

FUNDAÇÃO DE ENSINO “EURÍPIDES SOARES DA ROCHA”
CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA - UNIVEM
CURSO DE CIÊNCIA DA COMPUTAÇÃO

GEYSLER NICLEVICZ DA SILVA

**ESTUDO DAS TÉCNICAS PCA (ANÁLISE DE COMPONENTES
PRINCIPAIS) E AUTO-FACES APLICADAS AO RECONHECIMENTO
DE FACES HUMANAS**

MARÍLIA
2009

GEYSLER NICLEVICZ DA SILVA

ESTUDO DAS TÉCNICAS PCA (ANÁLISE DE COMPONENTES
PRINCIPAIS) E AUTO-FACES APLICADAS AO RECONHECIMENTO DE
FACES HUMANAS

Trabalho de Conclusão de Curso
apresentado ao Curso de Bacharelado em
Ciência da Computação da Fundação de
Ensino “Eurípides Soares da Rocha”,
mantenedora do Centro Universitário
Eurípides de Marília – UNIVEM, como
requisito parcial para obtenção do grau de
Bacharel em Ciência da Computação

Orientador:
Prof. Ms. Ricardo Petruzza do Prado

Marília
2009



CENTRO UNIVERSITÁRIO EURÍPIDES DE MARÍLIA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

TRABALHO DE CONCLUSÃO DE CURSO – AVALIAÇÃO FINAL

Geysler Niclevicz da Silva

**ESTUDO DAS TÉCNICAS PCA (PRINCIPAL COMPONENT ANALYSIS) E EIGENFACES
APLICADAS AO RECONHECIMENTO DE FACES HUMANAS**

Banca examinadora da monografia apresentada ao Curso de Bacharelado em Ciência da Computação do UNIVEM/F.E.E.S.R., para obtenção do Título de Bacharel em Ciência da Computação.

Nota: 8.0 (OITO)

Orientador: Ricardo Petruzza do Prado

1º. Examinador: Emerson Alberto Marconato

2º. Examinador: Leonardo Castro Botega

Three handwritten signatures in blue ink are present, each written over a horizontal line. The top signature is the largest and most stylized. The middle signature is smaller and more legible. The bottom signature is the smallest and most legible.

Marília, 04 de dezembro de 2009.

*A Deus pelo carinho, cuidado e
amor que ele tem por todos nós;*

À minha família pelo apoio;

*À minha amada pelo apoio,
por acreditar em mim e pela
companhia nas noites em claro*

AGRADECIMENTOS

Agradeço primeiramente a Deus pelo dom da vida;

A minha família pelo apoio incondicional e paciência;

A minha amada pelo apoio, paciência, incentivo e pelo ombro amigo;

Ao Prof. Ms. Ricardo Petruzza do Prado pela orientação e atenção nas horas mais necessárias;

Ao amigo Guilherme Kami pelas dicas com o Visual Studio e com as linguagens C e C++;

Ao Prof. Leonardo Castro Botega pela atenção e dicas de referências bibliográficas;

A todos os outros companheiros que, direta ou indiretamente, contribuíram para a conclusão deste trabalho.

*Omnia possum in eo
qui me confortat.*

*Tudo posso naquele
que me fortalece.*

Bíblia Sagrada - Filipenses 4.13

SILVA, Geysler Niclevicz da. Estudo da técnica PCA (Análise de Componentes Principais) e Auto-faces aplicadas ao reconhecimento de faces humanas. 2009. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Centro Universitário Eurípides de Marília, Fundação de Ensino “Eurípides Soares da Rocha”, Marília, 2009.

RESUMO

Reconhecimento de padrões é uma área muito pesquisada atualmente, que consiste em utilizar métodos de extração de características de imagens digitalizadas, reduzir a dimensionalidade dos dados gerando um vetor com as características obtidas.

O vetor de características consiste em um vetor n-dimensional contendo informações numéricas de uma imagem, ou de parte dela, descrevendo suas características mais relevantes.

Uma técnica muito utilizada para extração dessas características é a PCA (Análise de Componentes Principais). A PCA é uma das técnicas mais simples do ponto de vista de implementação e proporciona resultados muito satisfatórios na compressão dos dados com pouca perda de informação relevante. Para representar as informações que são utilizadas para o reconhecimento de faces foi escolhida a técnica de Auto-faces devido ao seu bom desempenho já comprovado em vários trabalhos de pesquisa.

O objetivo de um sistema de reconhecimento de faces humanas é utilizar uma imagem de consulta e pesquisar, dentro de um banco de dados de imagem, a mais semelhante a ela.

Este trabalho descreve uma aplicação teste de reconhecimento de faces humanas implementado na ferramenta Matlab utilizando a técnica PCA para extração das características das imagens e Auto-faces para o reconhecimento facial.

Após a implementação da aplicação de teste no Matlab a mesma técnica é implementada no Microsoft Visual Studio 2008 com o auxílio da biblioteca OpenCV onde desenvolve-se o WhoRU, sistema de reconhecimento facial com características de um sistema que possa ser utilizado em um ambiente real.

Palavras-chave: Reconhecimento de Padrões, Reconhecimento facial, PCA, Auto-faces, OpenCV.

SILVA, Geysler Niclevicz da. Estudo da técnica PCA (Análise de Componentes Principais) e Auto-faces aplicadas ao reconhecimento de faces humanas. 2009. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Centro Universitário Eurípides de Marília, Fundação de Ensino “Eurípides Soares da Rocha”, Marília, 2009.

ABSTRACT

Pattern recognition is a field researched mostly, which is use features extraction methods on digital images, reduce the dimensionality of the data and create a vector containing the obtained features.

The features vector is a n-dimensional vector which contains numbers representing a image or a part of it and those numbers describes the most relevant features of image.

There is a method used for extract those features called PCA (Principal Component Analysis). PCA is one of most simple methods to implement and it gives great results compressing data and without loss relevant information. It has been chosen a method called Eigenface for represents information used for face recognition because your great performance proved in so many research works.

The goal of a face recognition system is get an input image and compare it with others images in a database and get the most similar image.

This work describes a test application of face recognition implemented in Matlab with PCA for features extractions and Eigenface for face recognition.

After, the PCA was implemented in Microsoft Visual Studio 2008 with OpenCV Library. The face recognition implemented was called WhoRU which have features of software that can be run on a real environment.

Keywords: Pattern Recongnition, Face Recognition, PCA, Eigenfaces, OpenCV.

Lista de Ilustrações

Figura 1 - Representação da face	20
Figura 2 - Criação da base de dados e cálculo da face média	21
Figura 3 - Subtração da face média e cálculo da matriz de covariância	22
Figura 4 - Cálculo dos Auto-vetores, Auto-valores e projeção das faces	22
Figura 5 - Extração das características da imagem de entrada.....	23
Figura 6 - Cálculo das distâncias.....	23
Figura 7 - Resultado da busca	24
Figura 8 - Sintaxe do método cvCalcEigenObjects.....	27
Figura 9 - Declaração de função C/C++ no C#.....	27
Figura 10 - Janela principal do WhoRU.....	28
Figura 11 - Leitura das imagens	29
Figura 12 - Definição do subespaço da face.....	29
Figura 13 - Projeção das imagens no subespaço	29
Figura 14 - Seleção da imagem de entrada.....	30
Figura 15 - Processo de reconhecimento da face	31
Figura 16 - Exibição da imagem encontrada.....	31
Figura 17 - Captura de imagem.....	32

SUMÁRIO

INTRODUÇÃO.....	10
CAPÍTULO 1 - RECONHECIMENTO DE PADRÕES	12
1.1 Redução da dimensionalidade	12
1.2 Biometria	13
1.2.1 Sistemas Biométricos.....	13
1.2.2 Reconhecimento Facial.....	14
1.2.3 Técnicas de reconhecimento facial	15
CAPÍTULO 2 - PCA (ANÁLISE DE COMPONENTES PRINCIPAIS).....	17
2.1 Cálculo da PCA	17
2.1.1 Fase de treinamento	17
2.1.2 Fase de reconhecimento	19
2.2 O que acontece na prática?	19
2.3 Considerações finais	20
CAPÍTULO 3 - MATLAB	21
3.1 Aplicação teste	21
3.1.1 Treinamento	21
3.1.2 Reconhecimento.....	23
3.1.3 Migração	24
CAPÍTULO 4 - WHORU.....	25
4.1 Linguagem C#.....	25
4.2 OpenCV (Open Computer Vision Library)	25
4.3 Emgu CV	26
4.3.1 Utilização do Emgu CV no WhoRU.....	26
4.4 Desenvolvimento do WhoRU	28
4.4.1 Treinamento	28
4.4.2 Reconhecimento.....	30
Conclusão	33
REFERÊNCIAS	34

INTRODUÇÃO

A habilidade que o ser humano possui de reconhecer faces é muito interessante, principalmente pelo fato desse processo ocorrer de forma instantânea mesmo sob as mais diversas alterações que a face pode sofrer como barba, óculos, expressões de emoção, ângulo de visão, baixa luminosidade, ou até mesmo quando são exibidas partes da face.

O reconhecimento de face é uma das características mais utilizadas pelo ser humano para se relacionar uns com os outros. Analisando a face de uma pessoa conseguimos saber se ela está alegre, triste, insatisfeita, com sono, entre outras. Entretanto, reproduzir essa habilidade humana para um sistema computacional é uma tarefa muito complexa, pois, ainda não é conhecido de forma exata o mecanismo que o cérebro utiliza para fazer o reconhecimento facial com tanta rapidez e precisão, mas vários trabalhos de pesquisa já propuseram soluções com resultados muitos satisfatórios (TURK, PENTLAND, Março 1991).

Um dos principais motivos de se estudar modelos computacionais para o reconhecimento de faces é a necessidade que existe atualmente de identificar os indivíduos dentro dos sistemas de informação. A facilidade de comunicação propiciada pelos computadores e internet causou um grande avanço em sistemas de comunicação que transmitem dados de operações financeiras, comerciais, logística e até mesmo de comunicação pessoal, mas surgiram também os problemas em relação à segurança na transmissão e armazenamento dessas informações como, por exemplo, fraudes com cartões de crédito, invasão de computadores, crimes de falsidade ideológica. Houve, então, uma grande necessidade de identificação dos indivíduos nos sistemas de informação para aumentar a segurança das informações mantidas por esses sistemas. Essa necessidade está presente tanto em ambientes físicos como bancos, bibliotecas, empresas, quanto em ambientes virtuais como sites de comércio eletrônico, sites de bancos, email, sites com conteúdo privado no qual é permitido o acesso apenas por pessoas autorizadas.

Essa tecnologia que permite a identificação verdadeira de um indivíduo é baseada em uma área chamada Biometria (LIN, 2000).

No Capítulo 1 desse trabalho é abordado o reconhecimento de padrões descrevendo o problema de redução de dimensionalidade e a biometria com foco no reconhecimento facial.

O Capítulo 2 descreve o PCA (Análise de Componentes Principais), uma das técnicas utilizadas em sistemas de reconhecimento facial e como é feito todo o processamento da imagem nessa técnica.

O Capítulo 3 demonstra uma implementação do PCA no Matlab, um software que oferece uma série de ferramentas para cálculos matemáticos complexos com foco em matrizes, sendo um ótimo ambiente para experimentos.

O Capítulo 4 descreve o WhoRU, um sistema de reconhecimento facial no qual foi implementado o PCA utilizando-se uma biblioteca para a linguagem C++ chamada OpenCV, a qual oferece uma série de recursos que facilitam a implementação de sistemas que utilizam o processamento de imagem de uma forma geral.

CAPÍTULO 1 - RECONHECIMENTO DE PADRÕES

O reconhecimento de padrões é o desenvolvimento de meios teóricos e computacionais para classificar objetos abstratos em categorias (THEODORIDIS, KOUTROUBAS, 2006). São considerados exemplos de padrão uma impressão digital, uma assinatura, uma face, um sinal de voz.

Existem dois tipos de classificação de padrões: classificação supervisionada e classificação não supervisionada (JAIN, DUIN, MAO, 2000). Na classificação supervisionada o padrão de entrada é classificado como um membro de uma classe predefinida. Na classificação não supervisionada a entrada é associada a uma classe inicialmente desconhecida.

O processo de reconhecimento de padrões envolve três etapas essenciais:

- Aquisição dos dados (extração das características) e pré-processamento (seleção das características mais importantes);
- Representação dos dados;
- Tomada de decisão.

Um problema de reconhecimento de padrões bem definido é aquele que possui uma pequena variação intra-classes e uma grande variação inter-classes, resultando em uma representação compacta do padrão. Dentre as abordagens sobre reconhecimento de padrões podemos citar: casamento de modelos, classificação estatística, casamento sintático ou estrutural e redes neurais (JAIN, DUIN, MAO, 2000).

1.1 Redução da dimensionalidade

Um dos grandes problemas encontrados no reconhecimento de padrões está relacionado à dimensionalidade dos dados que, geralmente, são valores representados em matrizes n -dimensionais.

Na maior parte dos casos a quantidade de informação adquirida é muito grande, podendo tornar inviável a implementação do reconhecimento de padrões desses dados. Por esse motivo, faz-se necessário o estudo de métodos para reduzir a dimensionalidade dos dados transformando os dados, que inicialmente pertencem a um espaço n -dimensional, em um espaço k -dimensional, tendo k um tamanho bem menor em relação à n .

O problema da redução de dimensionalidade é que isso causa uma perda de dados que podem ser importantes na discriminação dos padrões. Por esse motivo, é muito

importante que a técnica de redução de dimensionalidade consiga preservar o máximo de informação relevante possível.

1.2 Biometria

A biometria parte do princípio que o seu corpo é a sua senha. Biometria é a ciência que estuda estabelecer a identidade de um indivíduo baseadas em suas características físicas, químicas ou comportamentais (JAIN, FLYNN, ROSS, 2008).

Existem várias aplicações para a biometria. Alunos do curso obrigatório para emissão da Carteira Nacional de Habilitação os alunos recebem a presença no curso pela identificação da digital (POGGUETTO, 2009).

Existe também o projeto STID (Soluções de Telecomunicações para Inclusão Digital) financiado pelo Funttel que utiliza o reconhecimento facial para autenticar usuários no sistema de agendamento de consulta médica nos postos de saúde da cidade de Bastos – SP (BASTOS, 2009).

O reconhecimento facial aplicado no projeto STID (BASTOS, 2009) é desenvolvido pela empresa Cognitec, a qual possui a patente da tecnologia FaceVACS (COGNITEC, 2009) desenvolvida desde 1995. A tecnologia FaceVACS é utilizada em várias soluções biométricas desenvolvidas pela Cognitec. Um dos casos de sucesso da empresa é a implantação do Lobby Watch (NETITECH, 2009) na Torre do Shopping Rio Sul, um dos condomínios comerciais mais sofisticados e movimentados do Rio de Janeiro. Entre os condôminos da Torre do Shopping Rio Sul estão grandes bancos como Bradesco, Itaú e Banco do Brasil e grandes empresas como HP, Brascan, clínicas e laboratórios. O problema no local era o longo tempo de espera para o cadastramento na entrada do edifício devido ao grande volume de visitante do local que é em torno de 3500 a 4000 pessoas por dia. Após a implantação do sistema LobbyWatch o tempo de espera diminuiu drasticamente.

1.2.1 Sistemas Biométricos

Sistemas biométricos capturam amostras de características do corpo humano com a finalidade de identificá-lo. Essas amostras, que podem ser íris, retina, dedo, rosto, veias da mão, voz, modo de andar (JAIN, FLYNN, ROSS, 2008), são colhidas e transformadas em padrão para comparações em futuras identificações.

Pela necessidade de sistemas robustos de reconhecimento humano em aplicações críticas como controle de acesso seguro, cruzamento de fronteiras internacionais e imposições da lei a biometria se mostra como uma tecnologia viável que pode ser aplicada em sistemas de manipulação de identidade em larga escala (JAIN, FLYNN, ROSS, 2008).

Um método muito eficiente é o reconhecimento da íris, mas trata-se de um método muito intrusivo em relação aos demais, pois, depende que o usuário posicione os olhos em frente a um scanner ou câmera para que a leitura seja realizada e esse procedimento, para muitas pessoas, acaba sendo um tanto desconfortável.

Outro método biométrico que possui uma aceitação maior por parte dos usuários e que tem certa vantagem em relação aos outros é o reconhecimento facial, pois, é um método pouco intrusivo havendo a necessidade apenas de que a pessoa se posicione em frente a uma câmera, o que acaba não sendo tão desconfortável em relação ao reconhecimento da íris.

Por conta das vantagens do reconhecimento facial existem numerosos estudos sobre essa técnica, mas esses estudos mostram que reproduzir a capacidade humana de reconhecer faces é muito mais complexo que se imaginava e a criação de um modelo matemático que reproduza essa capacidade humana não uma tarefa das mais simples.

1.2.2 Reconhecimento Facial

O reconhecimento facial é um sistema biométrico que faz a identificação de um indivíduo pela face.

Os primeiros trabalhos desenvolvidos no reconhecimento facial por computador necessitavam de uma marcação manual dos pontos de localização das características da face como posição da boca, distância entre os olhos, altura do nariz em relação ao queixo. A partir daí eram usadas regras de classificação para identificar cada característica.

Pelo fato da marcação ser manual, o sistema não encontrava problemas quando a imagem apresentava variações como, rotação da cabeça, intensidade da luz, qualidade da imagem etc. Tempos depois outros trabalhos semelhantes foram desenvolvidos, mas sem a intervenção humana para localização da face em uma imagem (CHELLAPPA, SOROHEY, WILSON, BARNES, 1994).

1.2.3 Técnicas de reconhecimento facial

Em um processo de reconhecimento facial as imagens, que são consideradas uma amostra com alta dimensionalidade, passam por um processo de redução de dimensionalidade para que essa nova amostra com um número de dimensões mais reduzido passe por um dos processos de decisão conhecido como classificadores.

Após a redução da dimensionalidade aplica-se um método classificador, pois, é ele que vai determinar qual imagem é mais parecida com a imagem testada. Existem várias técnicas de classificação sendo as mais conhecidas Redes Neurais, Distância Euclidiana, Classificador Bayesiano e Distância de Mahalanobis.

Redes Neurais são elementos computacionais não lineares chamados neurônios organizados como uma rede a qual armazena a forma como os neurônios estão interconectados. O interesse pelo estudo das Redes Neurais iniciou por volta da década de 1940 como o trabalho proposto por McCulloch e Pitts em 1943 (GONZALEZ, WOODS, 2002).

A Distância Euclidiana pode ser definida como determinar a relação entre pontos em termos de distância e ângulo onde cada ponto representa uma localização ou, de uma forma abstrata, onde cada ponto representa uma entidade que possa ser expressa como um vetor em um espaço Euclidiano finito (Dattorro, 2005).

A Distância de Mahalanobis é uma medida de distância que permite entender a covariância dos dados no espaço no qual esses dados estão situados (BRADSKI, KAEHLER, 2008).

Dentre as técnicas de reconhecimento facial mais conhecidas podemos citar Métodos de correlação, Métodos de decomposição de valor singular, Métodos baseados em Decomposição Karhunen-Loeve (PCA), Métodos baseados em discriminante linear Fisher, Métodos baseados em Cadeias Ocultas de Markov.

Os algoritmos utilizados no reconhecimento facial podem ser classificados em dois tipos: métodos baseados em características e métodos baseados em aparência.

Os métodos baseados em características utilizam as propriedades e a relação geométrica entre os pontos das características da face como descritores para o reconhecimento da face e os métodos baseados em aparência utilizam as propriedades globais do padrão da face como descritores para o mesmo fim (JAIN, FLYNN, ROSS, 2008).

Esse trabalho é um estudo sobre a técnica PCA (Análise de Componentes Principais) aplicada ao reconhecimento facial.

CAPÍTULO 2 - PCA (ANÁLISE DE COMPONENTES PRINCIPAIS)

A técnica PCA (Análise de Componentes Principais) foi descrita inicialmente por Karl Pearson em 1901, pois, ele acreditava que seria a solução ideal para alguns problemas biométricos da época, mas ele não propôs nenhuma implementação dessa técnica para um caso com mais de duas variáveis.

Em 1933 essa técnica foi descrita por Hotteling, mas os cálculos ainda eram complexos quando se tratava de um problema com dezenas de variáveis. Apenas em época mais recente, com o surgimento dos computadores atuais, essa técnica ficou mais difundida.

A PCA tem como objetivo determinar uma transformação linear de um conjunto de n variáveis originais X_1, X_2, \dots, X_n em um novo conjunto de p variáveis Y_1, Y_2, \dots, Y_p de forma que essas últimas sejam descorrelacionadas. Essas novas variáveis Y_1, Y_2, \dots, Y_p precisam ter a mesma variância das variáveis originais e estejam ordenadas de tal forma que $var(Y_1) \geq var(Y_2) \geq \dots \geq var(Y_p)$. Essas novas variáveis Y_p são chamadas Componentes Principais.

Um das principais aplicações da PCA é a redução de dimensionalidade, onde, são retiradas as variáveis originais que possuem menor variância. A variância total é definida por n variáveis, mas essa variância pode ser definida por um número p bem menor de componentes principais.

Por esse motivo a PCA é utilizada em muitas aplicações como pré-processamento das informações servindo, sem perda significativa de informações relevantes, como entrada para os métodos seguintes a ela.

2.1 Cálculo da PCA

2.1.1 Fase de treinamento

Tomando como conjunto de imagens de faces de treinamento X_1, X_2, \dots, X_n . Cada imagem está representada em um vetor unidimensional de ordem n^2

A face média desse conjunto é definida por

$$1) \quad \Psi = \frac{1}{n} \sum_{j=1}^n X_n$$

Agora, deve-se subtrair a face média de cada face do conjunto. Cada face difere de sua média por

$$2) \quad \Phi_n = X_n - \Psi$$

Em seguida devemos calcular a matriz de covariância para definir o subespaço da imagem.

$$3) \quad C = \Phi^T \Phi$$

No próximo passo, identifica-se o espaço face calculando os Auto-vetores e os Auto-valores da matriz de covariância.

Cálculo dos Auto-valores:

$$4) \quad |\Phi - \lambda I| = 0$$

Sendo:

I = matriz identidade da matriz de covariância

λ = Auto-valores

Cálculo dos Auto-vetores:

$$5) \quad \Phi e = \lambda e$$

Sendo:

e = Auto-vetor

λ = Auto-valor

Se apenas os Auto-vetores com os maiores Auto-valores são considerados, a variância total do padrão não muda muito e a dimensionalidade é m sendo $m \ll n$.

No último passo cada imagem de treinamento é projetada no espaço face. O descritor PCA é calculado por uma combinação linear de Auto-vetores com os vetores originais.

$$6) \quad W_n = e_n^T (X - \Phi)$$

Sendo:

$n = 1, 2, \dots, m$

e = Auto-vetores

X = vetor de faces de treinamento

Φ = face média

Para cada face, apenas os coeficientes W são armazenados para futura comparação.

2.1.2 Fase de reconhecimento

O primeiro passo é colocar a imagem de consulta em um vetor Q .

O segundo passo é projetar o vetor de consulta no espaço face (combinação linear de Auto-vetores).

$$1) \quad W_n = e_n^T(Q - \Phi)$$

Sendo:

$n = 1, 2, \dots, m$

$e =$ Auto-vetores

$Q =$ vetor de consulta

$\Phi =$ face média

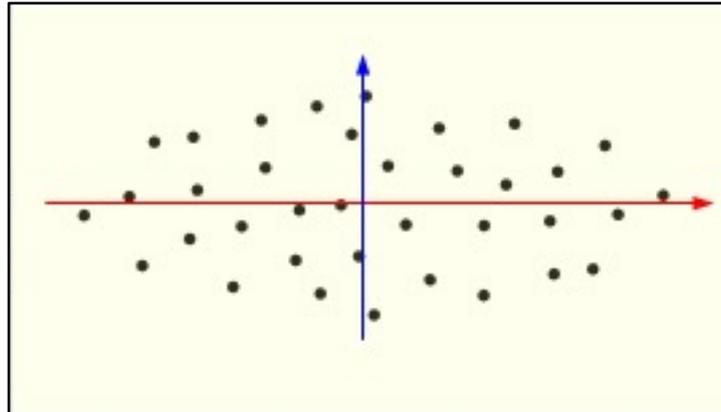
O último passo é comparar a distância entre o descritor do vetor de consulta com cada um dos descritores armazenados na base de dados. Isso pode ser feito com o cálculo da Distância Euclidiana ou Distância de Mahalanobis. O vetor com a menor distância é o mais semelhante.

Um limite máximo para a distância pode ser usado para aceitar ou recusar o descritor encontrado.

2.2 O que acontece na prática?

Demonstrando apenas os cálculos do PCA ainda pode ficar confuso o seu funcionamento, portanto, imagina-se que uma imagem seja representada por uma nuvem de pontos que representam as características da face. A disposição desses pontos possui um formato que define o sentido da variação das características.

Figura 1 - Representação da face



As linhas vermelha e azul na Figura 1 representam os componentes principais da face definindo uma relação entre os pontos. A linha vermelha possui um comprimento maior, pois, a variação das características da face é mais nesse sentido. A linha azul possui um comprimento menor, pois, a variação das características é menor nesse sentido.

Cada face possui a maior variação em um sentido diferente em relação às outras faces. É esse sentido que define a diferença entre as faces, portanto, quando a face testada é comparada com as faces de treinamento é o sentido do maior componente principal que define o nível de semelhança entre as faces.

A redução de dimensionalidade ocorre na projeção desses pontos em cada componente principal. Cada ponto possui uma coordenada de localização e, em uma representação com coordenadas em três dimensões, cada ponto possui três dimensões para que seja definida sua localização. Quando os pontos são projetados em um vetor que representa um componente principal, apenas a localização desse ponto dentro do vetor é armazenada e esse vetor possui apenas uma dimensão.

2.3 Considerações finais

O cálculo descrito no capítulo 2.1 pode ser implementado em várias ferramentas. Um bom exemplo desse tipo de ferramenta é o Matlab, o qual possui uma série de recursos para cálculos matemáticos com foco em matrizes.

CAPÍTULO 3 - MATLAB

Matlab é um ambiente de trabalho interativo no qual o usuário pode executar tarefas computacionais complexas com apenas alguns comandos. Foi desenvolvido inicialmente por Cleve Moler na década de 1970 com a linguagem Fortran, passando, mais tarde, para a linguagem C. No que diz respeito à programação numérica o Matlab possui várias rotinas que permitem o usuário se concentrar nos experimentos. Os resultados podem ser visualizados tanto na forma numérica como na forma de gráficos 2D ou 3D com rapidez e facilidade.

Justamente pela facilidade que o Matlab oferece na execução de cálculos complexos foi possível efetuar uma aplicação para testar a funcionalidade da técnica PCA.

3.1 Aplicação teste

Ao executar o script criado para o reconhecimento é solicitado ao usuário indicar a localização das imagens de treinamento. Em seguida é solicitado ao usuário indicar a imagem que será testada para, então, iniciar o processo de extração das informações das imagens de treinamento.

3.1.1 Treinamento

Inicialmente cria-se a base de dados e em seguida calcula-se a face média como demonstrado na Figura 2.

Figura 2 - Criação da base de dados e cálculo da face média

```
fprintf('Criando base de dados para consulta\n');

T = [];
for i = 1 : Train_Number
    str = strcat(TrainDatabasePath,'\ ',FilesNames(i).name);
    img = imread(str);
    img = rgb2gray(img);
    [irow icol] = size(img);
    %Tranformando imagens 2D em vetores de imagens 1D
    temp = reshape(img',irow*icol,1);
    T = [T temp]; % 'T' cresce a cada turno
end

fprintf('Base de consulta criada\n');
fprintf ('Calculando o vetor médio\n');
m = mean(T,2);
Train_Number = size(T,2);
```

O próximo passo é a subtração da face média de cada imagem e esse resultado é utilizado para o cálculo da matriz de covariância.

Figura 3 - Subtração da face média e cálculo da matriz de covariância

```
fprintf ('Criando matriz de covariância\n');
%Para obter a matriz de covariancia é necessário subtrair o vetor médio
%dos vetores obtidos pelo mapeamento das faces.
A = [];
for i = 1 : Train_Number
    temp = double(T(:,i)) - m;
    A = [A temp];
end
L = A'*A;
```

Após o cálculo da matriz de covariância os Auto-vetores e Auto-valores são calculados pela função eig() projetando-se as imagens no subespaço.

Figura 4 - Cálculo dos Auto-vetores, Auto-valores e projeção das faces

```
fprintf ('Determinando auto-vetores e auto-valores\n');
[V D] = eig(L);

fprintf ('Ordenando auto-vetores\n');
L_eig_vec = [];
for i = 1 : size(V,2)
    if( D(i,i)>1 )
        L_eig_vec = [L_eig_vec V(:,i)];
    end
end

fprintf('Criando espaço face\n');

ProjectedImages = [];
Eigenfaces = A * L_eig_vec;
Train_Number = size(Eigenfaces,2);
for i = 1 : Train_Number
    temp = Eigenfaces'*A(:,i);
    ProjectedImages = [ProjectedImages temp];
end
```

3.1.2 Reconhecimento

A fase de reconhecimento inicia-se executando o mesmo procedimento de treinamento com a imagem que será testada. Esse processo é exibido na Figura 5.

Figura 5 - Extração das características da imagem de entrada

```
InputImage = imread(TestImage);
temp = InputImage(:,:,1);

[irow icol] = size(temp);
InImage = reshape(temp', irow*icol, 1);
Difference = double(InImage)-m;
ProjectedTestImage = Eigenfaces'*Difference;
```

A última etapa do processo de reconhecimento é o cálculo da distância entre a matriz de características da imagem de entrada e as matrizes de características das imagens de treinamento.

Figura 6 - Cálculo das distâncias

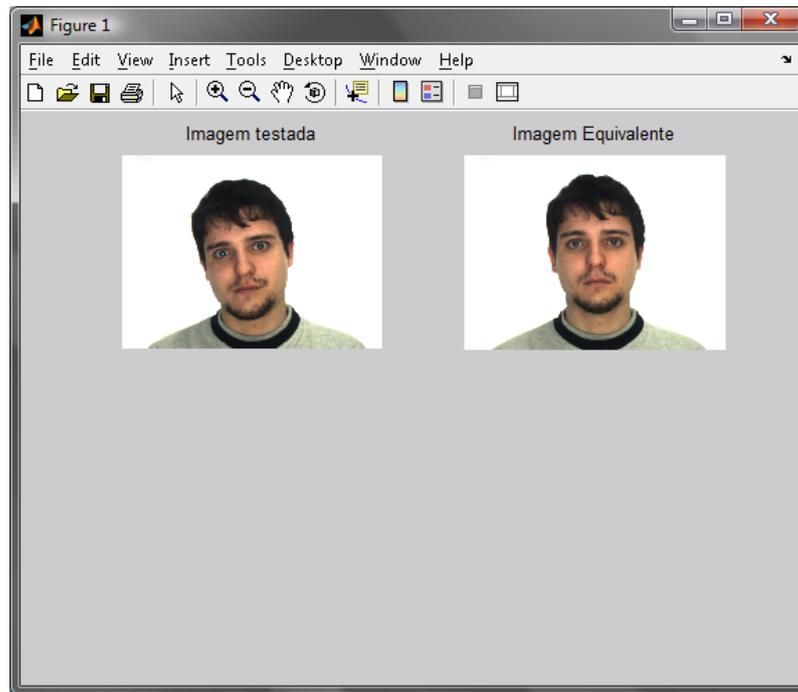
```
fprintf('Calculando distância entre as imagens\n');
Euc_dist = [];
for i = 1 : Train_Number
    q = ProjectedImages(:,i);
    temp = ( norm( ProjectedTestImage - q ) )^2;
    Euc_dist = [Euc_dist temp];
end

[Euc_dist_min , Recognized_index] = min(Euc_dist);
OutputName = FileNames(Recognized_index).name;

SelectedImage = strcat(TrainDatabasePath, '\\', OutputName);
SelectedImage = imread(SelectedImage);
```

Após encontrar a imagem mais próxima a mesma é exibida para o usuário como demonstrado na Figura 7.

Figura 7 - Resultado da busca



3.1.3 Migração

Como demonstrado na seção 3.1.2 o Matlab oferece ferramentas que facilitam a implementação da PCA, mas o Matlab é apenas um ambiente para experimentos matemáticos sendo difícil a utilização de um sistema de reconhecimento facial por um usuário comum, pois, a interface gráfica que é possível ser construída no Matlab não é muito amigável a ponto de um usuário comum se sentir confortável ao utilizá-lo. Outro ponto a ser considerado é a possibilidade de desenvolver uma aplicação portátil para diferentes plataformas como um dispositivo móvel. Diante dessa situação, surge a necessidade de criar uma aplicação com características de um sistema com uma interface mais amigável e que pudesse ser aplicado em um ambiente real. Pensando nisso foi efetuado um estudo sobre alguma ferramenta que pudesse ser utilizada em tal implementação. Surge, então, o WhoRU, uma aplicação desenvolvida no Microsoft Visual Studio 2008.

O Microsoft Visual Studio 2008 é uma plataforma de desenvolvimento que possui suporte para as linguagens C/C++ e C#. A possibilidade de desenvolver um sistema utilizando uma plataforma de desenvolvimento que oferece esse suporte às linguagens C/C++ e C# foi muito útil, pois, utilizando-se uma biblioteca de visão computacional chamada OpenCV possibilitou a implementação das funções responsáveis pelo processamento das imagens e a linguagem C# possibilitou a criação de uma interface gráfica amigável.

A implementação do WhoRU é explicada no capítulo 4 desse trabalho.

CAPÍTULO 4 - WHORU

Visando demonstrar o funcionamento da PCA foi desenvolvido o WhoRU utilizando a linguagem C# e C/C++.

Para os cálculos matemáticos envolvidos na PCA utilizou-se uma biblioteca para linguagem C/C++ chamada OpenCV.

4.1 Linguagem C#

A linguagem C# é uma linguagem de programação orientada a objetos, simples, e poderosa que surgiu com a criação da plataforma de desenvolvimento Microsoft .NET (dot net).

A plataforma .NET oferece suporte ao desenvolvimento de aplicações desktop e web com a possibilidade de fácil integração entre as aplicações.

Nesse trabalho utilizou-se o Microsoft Visual Studio 2008 com a linguagem C# para a criação da interface gráfica do WhoRU, pois, o Microsoft Visual Studio 2008 oferece ferramentas para um rápido desenvolvimento de aplicações economizando tempo para o programador por haver muitos componentes prontos de fácil utilização.

4.2 OpenCV (Open Computer Vision Library)

OpenCV é uma biblioteca de visão computacional escrita em C e C++ e é compatível com as plataformas Linux, Windows e MacOS X. Também há o desenvolvimento de interfaces para Python(BRADSKI, KAEHLER, 2008).

Foi desenvolvida por um grupo de pesquisadores da Intel para funcionar com eficiência em aplicações de tempo real e possui suporte para melhorar o desempenho quando utilizada em equipamentos com processador de múltiplos núcleos. Quando deseja-se uma otimização automática em arquiteturas Intel pode-se utilizar o OpenCV combinado com outra biblioteca, também da Intel, chamada Integrated Performance Primitives (IPP), a qual possui rotinas otimizadas em baixo-nível. A OpenCV utiliza-se automaticamente da rotina mais apropriada quando a IPP está instalada.

Um dos objetivos da OpenCV é fornecer uma estrutura para o desenvolvimento rápido de aplicações sofisticadas contendo mais de quinhentas funções que englobam várias

áreas da visão computacional como imagens médicas, segurança, interface de usuário, calibragem de câmeras, visão estéreo, robótica entre outras.

Pela relação existente entre a visão computacional e a aprendizagem de máquina a OpenCV também possui uma biblioteca chamada Machine Learning Library (MLL). Essa sub-biblioteca possui funções voltadas ao uso em aplicações de reconhecimento e classificação estatísticos de padrões.

4.3 Emgu CV

O Emgu CV é um wrapper para a plataforma Microsoft .NET permitindo que as funções do OpenCV sejam executadas a partir das linguagens suportadas pela plataforma Microsoft .NET como C#, VB .NET e Visual C++.

O Emgu CV também pode ser compilado no Mono, uma alternativa ao Microsoft Framework .NET para os sistemas Linux e Mac OS X que permite a execução de softwares desenvolvidos na plataforma Microsoft .NET.

4.3.1 Utilização do Emgu CV no WhoRU

Toda a interface gráfica do WhoRU é desenvolvida com a linguagem C#, portanto, o EmguCV é uma opção para a integração do OpenCV, desenvolvido em C/C++, com a linguagem C#.

Durante o desenvolvimento do WhoRU encontra-se alguns problemas de incompatibilidade de tipos. Isso acontece porque o EmguCV não possui um método relacionado com cada função do OpenCV. Algumas funções do OpenCV são executadas através da classe Emgu.CV.CvInvoke. As funções que precisam ser executadas através dessa classe possuem seus parâmetros originais definidos na biblioteca OpenCV. O problema aparece quando o tipo dos parâmetros de alguma função da biblioteca OpenCV não possui um tipo equivalente na linguagem C#, o que gera um erro durante a compilação ou execução do programa.

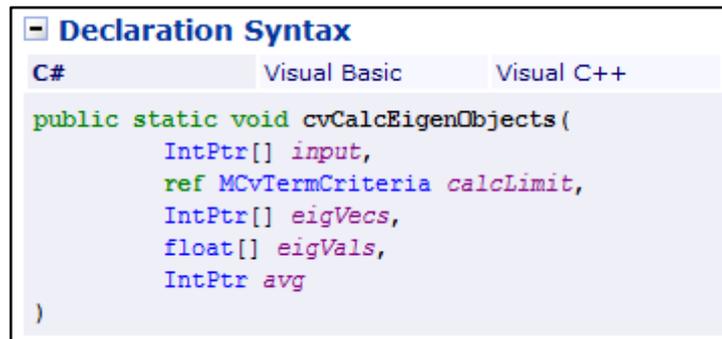
Quando utiliza-se o Emgu CV uma lista de imagens pode ser declarada da seguinte forma:

```
Image<Gray, float>[] imgList = new Image<Gray, float>[50];
```

Nessa declaração utiliza-se um array nativo do C# para declarar uma variável que representa uma lista com várias imagens.

O problema de incompatibilidade de tipos pode ser verificado quando o método `Emgu.CV.CvInvoke.cvCalcEigenObjects()` é executado. De acordo com a documentação do EmguCV o método `Emgu.CV.CvInvoke.cvCalcEigenObjects()` possui o formato exibido na Figura 8.

Figura 8 - Sintaxe do método `cvCalcEigenObjects`



```

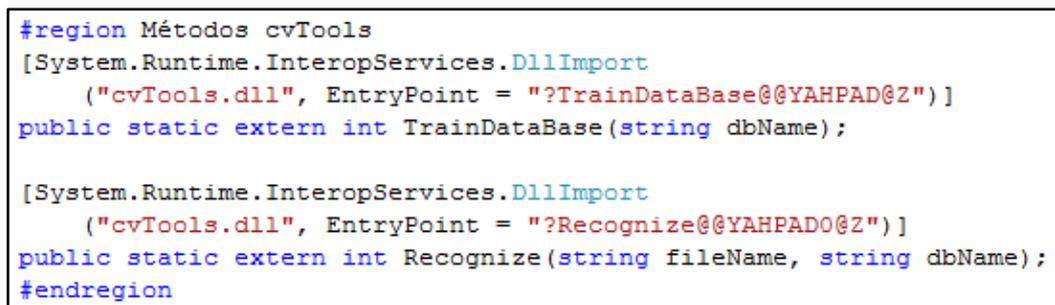
Declaration Syntax
C#      Visual Basic      Visual C++
public static void cvCalcEigenObjects(
    IntPtr[] input,
    ref MCvTermCriteria calcLimit,
    IntPtr[] eigVecs,
    float[] eigVals,
    IntPtr avg
)
  
```

O primeiro parâmetro desse método deve ser um array de elementos do tipo `IntPtr`, o que significa uma lista de ponteiros de várias imagens, mas em C# a declaração de um array de elementos do tipo `Image<Gray, float>` não possui uma propriedade ou método que retorne um array de elementos `IntPtr` de cada imagem que faz parte desse conjunto, portanto, não é possível utilizar a variável `imgList`, declarada anteriormente, como primeiro parâmetro do método `Emgu.CV.CvInvoke.cvCalcEigenObjects()`.

Para contornar esse problema toda a implementação das operações de processamento das imagens é feita na linguagem C/C++ e apenas parâmetros simples são transferidos entre as funções em C/C++ e os métodos em C#.

A Figura 9 exibe a declaração de uma função implementada em C/C++ para que essa função seja executada a partir do código C#.

Figura 9 - Declaração de função C/C++ no C#



```

#region Métodos cvTools
[System.Runtime.InteropServices.DllImport
    ("cvTools.dll", EntryPoint = "?TrainDataBase@@YAHPAD@Z")]
public static extern int TrainDataBase(string dbName);

[System.Runtime.InteropServices.DllImport
    ("cvTools.dll", EntryPoint = "?Recognize@@YAHPAD@Z")]
public static extern int Recognize(string fileName, string dbName);
#endregion
  
```

Após declarar as funções em C/C++ de forma como está na Figura 9, essas funções podem ser executadas no código C# como se fossem nativas do C#.

Por fim, apenas alguns recursos do Emgu CV estão presentes no WhoRU. Podem ser citados os componentes de exibição das imagens e o controle do acesso à webcam.

4.4 Desenvolvimento do WhoRU

O WhoRU é uma aplicação de reconhecimento facial onde uma imagem de entrada é selecionada e comparada com as imagens armazenadas na base de dados, em seguida é apresentada a imagem mais próxima à imagem de entrada.

A plataforma de desenvolvimento utilizada foi o Microsoft Visual Studio 2008 que possui ferramentas para desenvolvimento de aplicações desktop e web. Dentre as linguagens de programação suportadas pelo Microsoft Visual Studio 2008 foram utilizadas C, C++ e C#. As linguagens C e C++ são utilizadas nas funções da biblioteca OpenCV e C# foi utilizada no desenvolvimento da interface gráfica.

Figura 10 - Janela principal do WhoRU



O WhoRU possui duas fases de funcionamento, a fase de treinamento e a fase de reconhecimento, as quais são descritas nos capítulos 4.4.1 e 4.4.2 respectivamente.

4.4.1 Treinamento

A fase de treinamento inicia com a leitura das imagens de treinamento. As imagens ficam armazenadas em uma pasta do programa.

Figura 11 - Leitura das imagens

```

for(iFace = 0; iFace < nFaces; iFace++)
{
    strcpy(imageFileName, GetImageFileName(dbName, iFace));
    *(personNumTruthMat->data.i+iFace) = iFace+1;
    faceImgArray[iFace] = cvLoadImage(imageFileName, CV_LOAD_IMAGE_GRAYSCALE);

    if(!faceImgArray[iFace])
    {
        fprintf(stderr, "Nao foi possivel carregar o arquivo %s\n", imageFileNa
        return 0;
    }
}

```

Após a leitura das imagens é definido o subespaço da face. Esse subespaço é criado por uma função da biblioteca OpenCV chamada `cvCalcEigenObjects()` como demonstrado na Figura 12

Figura 12 - Definição do subespaço da face

```

eigenValMat = cvCreateMat(1, nEigens, CV_32FC1);
pAvgTrainImg = cvCreateImage(faceImageSize, IPL_DEPTH_32F, 1);
calcLimit = cvTermCriteria(CV_TERMCRIT_ITER, nEigens, 1);

cvCalcEigenObjects(nTrainFaces, (void*)faceImgArray, (void*)eigenVectArr,
cvNormalize(eigenValMat, eigenValMat, 1, 0, CV_L1, 0);

```

De acordo com o funcionamento da PCA precisamos extrair a face média de todas as faces, calcular a matriz de covariância etc, mas todo esse processo é executado pela função `cvCalcEigenObjects()`.

Agora que temos o subespaço, projeta-se as faces nesse subespaço. Isso é feito com a função `cvEigenDecomposite()` como demonstrado na Figura 13.

Figura 13 - Projeção das imagens no subespaço

```

for(i = 0; i < nTrainFaces; i++)
{
    cvEigenDecomposite(faceImgArray[i], nEigens, eigenVectArr, 0, 0,
        pAvgTrainImg, projectedTrainFaceMat->data.fl + i * offset);
}

StoreTrainingData(dbFolderName);

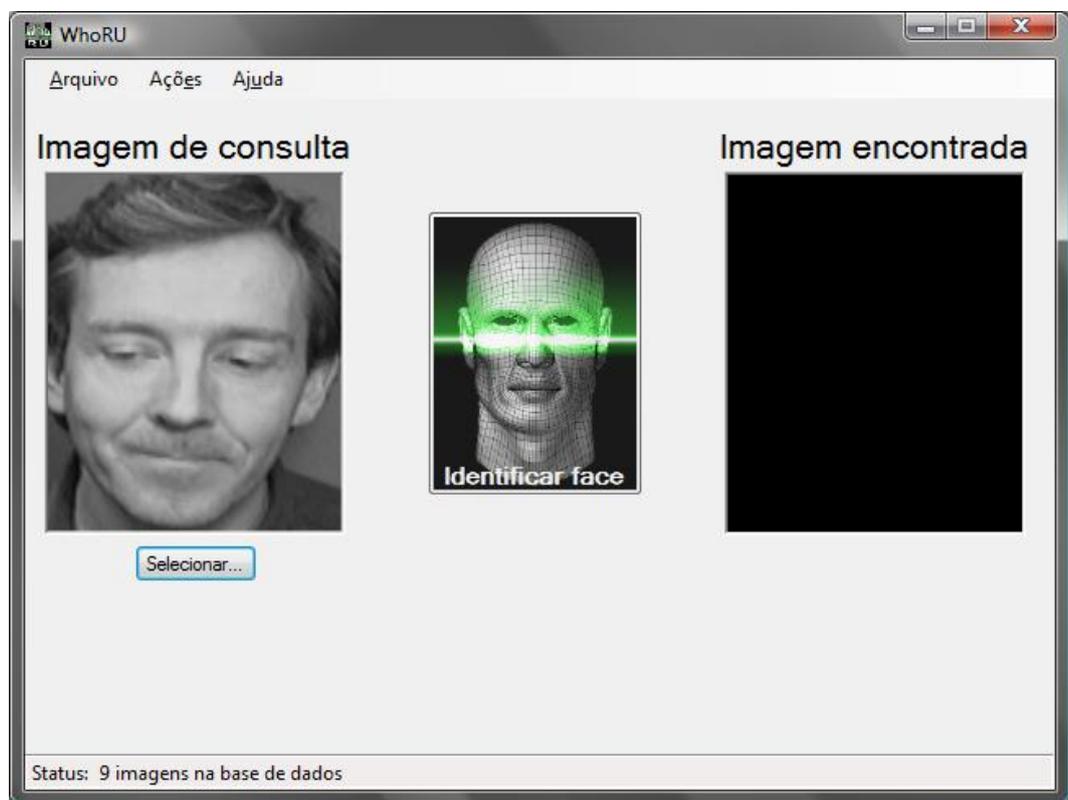
```

Após a projeção das faces as informações são armazenadas em um arquivo XML pela função StoreTrainingData().

4.4.2 Reconhecimento

A fase de reconhecimento inicia-se com a seleção da imagem de entrada conforme exibido na Figura 14.

Figura 14 - Seleção da imagem de entrada



Após selecionar a imagem de entrada clica-se no botão “Identificar face” e em seguida a imagem de entrada passa pelo mesmo processo aplicado às imagens de treinamento resultando no mesmo tipo de informação das imagens de treinamento. Após extrair as informações necessárias da imagem de entrada esse resultado é comparado com as informações de cada imagem de treinamento aplicando-se alguma técnica de cálculo de distância como, por exemplo, a Distância Euclidiana. A função que executa o processo de reconhecimento é demonstrada na Figura 15.

Figura 15 - Processo de reconhecimento da face

```

CVTOOLS_API int Recognize(char* filePath, char* dbName)
{
    int i, iNearest, nearest, truth;;
    CvMat* trainPersonNumMat = 0;
    float* projectedTestFace;
    dbFolderName = dbName;
    IplImage* testImage;

    if(!loadTrainingData(&trainPersonNumMat))
        return -1;
    projectedTestFace = (float*)cvAlloc(nEigens * sizeof(float));
    testImage = cvLoadImage(filePath, CV_LOAD_IMAGE_GRAYSCALE);

    cvEigenDecomposite(testImage, nEigens, eigenVectArr,
        0, 0, pAvgTrainImg, projectedTestFace);

    iNearest = findNearest(projectedTestFace);
    nearest = trainPersonNumMat->data.i[iNearest];

    return nearest;
}

```

Após a execução da função de reconhecimento é retornado um valor que identifica a face encontrada e a mesma é exibida para o usuário.

Figura 16 - Exibição da imagem encontrada



Se for desejo do usuário é possível acrescentar imagens de treinamento. Essa funcionalidade é demonstrada na Figura 17.

Figura 17 - Captura de imagem



O WhoRU ainda não possui uma implementação de localização de face em uma imagem, por conta disso, é necessário que o usuário ajuste a posição do rosto dentro da marca azul, pois, essa é a área capturada. A área capturada possui as mesmas dimensões das imagens de treinamento. Após armazenar a nova imagem na base de dados é necessário executar o treinamento das imagens novamente para que as informações da nova imagem sejam extraídas e armazenadas com as informações existentes.

Conclusão

Nesse trabalho foi descrito o desenvolvimento matemático das técnicas PCA e Eigenface aplicadas no reconhecimento de faces humanas. A PCA é uma ótima abordagem para efetuar a redução da dimensionalidade dos dados, viabilizando a implementação de um sistema de reconhecimento facial com baixo custo computacional.

Foi desenvolvida uma aplicação teste utilizando a ferramenta Matlab 7, a qual possui recursos que facilitam o trabalho com cálculos de matrizes.

Em seguida foi desenvolvido um sistema de reconhecimento facial com características de um sistema que possa ser utilizado em um ambiente real chamado WhoRU.

Durante o desenvolvimento do WhoRU foram encontradas várias dificuldades e uma das principais foi a tentativa de utilização de um wrapper do OpenCV para C#, mas apenas alguns recursos relacionados à interface gráfica foram utilizados. Algumas funções essenciais para o processo de cálculo da PCA possuem parâmetros incompatíveis com as funções nativas do OpenCV. Diante dessa dificuldade todo o processo de cálculo da PCA foi desenvolvido em sua linguagem nativa e o C# foi utilizado apenas para o desenvolvimento da interface gráfica.

Alguns trabalhos futuros podem ser desenvolvidos para a melhoria do WhoRU. Como exemplo podem ser citados a implementação da localização da face em uma imagem na captura de uma nova face através da câmera pra que não haja a necessidade do usuário ajustar a posição do rosto na marca azul. Outra opção de melhoria é a utilização da PCA Incremental que possui o mesmo funcionamento da PCA apresentado nesse trabalho, mas ao inserir uma nova face na base de dados, não há a necessidade de efetuar o treinamento de todas as faces novamente, é necessário apenas extrair as informações da nova face e acrescentá-las às informações de treinamento já existentes.

REFERÊNCIAS

BASTOS, Prefeitura. Realizada oficina de trabalho que consolida a implantação do serviço eletrônico InluaSaúde. Janeiro 2009. Disponível em <http://www.bastos.sp.gov.br/noticia.php?id=33>. Acesso em 10 Nov. 2009.

COGNITEC. Cognitec Brasil, Novembro 2009. Disponível em <http://www.cognitec.com.br>. Acesso em 10 Nov. 2009.

DATTORRO, J. Convex optimization and euclidean distance geometry, v. 2009.12.02. Meboo, 2005.

BRADSKI, Gary; KAEHLER, Adrian. Learning OpenCV, USA. O'Reilly Media, 2008.

CHELLAPPA, R.; SOROHEY, S.; WILSON, C. L.; BARNES, C. S., Human and machine recognition of faces: a survey. Agosto, 1994.

GONZALEZ, R. C.; WOODS, R. E., Digital image processing, 2ª Ed. Prentice Hall, 2002.

JAIN, Anil K.; DUIN, Robert P. W.; MAO, Jianchang. Statistical pattern recognition: a review. November, 2000.

JAIN, Anil K.; FLYNN, Patrick; ROSS, Arun A..Handbook of biometrics. USA: Springer, 2008.

LIN, Shang-Hung. An introduction to face recognition technology. Informing Science Special Issue on Multimedia Informing Technologies, Vol. 3, Nº 1, 2000.

NETITECH, NETi Tecnologia, Novembro 2009. Disponível em http://www.netitec.com.br/index.php?option=com_content&task=view&id=18&Itemid=35. Acesso em 10 Nov. 2009.

POGGUETTO, Priscila D., Denatran quer transformar exame de motorista em vestibular, Agosto 2008. Disponível em <http://g1.globo.com/Noticias/Carros/0,,MUL737666-9658,00.html>. Acesso em 20 Nov. 2009.

THEODORIDIS, S.; KOUTROUMBAS, K..Pattern Recognition, 3^a ed. USA: Academic Press, 2006.

TURK, M.; PENTLAND, A..Eigenfaces for recognition, Journal of Cognitive Neuroscience, v. 3, n. 1, Março 1991.