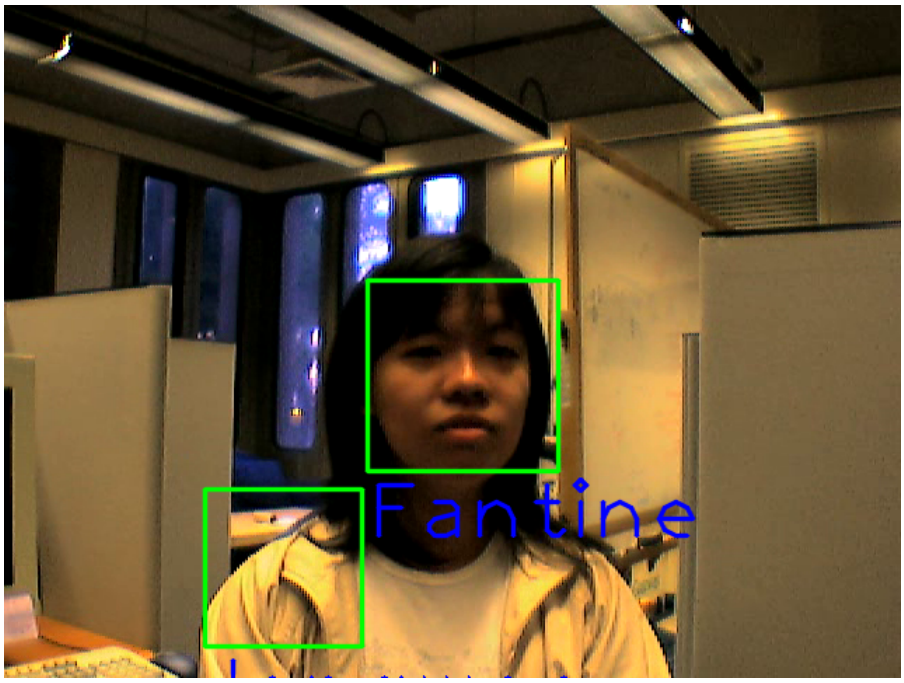


Figura 26 – Detecção Incorreta em frame do Sujeito 1 - Fantine no teste com vídeos do Honda/UCSD Video Database

Através da análise de resultados da Tabela 4, verifica-se que apesar de alguns testes realizados terem apresentado resultados insatisfatórios (sujeitos 5, 12, 13 e 14), no geral, o sistema conseguiu efetuar o reconhecimento correto da maioria dos casos (11 de 15). O método de análise utilizado foi o da contagem de reconhecimentos corretos no tempo total do vídeo (por volta de 50s), o que possibilitaria reconhecer o usuário corretamente em 73% dos casos.

4.3 Execução em Tempo Real

4.3.1 Procedimentos experimentais

O objetivo dos testes de execução é fazer a validação do sistema para utilização em tempo real no robô sociável.

O procedimento experimental utilizou o sistema final, descrito no capítulo 3, com a implementação de agentes *ImageAgent*, *DetectionAgent*, *RecognitionAgent* e *MonitorAgent*. A impressão do tempo inicial foi feita no agente *ImageAgent*, após a captura da frame atual e a impressão do tempo final foi feita no agente *MonitorAgent*, após ter recebido as informações do *DetectionAgent* e do *RecognitionAgent*. A função utilizada em Java foi *Timestamp(date.getTime())*. O banco de dados de treino contém 20 imagens de cada um dos 3 sujeitos (os dois autores e o orientador), ou seja, 60 imagens no total. A amostragem feita utiliza a taxa de 1 frame/s.

4.3.2 Resultados obtidos

Após a captura de diversos tempos, foi calculada uma média de tempo necessário para realizar as tarefas de detecção e reconhecimento, que foi de 620 ms. A diferença mais significativa é observada no tempo de realização da tarefa pela primeira vez, que demora em média 2s. O restante dos tempos se mantém por volta da média calculada,ok entre 540-670ms.

5 Conclusão

Dadas as limitações dos testes feitos com a base de dados de faces publicada pela Universidade de Cambridge (utilização de fotos frontais com um plano de fundo escuro e homogêneo), o sistema se mostrou bastante eficiente e alcançou-se o objetivo esperado na maioria dos testes (reconhecimento correto na ordem de 90%). A solução implementada conseguiu superar algumas dificuldades encontradas em outros trabalhos (presença de características instáveis e de olhos fechados). Observou-se que a solução implementada funciona relativamente bem, tanto com a aplicação somente do PCA quanto com a aplicação do DCT juntamente com o PCA. Não foi possível constatar diferença significativa na utilização do DCT juntamente com o PCA, em relação à utilização somente do PCA. Observou-se ainda que o tamanho da uma base de dados utilizada no treinamento influencia no tempo do teste, o que pode comprometer a fluidez em um reconhecimento em vídeo.

Os testes com vídeo utilizaram o Honda/UCSD Video Database, já utilizado em diversos artigos. Eles foram importantes não só na validação do reconhecimento, mas também da detecção. Verificou-se que o algoritmo Viola-Jones demonstrou-se muito eficiente na taxa total de detecções corretas, que foi de 93,24%. Garante-se com isso uma alta taxa de detecções corretas, atingindo-se assim o resultado esperado. Com relação ao reconhecimento, apesar de alguns testes realizados terem apresentado resultados instatisfatórios (sujeitos 5, 12, 13 e 14), a maioria deles apresentou resultados bastante satisfatórios, de modo que o resultado final, apesar de não ser tão alto quanto os dos testes realizados com imagens, ainda mostra-se adequado para a aplicação em robôs sociáveis. Haveria a possibilidade de reconhecimento correto de usuários em 73% dos casos.

Através dos testes de execução, demonstra-se que o sistema pode ser utilizado em tempo real, para aplicação em robôs sociáveis, pois a média de tempo calculada para realizar as tarefas de detecção e reconhecimento foi de 620 ms.

Referências

ATHITSOS, V. Rectangle filters and adaboost. University of Texas, 2012.

BELLIFEMINE, F.; POGGI, A.; RIMASSA, G. Jade - a fipa-compliant agent framework. *Proceedings of the fourth conference on the practical application of intelligent agents and multi-agent technology*, 1999.

DIGITALOPTICS CORPORATION EUROPE LIMITED. P. Bigioi, E. Steinberg e P. Corcoran. *Face recognition training method and apparatus.* 2013. US 8363951 B2.

BRITANAK, V. The transform and data compression handbook. In: _____. [S.l.: s.n.], 2001. cap. Discrete Cosine and Sine Transforms.

BRITO, R. W. Banco de dados nosql x sgbds relacionais Análise comparativa. *InfoBrasil*, 2010.

CEVIKALP, H.; TRIGGS, B. Face recognition based on image sets. *Computer Vision and Pattern Recognition*, 2010.

CONNOLLY, J. F.; GRANGER, E.; SABOURIN, R. An adptative classification system for video-based face recognition. *Information Sciences*, 2012.

EKENEL, H. et al. Face recognition for smart interactions. In: . [S.l.]: IEEE International Conference on Multimedia and Expo (pp. 1007-1010)., 2007.

FADZIL, M. H. A.; ABU, B. H. Human face recognition using neural networks. In: . [S.l.]: IEEE International Conference on Image Processing (pp. 936-939)., 1994.

FAN, X. et al. The system of face detection based on opencv. In: . [S.l.]: Key Laboratory for Robot & Intelligent Technology of Shandong Province, Shandong University of Science and Technology, 2012.

MICROSOFT CORPORATION. F. O. Folta, Y. He, K. W. Hor, M. G. Shilotri, S. Spears e C. Gu. *Face recognition in video content.* 2013. US 8494231 B2.

HADID, A.; PIETIKAINEN, M. Combining appearence and motion for face and gender recognition from videos. *Pattern Recognition*, 2009.

HEWITT, R. Seeing with opencv - a five-part series. *SERVO Magazine*, 2007.

JENKINS, R.; BURTON, A. M. 100Science, 435, 2008.

JOSHI, K. R. *Graph Visualization Using The NoSQL Database.* Dissertação (Mestrado) — North Dakota State University of Agriculture and Applied Science, 2013.

KAR, S. et al. A multi-algorithmic face recognition system. In: . [S.l.]: International Conference on Advanced Computing and Communications (pp. 321-326)., 2006.

NATIONAL CHIAO TUNG UNIVERSITY. Y-P. Hung P-H. Lee. *Method for face recognition and synthesis.* 2013. US 8483451 B2.

LONE, M. A.; ZAKARIYA, S. M.; ALI, R. Automatic face recognition system by combining four individual algorithms. In: . [S.l.]: International Conference on Computational Intelligence and Communication Systems (pp. 222-226)., 2011.

MIAN, A. Online learning from local features for video-based face recognition. *Pattern Recognition*, 2011.

MORAES, A. F. Método para avaliação da tecnologia biométrica na segurança de aeroportos. 2006.

EASTMAN KODAK COMPANY. Henry c/o EASTMAN KODAK COMPANY Nicponski e Lawrence A. C/O Eastman Kodak Company Ray. *Face detecting and recognition camera and method*. 2001. EP 1128316 A1.

PENTLAND, A.; TURK, M. Eigenfaces for recognition. *Journal Of Cognitive Neuroscience, vol. 3, no. 1 (pp. 71-86).*, 1991.

SONY UNITED KINGDOM LIMITED. R. M. S. Porter, R. Rambaruth, S. D. Haynes e C. H. Gillard. *Video camera for face detection*. 2013. US 8384791 B2.

ROBINSON, I.; WEBBER, J.; EIFREM, E. *Graph Databases*. [S.l.]: O'Reilly Media, 2013.

ROJAS, R. Adaboost and the super bowl of classifiers: A tutorial introduction to adaptive boosting. Freie Universitat, 2009.

VIOLA, P.; JONES, M. J. Robust real time face detection. *International Journal Of Computer Vision*, 2004.

WILSON, P. I.; FERNANDEZ, J. Facial feature detection using haar classifiers. Journal Of Circuits, Systems and Computers, 2006.

WU, L.; WU, P.; MENG, F. A fast face detection for video sequences. In: . [S.l.]: Second International Conference on Intelligent-Human Machine Systems and Cybernetics, 2010.

WU, T. et al. New method using feature level image fusion and entropy component analysis for multimodal human face recognition. In: . [S.l.]: Elsevier, 2012.

XUAN, L. H.; NITSUWAT, S. Face recognition in video, a combination of eigenface and adaptative skin-color model. In: . [S.l.]: IEEE Malaysia, 2007.

ZHANG, P. A video-based face detection and recognition system using cascade face verification modules. In: . [S.l.]: Applied Imagery Pattern Recognition Workshop (pp. 1-8)., 2008.

ZHAO, M.; CHUA, T. S. Markovian mixture face recognition with discriminative face alignment. In: . [S.l.]: IEE International Conference on Automatic Face and Gesture Recognition (pp. 1-6)., 2008.