

Introdução ao Processamento de Imagens

Projeto Final

Detecção de Faces em Imagens

Gabriel Martins de Miranda 130111350
Jeoman Mariano Goes 130116114
Marina Martins de Miranda 110132351
Departamento de Ciência da Computação
Universidade de Brasília
Brasília - DF

Professor: Alexandre Zaghetto
Introdução ao Processamento de Imagens
Turma: A
2º semestre de 2014

Resumo – Este projeto baseou-se no artigo *Simple Face-detection Algorithm Based on Minimum Facial Features* (de Yao-Jiunn Chen e Yen-Chun Lin) disponibilizado pelo professor. Nele, é apresentado um algoritmo rápido e preciso para detectar rostos humanos pelas correlações geométricas entre a localização de rosto e cabelo (intersecção desses componentes). Primeiro se adquire a imagem e há uma segmentação do que pode ser pele, por uma gama de cores de pele, localizando inicialmente as faces. Isso também é feito para o que pode ser cabelo, pelo espectro de cores de cabelo. Após tal procedimento, faz-se uma quantização por limiarização para agrupar blocos de pele e blocos de cabelo para decidir possíveis áreas candidatas a ser face. Por fim, é testado se há intersecção entre esses blocos e se determinam as reais localizações de face. A precisão é perto de 90% em um ambiente com iluminação e fundo adequados. Tal algoritmo pode ser transferido para sistemas embarcados, atingindo novas áreas de interesse, como até sistemas em miniatura.

Palavras-chave: detecção de faces; correlação geométrica; segmentação; gama de cor; espectro de cor; quantização; intersecção de blocos.

I. INTRODUÇÃO

Nos últimos anos, as aplicações baseadas em detecção de de face foram bastante desenvolvidas e têm um importante papel no nosso cotidiano, desde smartphones a câmeras digitais, sistemas de segurança e robôs com inteligência artificial. Porém, muitas vezes esses algoritmos são deveras complexos e não factíveis em tempo real.

Por isso, várias abordagens têm sido propostas para melhorar a performance da detecção de faces e são categorizadas em algumas classes, como o Método Baseado no Conhecimento, que objetiva encontrar características invariantes de uma face dentro de um ambiente complexo, localizando a sua posição, com a ajuda das correlações geométricas entre essas características [1]. Há várias outras que utilizam abordagens baseadas em padrões de características e matching com estes ou modelos de forma ou ainda baseados em redes neurais de treinamento de reconhecimento. Porém o Método Baseado no Conhecimento é um dos menos complexos e foi o que utilizamos, propondo um algoritmo de complexidade reduzida para detecção de face em tempo real, adotando as as características geométricas de pele e cor de cabelo.

Além disso, pode ser usado em sistemas embarcados, como até em robôs de estimação.

II. METODOLOGIA

Para atender aos objetivos do programa, foram criadas uma rotina principal `trabalho_final_ipi.m` e 5 funções: `hsi_pele.m`, `hsi_cabelo.m`, `qtz_pele_cabelo.m`, `F1.m` e `F2.m`.

No nosso trabalho, optamos por implementar a detecção em vídeo, a fim de mostrá-la em tempo real. Para tanto, utilizamos algumas funções prontas do Matlab, como `webcam()`, que cria um objeto para a webcam e a conecta a uma câmera do sistema, e outras funções suporte. É importante destacar que tais funções só estão disponíveis em pacotes a partir da versão R2014a do Matlab.

A fim de tornar o algoritmo mais rápido, a imagens obtidas foram multiplicadas por um fator de redimensionamento igual a 0,2 para facilitar o seu processamento e ao final são convertidas para o tamanho real para serem mostradas na tela.

As informações a respeito da cor foram as principais ferramentas para extrair características de pele e de cabelo das imagens. Abaixo, um diagrama retirado do artigo principal representa os passos do algoritmo, a saber:

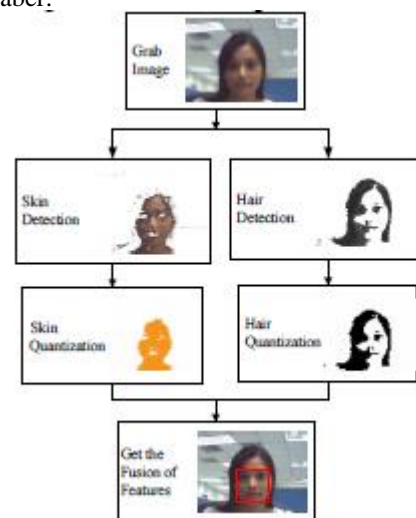


Fig 1 : Diagrama geral

- A) Detecção de Pele, por meio da informação de cor a fim de segmentar possíveis áreas de cor de pele. B) Detecção de cabelo, por meio de informação de brilho, a fim de encontrar onde o cabelo está. C) Quantização de pele, quantizando os pixels de cor de pele e identificando blocos de pele. D) Quantização de cabelo, quantizando os pixels de cor de cabelo e identificando blocos de cabelo. E) Fusão das características: determina se as áreas detectadas como pele fazem parte de um rosto, verificando a sua correlação geométrica com áreas de cabelo.

Cada um desses módulos será explicado a seguir:

A) Detecção de pele

Após muitos estudos, foi possível obter uma gama de cor de pele em uma imagem. A extração de cor do modelo RGB normalizado foi mais eficiente do que a de cor do RGB padrão, pois este se mostrou muito sensível a variações de cor. Então, podemos ter os valores do RGB normalizado por meio do RGB padrão pelas seguintes fórmulas:

$$r = \frac{R}{R+G+B} \quad (1)$$

$$g = \frac{G}{R+G+B} \quad (2)$$

As equações 1 e 2 representam as normalizações para um pixel vermelho e verde, respectivamente. A principal distribuição de cor de pele está concentrada no diagrama abaixo, com g em função de r, na Figura 2 também retirada do artigo principal.

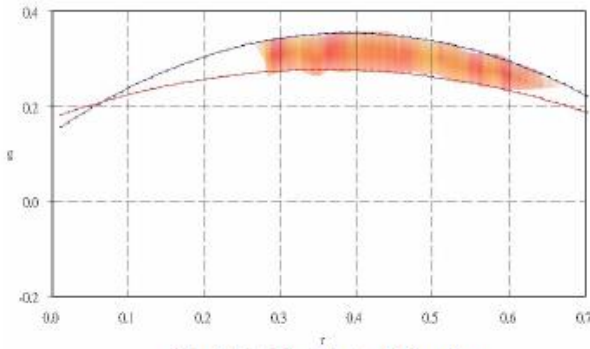


Fig. 2 Statistical chart of skin color

As duas curvas acima que determinam o limite superior $F_1(r)$ e o inferior $F_2(r)$ da cor de pele foram obtidas experimentalmente [11] e podem ser representadas pelas equações:

$$F_1(r) = -1.376r^2 + 1.0743r + 0.2 \quad (3)$$

$$F_2(r) = -0.776r^2 + 0.5601r + 0.18 \quad (4)$$

A cor branca, em que $g = 0,33$ e $r = 0,33$ também é incluído nessas equações, então é necessário retirá-los, por meio de:

$$w = (r - 0.33)^2 + (g - 0.33)^2 > 0.001 \quad (5)$$

Assim, unindo as equações, temos:

$$Skin = \begin{cases} 1 & \text{if } (g < F_1(r) \cap g > F_2(r) \cap w > 0.001) \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Porém, ao realizar os experimentos, percebeu-se que este modelo de cor apresentava algumas falhas, como segmentar pixels azuis, amarelos e cinzas junto dos pixels da pele. Assim, um novo modelo de cor foi escolhido para estimar cor de pele: o HSI, que é mais robusto e se baseia em hue(matiz), saturation(saturação) e intensity(intensidade). Abaixo, uma figura mostra a relação entre os vários espaços de cor:

Aprimoramento de cor com modelo HSI

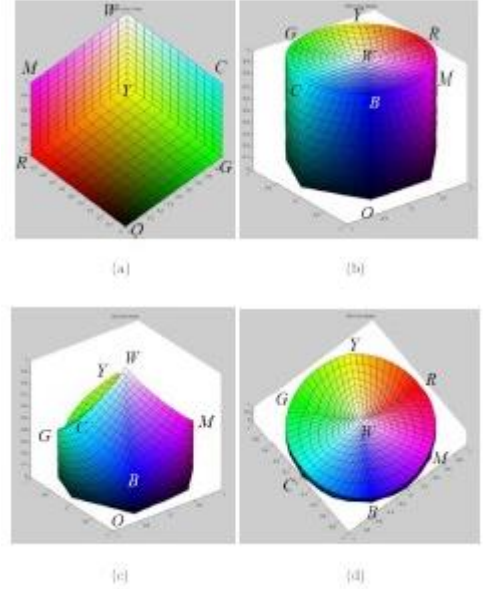


Fig. 3 Análise de cor: a) Análise no Cubo RGB b) Análise no espaço HSV c) Análise no espaço HSI d) Vista superior do HSI [15]

A seguintes fórmulas mostram a relação entre o espaço de cores HSI e o RGB:

$$\theta = \cos^{-1} \left\{ \frac{0.5[(R-G)+(R-B)]}{\sqrt{(R-G)^2 + (R-B)(G-B)}} \right\}$$

$$\begin{cases} H = \theta & \text{if } B \leq G \\ H = 360^\circ - \theta & \text{if } B > G \end{cases} \quad (7)$$

Após os experimentos, algumas modificações foram feitas até se chegar nas equações:

$$Skin = \begin{cases} 1 & \text{if } (g < F_1(r) \cap g > F_2(r) \cap w > 0.001 \cap (H > 240 \cup H \leq 20)) \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

A seguir, mostramos um frame de um vídeo que fizemos com os três integrantes deste projeto e a segmentação por cor de pele.



Fig. 4 Imagem original



Fig. 5 Detecção de Pele

No algoritmo, e na Fig. 5, pixels reconhecidos como não pele foram substituídos por pixels pretos.

A Figura 5 mostra que muitos pixels de não- pele são incluídos como se fossem pele, como algumas partes da roupa, por exemplo. Porém, o Módulo de quantização de pele realiza as operações de acompanhamento do algoritmo como um todo , de modo a minimizar o impacto desse “mal-julgamento” .

B) Detecção de Cabelo

O elemento intensidade do modelo HSI foi utilizado para a detecção de cor de cabelo. A relação dele com o espaço RGB é descrito na seguinte fórmula:

$$I = \frac{1}{3}(R + G + B) \quad (9)$$

Este elemento permite avaliar a luminosidade ou brilho da imagem; além disso, a cor do cabelo é definida para estar na gama de cor mais escura . A equação 10 define a faixa de intensidade da cor do cabelo com base em experiências .

$$Hair = \begin{cases} 1 & \text{if } (I < 30 \cap (B - G < 15 \cup B - R < 15)) \\ & \cup \\ & (20 < H \leq 40) \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

Na equação do artigo principal escolheram $I < 80$, mas no nosso experimento optamos por escolher $I < 30$ na equação acima, por captar melhor as cores de cabelo na maior parte das vezes em que foi testado com as condições de iluminação disponíveis .

A figura a seguir mostra a Fig 4 após ser processada pela equação 10.



Fig. 6 Detecção de Cabelo

Os pixels reconhecidos como não cabelo foram substituídos por pixels brancos.

Os pixels de cor de cabelo comumente e estatisticamente são incluídos na condição ($B - G < 15 \cup B - R < 15$), que por sua vez, exclui os pixels de um azul profundo. A condição ($I < 30$) inclui os pixels que são mais escuros, e a condição ($20 \leq 40 < H$) inclui os pixels que são castanhos.

A Figura 6 mostra que muitos pixels de não- cabelo são incluídos como se fossem cabelo, como a roupa, por exemplo. Porém, o Módulo de quantização de cabelo realiza as operações de acompanhamento do algoritmo como um todo, de modo a minimizar o impacto desse “mal-julgamento”.

C) Quantização da pele

Este módulo quantifica os pixels cor de pele por meio de blocos de tamanho 5x5 para decidir se essa região será ou não incluída como de pixels cor de pele. A quantização reduz a resolução da imagem; no entanto, o cálculo da localização geométrica dos pixels é acelerado. O número de pixels pretos em cada bloco de 25 é contado se esse número for maior que um limiar de 12, o bloco inteiro de 25 pixels é considerado como de cor de não-pele. O resto é visto como blocos de cor de pele e é mostrado como branco na Figura 7, que mostra a imagem da Figura 5 depois de passar pela Quantização de Pele.



Fig. 7 Quantização de Pele

D) Quantização de Cabelo

O procedimento de Quantização de Cabelo foi análogo ao anterior. Este módulo quantifica os pixels cor de cabelo por meio de blocos de tamanho 5x5 para decidir se essa região será ou não incluída como de pixels cor de cabelo. A quantização reduz a resolução da imagem; no entanto, o cálculo da localização geométrica dos pixels é acelerado. Aqui é importante notar que os pixels O número de pixels brancos em cada bloco de 25 é contado, e se esse número for maior que um limiar de 12, o bloco inteiro de 25 pixels é considerado como de cor de não-cabelo e é levado para a cor preta. O resto é visto como blocos de cor de cabelo e é mostrado como branco na Figura 8, que mostra a imagem da Figura 7 depois de passar pela Quantização de Cabelo.



Fig. 8 Quantização de Cabelo

E) Fusão das características

Este módulo, referindo-se aos resultados a partir da Quantização de Cabelo, analisa o resultado da Quantização de Pele e analisa se há intersecção entre as duas, determinando a existência de uma face humana.

O nosso algoritmo pode ser utilizado para detectar uma ou mais faces. O diagrama abaixo demonstra o diagrama de multi detecção de faces.

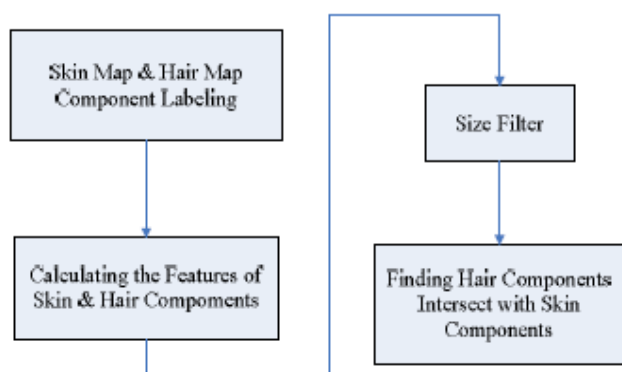


Fig. 9 Diagrama para multi-detecção de faces

Rotulação de componentes (Skin Map & Hair Map Component Labeling) [14]:

Os resultados da Quantização de Pele e da Quantização de Cabelo são novamente calculadas usando “Component Labeling”, ou “Rotulação de Componentes”. Os pixels adjacentes são definidos como um componente; dois componentes não têm pixels adjacentes na fronteira. A Rotulação de Componentes, aplicando o método de 8-adjacência, é capaz de obter os componentes em uma imagem e rotular cada um deles.

Para tanto, no nosso algoritmo utilizamos a função pronta do Matlab, **bwconncomp**, que usa como default conectividade de 8 ou 8-adjacência. Fazendo:

CC = bwconncomp(BW)

temos como retorno os componentes conectados CC encontrados na imagem binária, que pode ser de qualquer tamanho.

CC é uma estrutura com 4 campos:

Connectivity: Conectividade dos components conectados; ImageSize: tamanho de BW; NumObjects: número de componentes conectados(objetos) em BW; PixelIdxList: vetor do tamanho de NumObjects em que cada k-ésimo elemento é um vetor com os índices lineares dos pixels do k-ésimo objeto.

Calculando as Características dos Componentes de Pele e Cabelo (Calculating the Features of Skin & Hair Components):

O algoritmo descrito na figura 9 calcula as características para todos os componentes rotulados. Tais características são a Área, o Centro de Gravidade e as coordenadas dos pixels extremos esquerdo, superior, direito e inferior de cada componente. Tais coordenadas formam um retângulo envolvendo cada objeto, o que é equivalente a uma ‘bounding box’.

Nesta parte, utilizamos a função pronta do Matlab **regionprops**, que mede um conjunto de propriedade de cada componente conectado(objeto) em CC. Fazendo:

S = regionprops(CC, 'basic'); %area, centroid e bounding box

Colocando a string ‘basic’, regionprops computa apenas as medições ‘area’, ‘centroid’ e ‘bounding box’, que é exatamente o que queríamos(!). O tamanho de S é igual ao número de componentes conectados.

Filtro de Tamanho (Size Filter)

O Filtro de Tamanho é necessário para filtrar ruídos e preservar as principais características de pele e de cabelo. As

condições de filtro podem mudar de acordo com os requisitos como a área de cada componente.

No artigo principal, é recomendado manter qualquer componente maior que 10 pixels. Contudo, esse valor não foi muito adequado nos nossos experimentos. Optamos por manter componentes de área maior que 500 para a pele e maiores que 250 para o cabelo.

Encontrando Intersecções de Componentes de Pele e de Cabelo(Finding Hair Components Intersect with Skin Component):

Neste passo, encontramos componentes de cabelo que apresentam intersecção com componentes de pele. A presença de face humana exige que haja tal intersecção. Assim, se não há componente de cabelo com intersecção, não há face humana presente na figura.

A figura 10 mostra uma possível intersecção das bounding boxes.

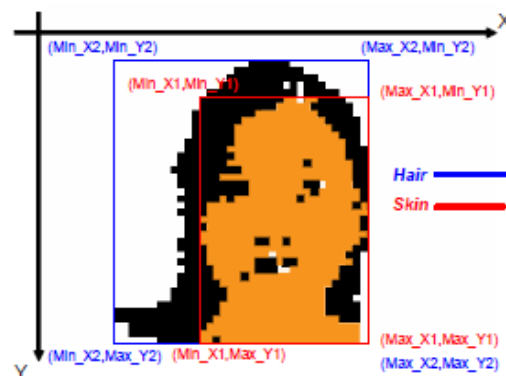


Fig. 10 Intersecção das bounding boxes

Para avaliar se há ou não intersecção, utilizamos a função pronta do Matlab, **rectint**. Fazendo

area = rectint(A,B)

temos como retorno a área de intersecção dos retângulos especificados pelas posições do vetor A e B. Assim, fazendo A e B como as Bounding Boxes de cabelo e pele, podemos facilmente saber se há ou não intersecção entre elas apenas verificando se a área é maior que zero.

Assim, se houver intersecção, a bounding box da pele é marcada como sendo face. A Fig. 11 mostra a detecção de faces da imagem original em Fig. 1.

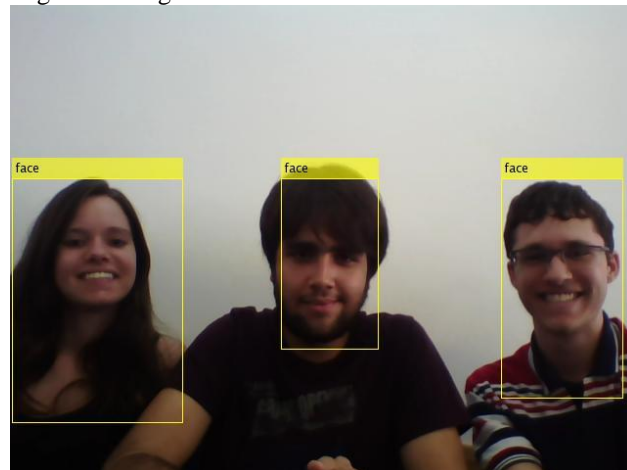


Fig. 11 Resultado da multi Detecção de Faces

O resultado foi excelente, conseguindo capturar as três faces de forma acurada. A Bounding Box na moça é maior porque pega o colo, que é cor de pele, e o cabelo comprido tem intersecção com esta parte do corpo, conectada ao rosto pelo pescoço.

III. RESULTADOS

Foram feitos vários experimentos, adquirindo 2 imagens(frames de vídeos) de um local sem faces, 10 imagens com apenas 1 pessoa e 5 imagens com 2 ou mais pessoas.

Abaixo, são apresentadas algumas dessas imagens. As outras estão em anexo junto do relatório.

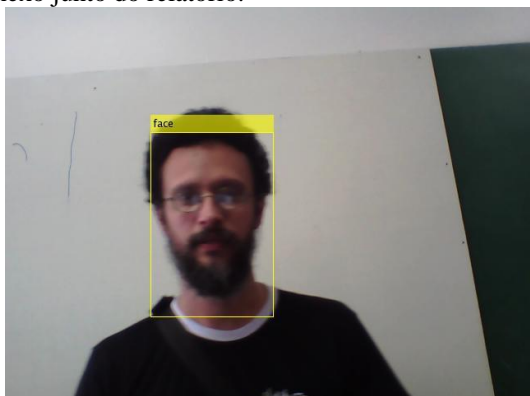


Fig. 12 Exemplo de Detecção com uma face – Prof. Zaghetto

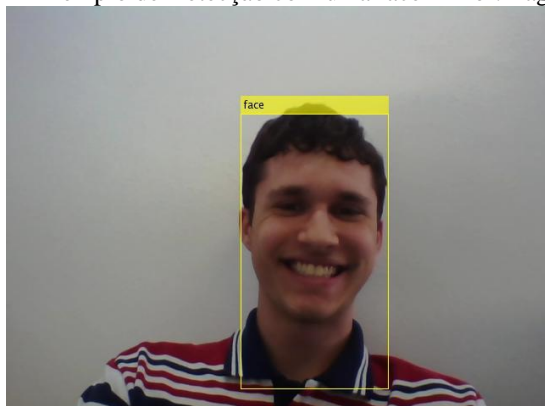


Fig. 13 Exemplo de Detecção com uma face – Gabriel

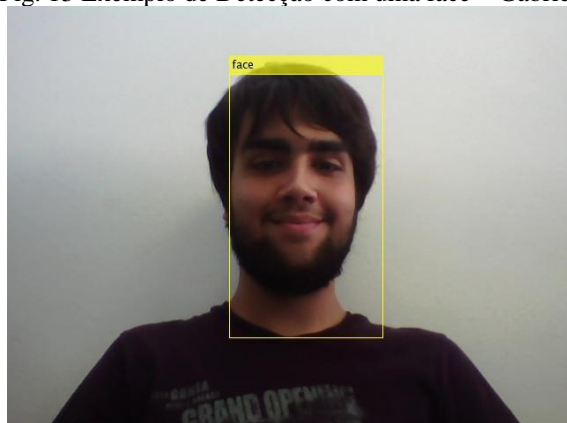


Fig. 14 Exemplo de Detecção com uma face – Jeoman

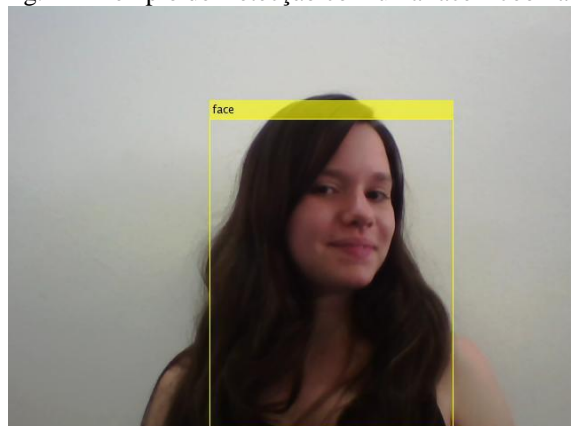


Fig. 15 Exemplo de Detecção com uma face – Marina

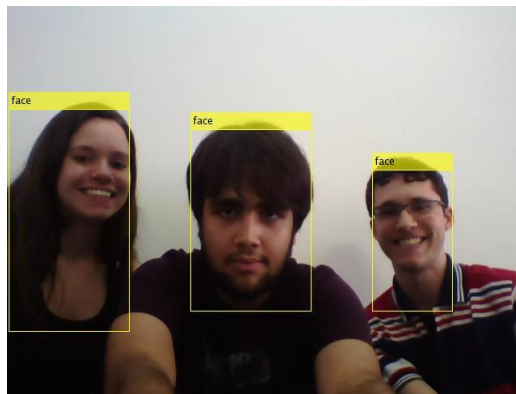


Fig. 16 Exemplo de Detecção com 3 faces

O resultado foi muito bom, com uma acurácia de cerca de 90%. Porém, é muito dependente da iluminação e também da qualidade da câmera, o que é totalmente plausível. Como a segmentação é feita principalmente por cor e cores mais claras são atribuídas a pele e mais escuras são atribuídas a cabelo, a detecção não é muito eficiente com pessoas de cor mais morena.

A quantidade de erros e acertos é descrita na tabela a seguir

Imagens	Amostras	Erros	Acertos
Sem face	2	0	2
Uma pessoa	10	0	10
Duas ou mais pessoas	5	1	4

Tabela 1. Resultados

IV. CONCLUSÕES

Um algoritmo para detecção de face humana em tempo real foi implementado e o resultado foi muito bom, com uma acurácia de cerca de 90%. Ele se utiliza principalmente das gamas de cor de cabelo e pele e da correlação geométrica entre pele do rosto e cabelo e foi eficiente mesmo com pessoas com mesmo cabelo. Porém, é muito dependente da iluminação e também da qualidade da câmera, o que é totalmente plausível. Como a segmentação é feita principalmente por cor e cores mais claras são atribuídas a pele e mais escuras são atribuídas a cabelo, a detecção não é muito eficiente com pessoas com pele mais morena, sendo mais eficaz com pessoas com pele clara e cabelo escuro. Consegue detectar faces mesmo que os rostos estejam inclinados. Uma dificuldade que ocorre é no momento de se detectar a cor de cabelo, porque o algoritmo reconhece várias vestimentas e artefatos do ambiente como tal. Então, se houver algum objeto com cor de pele, pode ser reconhecido como face erroneamente. Uma alternativa seria fazer a detecção encontrando padrões de olhos e boca ao invés de utilizar segmentação por cor. O acompanhamento da implementação do algoritmo foi uma experiência deveras interessante e gratificante a todos os integrantes do grupo.

V. REFERÊNCIAS

* Simple Face-detection Algorithm Based on Minimum Facial Features(de Yao-Jiunn Chen e Yen-Chun Lin)

- [1] Jerome M. Shapiro, "Embedded Image Coding Using Zerotress of Wavelet Coefficients", IEEE Transaction on Signal Processing Vol.41 No.12 Decemrer 1993.
- [2] T.K.Leung, M.C.Burl,and P.Perona, "Finding Face in Cluttered Scenes Using Random Labeled Graph Matching",Proc. Fifth IEEE Int'l Conf. Computer Vision, 1995, pp637-644.
- [3] .Y. Dai and Y.Nakano, "Face-Texture Model Based on SGLD and Its Application in Face Detection in a Color Scence",Pattern Recognition,vol. 29,no. 6, 1996, pp.1007-1017.
- [4] J. Yang and A.Waibel, "A Real-Time Face Tracker", Proc. Third Workshop Applications of Computer Vision, 1996,pp. 142-147.
- [5] I. Craw, D. Tock, and A. Bennett, "Finding Face Features," Proc. Second European Conf. Computer Vision, 1992, pp. 92-96.

- [6] A. Lanitis,C.J. Taylor,and T.F. Cootes, “An Automatic Face Identification System Using Flexible Appearance Models” ,Image and Vision Computing,vol. 13,no. 5, 1995, pp. 393-401.
- [7] M. Turk and A.Pentland,”Eigenface for Recognition, “J.Cognitive Neuroscience,vol.3, 1991, pp.71-86.
- [8] H. Rowley,S. Baluja,and T. Kanade, “Neural Network-Based Face Detection,”IEEE Trans. Pattern Analysis and Machine Intelligence,vol.20,no.1,Jan. 1998.
- [9] A. Rajagopalan,K. Kumar,J.Karlekar,R. Manivasakan,M. Patil,U. Desai, P. Poonacha, and S. Chaudhuri, “Finding Faces in Photographs” ,Proc. Sixth IEEE Int. l Conf. Computer Vision,1998.
- [10] Linda G. Shapiro, George C. Stockman , “Computer Vision” , Prentice Hall, 2001, pp.192-193.
- [11] M. Soriano, S. Huovinen, B. Martinkauppi, and M. Laaksonen, “Using the Skin Locus to Cope with Changing Illumination Conditions in Color-Based Face Tracking,” Proc. of IEEE Nordic Signal Processing Symposium, pp. 383-386, 2000.
- [12] Rafael C. Gonzalez, Richard E. Woods, “Digital Image Processing”, 2nd Edition, Prentice Hall, 2002, pp.299-300.
- [13] William K. Pratt, “Digital Image Processing”, 3rd Edition, John Willey & Sons INC. 2001, pp.63-87.
- [14] Ramesh Jain, Rangachar Kasturi, Brian G. Schunck, “Machine Vision”, McGraw-Hill, 1995 ,pp.44-48.
- [15] Chun-Liang Chien, Din-ChanTseng, Color Image Enhancement with Exact HSI Color Model