



Universidade Federal do Rio de Janeiro

Instituto Alberto Luiz Coimbra de Pós-Graduação e Pesquisa de Engenharia
Programa de Engenharia de Sistemas e Computação

Rio de Janeiro, RJ – Brasil

RECONHECIMENTO FACIAL UTILIZANDO EIGENFACES

Abel Bruno Nascimento Silva

16 de junho de 2013

Disciplina: Introdução ao Processamento de Imagens
Professores: Antônio Oliveira
Ricardo Marroquim

Sumário

1. Introdução	3
2. Detecção Facial	4
3. Banco de Faces.....	5
4. Cálculo das Eigenfaces	6
4.1. Passo 1:	6
4.2. Passo 2:	6
4.3. Passo 3:	7
4.4. Passo 4:	7
4.5. Passo 5:	7
4.6. Passo 6:	8
5. Projeção no Espaço de Faces	9
6. Reconhecimento Facial.....	9
6.1. Passo 1:	9
6.2. Passo 2:	9
6.3. Passo 3:	10
6.4. Passo 4:	10
7. Resultados	10
8. Conclusão.....	12

1. Introdução

A capacidade do ser humano de reconhecer faces é notável. Uma pessoa pode reconhecer milhares de faces ao longo de sua vida e identificar rostos familiares de forma rápida mesmo após anos de separação. Embora haja diversas mudanças faciais ocorridas devido a diferentes expressões faciais, distrações como acessórios (óculos, tapa-olho, etc.) ou mudanças de corte de cabelo e barba, o ser humano consegue realizar esta tarefa de forma robusta [1].

Computadores que reconhecem faces podem ser utilizados em uma grande variedade de problemas: identificação criminal, sistemas de segurança, processamento de imagens e vídeos e interação homem-máquina. Entretanto, desenvolver sistemas de reconhecimento facial não é uma tarefa trivial devido à complexidade das faces e suas grandes variações. A fim de desenvolver nos computadores a habilidade de reconhecer rostos, foram desenvolvidos modelos de reconhecimento facial tais como Linear Discriminant Analysis, Elastic Bunch Graph Matching e Reconhecimento Facial utilizando Eigenfaces.

Eigenfaces são um conjunto de autovetores de uma matriz de covariância formada por imagens de faces. A ideia de utilizar eigenfaces no reconhecimento de faces surgiu com a técnica desenvolvida por Sirovich e Kirby [2] para representar imagens de rostos de forma eficiente utilizando Análise de Componentes Principais. Turk e Pentland [1] utilizam esta representação para avaliar a distância entre rostos em um “espaço de faces” e assim determinar rostos similares.

Este trabalho teve como objetivo implementar o algoritmo de reconhecimento facial proposto por Turk e Pentland. O sistema desenvolvido recebe como entrada imagens capturadas através de uma webcam e utiliza o algoritmo de Turk e Pentland para determinar em tempo-real se o rosto obtido pertence a alguma pessoa armazenada previamente em um banco de faces, caso pertença, o programa indica qual pessoa foi reconhecida.

Nesta seção de introdução foram apresentados o contexto e a motivação para a realização do trabalho. As demais seções estão organizadas da seguinte maneira: as seções 2 a 6 detalham cada etapa do algoritmo. A seção 7 apresenta os resultados obtidos e a seção 8 apresenta as conclusões deste trabalho.

2. Detecção Facial

A etapa de detecção facial é o primeiro passo para a implementação de sistemas de reconhecimento facial. Assim como o reconhecimento, apesar de ser uma tarefa facilmente executada pelo ser humano, a detecção de rostos pelo computador necessita da utilização de diversas técnicas de processamento de imagens. Alguns desafios encontrados na detecção facial incluem: faces em diferentes tamanhos na imagem, oclusão de faces, faces não frontais, variação de luminosidade, etc. Na Figura 1 observa-se a detecção de rostos aplicada em uma fotografia.

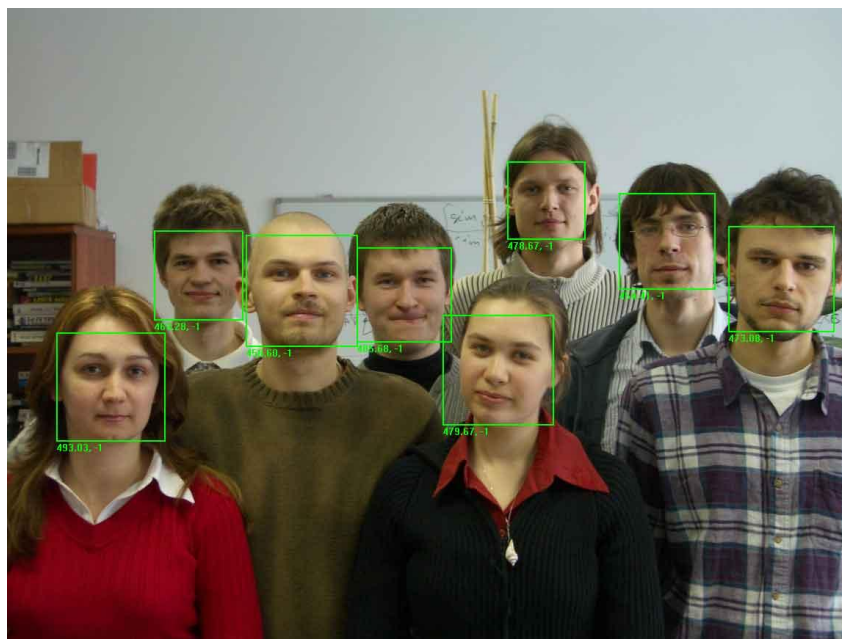


Figura 1. Detecção de rostos em uma imagem.

Por não fazer parte do escopo deste trabalho, a detecção da face do usuário não é realizada de forma automática. O usuário deve posicionar manualmente o seu rosto em um retângulo centralizado na imagem, sendo esta a área em que será processado o reconhecimento facial.

3. Banco de Faces

Para realizar o reconhecimento deve-se estabelecer um conjunto de imagens de pessoas que serão reconhecidas – no caso deste trabalho, foram reconhecidas 10 pessoas. O algoritmo desenvolvido por Turk e Pentland necessita que todas as imagens contenham os rostos centralizados, alinhados e com o mesmo tamanho. O banco de faces utilizado neste trabalho contém, para cada pessoa, 4 imagens com variações de expressões, formando um total de 40 imagens presentes no banco. A Figura 2 apresenta algumas imagens presentes no banco utilizado neste trabalho.



Figura 2. Banco de faces.

As imagens armazenadas possuem dimensão de 92×112 pixels, e para simplificar o cálculo das eigenfaces trabalhou-se apenas com imagens em nível de cinza.

4. Cálculo das Eigenfaces

4.1. Passo 1:

Uma imagem pode ser vista como uma matriz A de $n \times m$ pixels, entretanto, para estabelecer o relacionamento entre um conjunto de imagens necessita-se de um “espaço de faces”, no qual se pode computar o quão similar duas faces são. Assim, ao invés de representar as imagens como matrizes, estas são representadas como pontos. Para isto, todas as linhas da matriz A são unidas em um único vetor linha Γ com $n \times m$ coordenadas, desta forma obtém-se um “espaço de faces” \mathbb{R}^{nm} . Nota-se que este espaço é grande mesmo para imagens pequenas, para as imagens utilizadas neste trabalho, por exemplo, o espaço possui 10304 dimensões.

4.2. Passo 2:

Após obter os vetores Γ de todas as M imagens do banco, computa-se a face média Ψ , que representa o centro de gravidade do conjunto de faces:

$$\Psi = \frac{1}{M} \sum_{i=1}^M \Gamma_i$$

A Figura 3 exhibe o resultado obtido.



Figura 3. Face média.

4.3. Passo 3:

Para todo vetor Γ calcula-se o vetor Φ :

$$\Phi_i = \Gamma_i - \Psi$$

A Figura 4 apresenta o resultado deste passo.



Figura 4. Faces subtraídas da média.

4.4. Passo 4:

Constrói-se a matriz A tal que cada i -ésima coluna da matriz é a transposta do vetor linha Φ_i . Portanto, neste trabalho, a matriz A possui dimensão 10304×40 .

4.5. Passo 5:

Calcula-se a matriz de covariância C :

$$C = AA^T$$

4.6. Passo 6:

Para se obter as eigenfaces é preciso apenas computar os autovetores da matriz AA^T . Entretanto, a matriz AA^T possui dimensão 10304×10304 , o que faz com que o cálculo de seus autovetores seja computacionalmente inviável. Para contornar este problema considera-se a matriz $A^T A$ de dimensão 40×40 e seus autovetores v tais que:

$$A^T A v_i = \lambda v_i$$

Multiplicando ambos os lados por A , tem-se:

$$AA^T A v_i = \lambda A v_i$$

Desta forma pode-se observar que Av_i são os autovetores de $C = AA^T$ associados aos 40 maiores autovalores da matriz.

O cálculo das eigenfaces (autovetores) neste trabalho, assim como toda a manipulação das matrizes, foi realizado através da biblioteca Eigen [3] de álgebra linear.

A Figura 5 apresenta algumas eigenfaces obtidas neste trabalho.

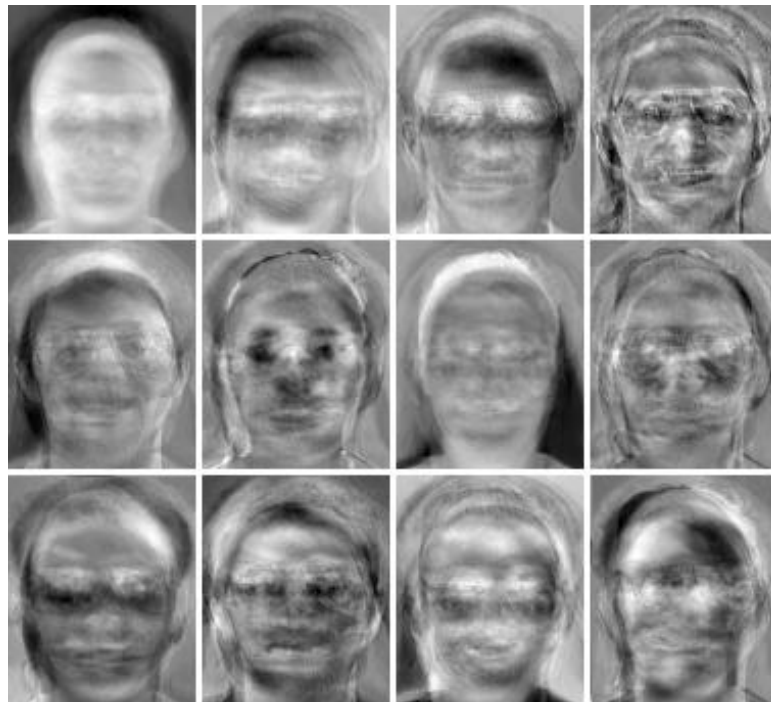


Figura 5. Eigenfaces.

5. Projeção no Espaço de Faces

Para se calcular a proximidade entre duas faces, faz-se necessário projetar a imagem do rosto no “espaço de faces”, o que pode ser facilmente computado considerando as M eigenfaces u :

$$\omega_k = u_k^T \Phi$$

Para $k = 1, \dots, M$. Desta forma, os pesos ω formam o vetor:

$$\Omega^T = [\omega_1, \omega_2, \dots, \omega_M]$$

Que representa a imagem projetada no “espaço de faces”. Após o cálculo do vetor Ω para todos os rostos armazenados no banco, inicia-se a etapa de reconhecimento facial.

6. Reconhecimento Facial

A etapa de reconhecimento facial recebe como entrada uma imagem capturada através da webcam. Para realizar esta captura utilizou-se a biblioteca de visão computacional OpenCV [4].

6.1. Passo 1:

Para a nova face calcula-se o vetor Φ .

6.2. Passo 2:

Projeta-se no “espaço de faces” o vetor obtido no passo anterior.

6.3. Passo 3:

Utilizando a projeção Ω obtida no passo 6.2 calcula-se a distância entre Ω e todas as projeções obtidas na etapa 5, selecionando a menor distância d entre as projeções:

$$d = \min_k \|\Omega - \Omega_k\|$$

6.4. Passo 4:

Se d for menor que um threshold pré-estabelecido, então a nova face pertence ao indivíduo cujo rosto armazenado no banco apresenta distância d da nova face. Caso contrário, trata-se de uma face desconhecida.

7. Resultados

As Figuras 6 e 7 exibem alguns resultados do projeto desenvolvido.

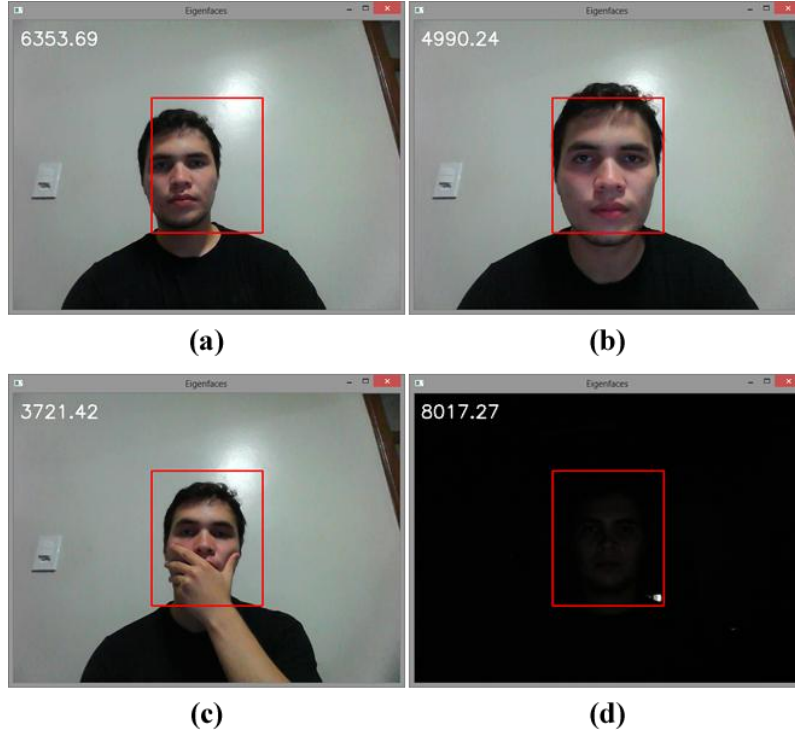


Figura 6. Resultados negativos obtidos na etapa de reconhecimento facial. (a) Face descentralizada, (b) face em escala diferente, (c) oclusão e (d) diferença de iluminação e background.

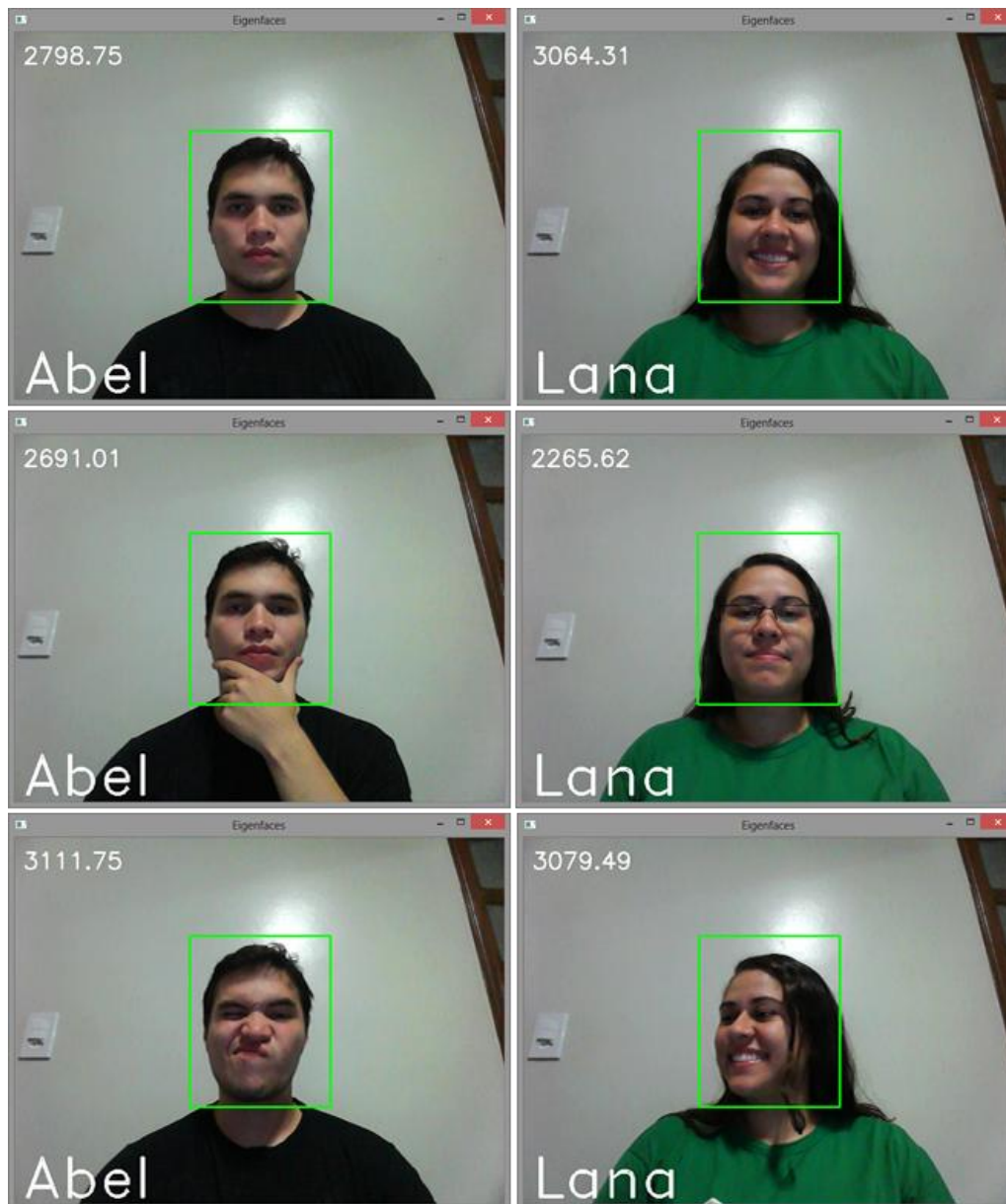


Figura 7. Resultados positivos obtidos na etapa de reconhecimento facial.

Como ilustrado na Figura 6, o algoritmo apresenta grande dependência da iluminação e da maneira como a face está posicionada na imagem, o que justifica a necessidade de que as imagens armazenadas no banco de faces estejam centralizadas e em uma mesma escala.

Entretanto, a partir dos resultados obtidos, pode-se observar que o algoritmo desenvolvido por Turk e Pentland, apesar de apresentar uma implementação relativamente simples, alcança um bom resultado no reconhecimento de faces mesmo em rostos com variação de expressão facial e com pequenas rotações e oclusões (Figuras 7).

8. Conclusão

Este relatório apresentou as etapas do sistema de reconhecimento facial desenvolvido na disciplina Introdução ao Processamento de Imagens. O algoritmo implementado apresentou bons resultados no reconhecimento de faces, mesmo estas apresentando pequenas variações.

Possíveis melhorias no trabalho apresentado incluem: detecção automática das faces, não sendo mais necessário o posicionamento manual do rosto por parte do usuário; processamento das imagens das faces visando atenuar problemas de iluminação; e adição automática de novos rostos no banco de faces.

Referências

- [1] Turk, M., Pentland, A. *Eigenfaces for Recognition*. Journal of Cognitive Neuroscience. vol 3, no. 1. pp. 71-86, 1991.
- [2] Sirovich, L., Kirby, M. *Low-dimensional Procedure for the Characterization of Human Faces*. Journal of the Optical Society of America A. 4(3), 519-524, 1987.
- [3] Eigen. *C++ Template Library for Linear Algebra*. Último acesso em: 15 de junho de 2013. Disponível em: < <http://eigen.tuxfamily.org> >.
- [4] OpenCV. *Open Source Computer Vision*. Último acesso em: 15 de junho de 2013. Disponível em: < <http://opencv.org> >.