

Relatório de projeto de segunda VA

Réplica do trabalho: Simple face-detection algorithm based on minimum facial features

Ismael Cesar da Silva Araujo
Departamento de computação
Ciência da computação
Universidade Federal Rural de Pernambuco (UFRPE)
ismael.cesar@ufrpe.br

I. INTRODUÇÃO

A tarefa de detecção de face pode ser utilizada em diversas aplicações nos dias atuais. Como por exemplo reconhecimento de faces em redes sociais ou até mesmo vigilância e segurança nacional. Porém a tarefa de detectar uma face pode muito custosa, principalmente quando se está utilizando um algoritmo de aprendizagem de máquina. Neste trabalho serão explanadas as etapas de detecção de face baseado no mínimo de características de face detectadas. A organização deste trabalho será da seguinte forma: Na seção II serão introduzidos os conceitos básicos por trás da extração das características mínimas de face. Na seção III é discutida a metodologia utilizada bem como algumas alterações na implementação de modo a se obter melhores detecções. Na seção IV são mostrados resultados e também é discutido como detecções espúrias afetam na precisão do algoritmo. Na seção V são discutidos trabalhos futuros.

II. CONCEITOS BÁSICOS

O algoritmo em [1] pode ser dividido em três passos diferentes. Detecção de Pele e cabelo, onde são computados os valores que representam os intervalos de cores de pele e cabelo para a binarização. Quantização de pele e cabelo, onde uma vez binarizadas as detecções são aplicadas operações morfológicas, são computados os componentes conexos e suas características e aplica-se os filtros de tamanho. Ao final, é feita a união das características, procurando inteseções entre retângulos que contém componentes conexos de cabelo e retângulos que contém componente conexos de pele. Como as etapas de detecção de pele e de cabelo são feitas de modo ligeiramente diferente, a explanação do funcionamento de ambas foi colocada em duas subseções diferentes. Porém as etapas de quantização e união de características ocorre de maneira similar, tanto para detecção de cabelo quanto para detecção de pele.

A. Detecção de pele

O modelo de cor normalizado, trata-se de um tipo de normalização feita por pixel. Considerando uma imagem com canais RGB (Sigla em inglês para Vermelho, Verde e Azul), A normalização da imagem segundo o modelo de cor normalizada calcularia pra cada pixel o valor contido no canal dividido pela soma de todos os valores dos canais no pixel avaliado [1], [3]. Seja ε uma constante da ordem de 10^{-8} somada a o denominador para se evitar divisão por zero(1).

$$\begin{aligned}r &= \frac{R}{R + G + B + \varepsilon} \\g &= \frac{G}{R + G + B + \varepsilon} \\b &= \frac{B}{R + G + B + \varepsilon}\end{aligned}\tag{1}$$

A normalização das cores da imagem possibilitam a diminuição da sensibilidade do algoritmo de detecção em relação as variações de cores e iluminação. Normalizados os intervalos de valores do pixel, é necessário definir funções que avaliam os tons de vermelho que foram normalizados. Tais funções são utilizadas para a definição dos limites superiores e inferiores do intervalo de tons de pele em relação ao canal r [1], [4].

$$\begin{aligned}F_1(r) &= -1.367r^2 + 1.0743r + 0.2 \\F_2(r) &= -0.776r^2 + 0.5601r + 0.18\end{aligned}\tag{2}$$

Para o aprimoramento da detecção de pele necessário definir funções para avaliação de tons e branco, em conjunto com valores de matiz ou *Hue* do píxel. A avaliação dos tons de branco é feita segundo os valores dos canais r e g do píxel. De modo que o píxel é considerado com algum tom de branco quando $r = 0.33$ e $g = 0.33$ [1]. Onde a diferença dos valores dos canais r, g e 0.33 é elevada ao quadrado para para que a mesma só retorne o valor absoluto caso r e g possuam valores menores que 0.33 .

$$White(r, g) = (r - 0.33)^2 + (g - 0.33)^2 \quad (3)$$

Para se constar se o pixel em questão tem algum tom de branco, verifica-se o resultado da comparação entre $White(r, g) > 0.001$. Para melhorar o desempenho da detecção de pele é necessário computar a relação entre o modelo de cor HSI (Hue Saturation and Intensity) com o modelo RGB. Onde *Hue* descreve a cor que está sendo utilizada, o valor está no intervalo em $[0, 360]$ o qual representa o ângulo no círculo unitário. *Saturation* representa o nível de pureza da cor, e *Intensity* trata-se de um valor acromático, que representa a itesidade da cor. Tanto o valor de *Saturation* quanto o de *Intensity* estão no intervalo de $[0, 1]$. A figura a seguir ilustra o espaço de cores do modelo HSI.

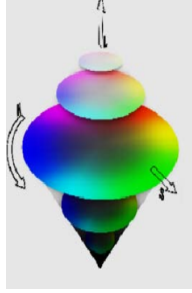


Fig. 1. Espaço de cores do modelo de cores HSI. fonte: [2]

Para o algoritmo de detecção de pele só é necessário computar os valores relativos a *Hue* e *Intensity*. O valor de *Hue* é atribuído segundo os valores B e G do esquema RGB. Porém, antes de se computar o valor de *Hue* é necessário computar o ângulo ao qual o valor de RGB do pixel correspondem no espaço de cores do esquema HSI Fig. 1. As equações para computar os valores de ângulo, *Hue* e *Intensity* respectivamente encontra-se a seguir:

$$\theta(R, G, B) = \cos^{-1} \left(\frac{0.5((R - G) + (R - B))}{\sqrt{(R - G)^2 + (R - B)(G - R)}} \right) \quad (4)$$

$$Hue(B, G, \theta) = \begin{cases} \theta, & \text{if } B \leq G \\ 360^\circ - \theta, & \text{if } B > G \end{cases} \quad (5)$$

$$I(R, G, B) = \frac{1}{3}(R + G + B) \quad (6)$$

Para se efetuar a detecção de pele numa imagem, faz-se uma binarização da imagem segundo os valores de computados segundo as equações mencionadas.

$$SkinDetect = \begin{cases} 1, & \text{if } (g < F_1(r) \cap g > F_2(r) \cap White(r, g) > 0.001 \cap (Hue(B, G, \theta) > 240^\circ \cup Hue(B, G, \theta) \leq 20^\circ)) \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

B. Detecção de Cabelo

Para a detecção de tons de cabelo também é efetuada uma binarização da imagem. Ao se fazer a binarização da imagem utiliza apenas as equações (4)(5)(6) como sub-rotinas. Outros valores de comparação considerados são as diferenças entre o valor B e os valores R e G.

$$HairDetect = \begin{cases} 1, & \text{if } ((I(R, G, B) < 80 \cap ((B - G) < 15 \cup (B - R) < 15)) \cup (20^\circ < Hue(B, G, \theta) \leq 40^\circ)) \\ 0, & \text{otherwise} \end{cases} \quad (8)$$

C. Quantização de pele e cabelo

A etapa de quantização ocorre de forma semelhante tanto para detecção de cabelo quanto para detecção de pele. Uma vez retornada as binarizações das imagens com as detecções de pele e cabelo, são aplicadas operações morfológicas nos retornos *SkinDetect* e *HairDetect*. Os elementos estruturantes tem de tamanho 5×5 . A morfologia é calculada para determinar se um pixel numa região pertence a um componente conexo de cabelo, pele, ou se o pixel varrido é apenas ruído de detecção.

Em seguida, são computados os componentes conexos e as características dos mesmos de *SkinDetect* e *HairDetect*. As características computadas são o centroide, a área do componente conexo e as posições (x, y) dos pixels mais externos, tanto o mais acima e mais a esquerda, quanto o mais abaixo mais a direita.

Após a computação é feita uma filtragem de componentes conexos. Também chamado de filtro de tamanho, ou *size filter*. O filtro de tamanho verifica a área de cada componente conexo e exclui o componente conexo cuja área é menor que um limiar qualquer λ .

D. União de características

Para a união de características, faz-se o uso das posições (x, y) computadas dos componentes conexos da detecção de cabelo e pele. Isso é feito devido a o fato das posições computadas das características representarem um retângulo que contém o componente conexo. Onde para cada componente conexo em *SkinDetect* é verificado se há intersecção com algum componente conexo em *HairDetect*. Caso afirmativo, o algoritmo considera que achou uma face. Na figura 2 são mostradas as relações de intersecção que são consideradas no algoritmo por simplicidade.

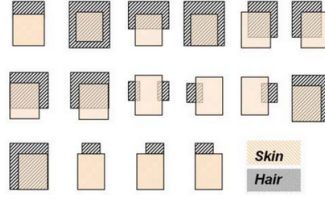


Fig. 2. Relações de intersecção entre retângulos que contém cabelo e pele, fonte: [1]

III. METODOLOGIA

A réplica dos conceitos apresentados em [1] foram feitas utilizando python 3.6. Também foram utilizadas funções das bibliotecas python OpenCV e numpy. As primitivas F_1 , F_2 , θ , $White$, Hue , I foram implementadas utilizando lambda expressões por simplicidade. Assim como na seção II haverá uma explanação diferente nas etapas de detecção de cabelo e detecção de pele. Isso acontecerá devido a fato de abordagens diferentes terem sido utilizadas tanto na réplica quanto na adaptação do código para que os resultados obtidos fossem parecidos ou até melhores.

A. Réplica: detecção de pele

Antes de se aplicar as primitivas de detecção de pele foi efetuada uma modificação do método de normalização. Primeiramente todos os valores contidos no array que representava a imagem foram divididos por 255, chamado neste trabalho de normalização geral. Somente em seguida é que foi aplicada uma normalização relativa a os valores das cores do pixel avaliado, segundo a Eq. (1). Isso foi feito pois, segundo observações experimentais, áreas pertencentes a um elemento de pele não estavam sendo detectadas como pelo algoritmo. Aplicando somente a normalização relativa pixel a pixel, o intervalo de valores de regiões com tonalidades de cor de pele não se encontravam nos limites das primitivas da subseção II-A.

Outra modificação aplicada foi a comparação da equação (7) em relação a o valor de Hue . O ângulo é retornado com valores em radianos não em graus. Sendo o maior e o menor ângulo da comparação na equação $\frac{4}{3}\pi$ e $\frac{\pi}{4}$, 240° e 45° respectivamente. Diferentemente do ângulo 20° apreentado na equação (8), foi empiricamente observado que o ângulo 45° resultava em melhores detecções de regiões de pele.

Sejam os valores r, g, b os valores dos canais RGB da imagem após a normalização geral seguida da normalização relativa pixel a pixel. Os mesmos foram utilizados como entrada das primitivas $\theta(r, g, b)$, $Hue(b, g, \theta)$ e $I(r, g, b)$. A comparação resultante das modificações da etapa de detecção de pele ficou como em Eq (9).

$$SkinDetect = \begin{cases} 1, & \text{if } (g < F_1(r) \cap g > F_2(r) \cap White(r, g) > 0.001 \cap (Hue(b, g, \theta) > \frac{4}{3}\pi \cup Hue(b, g, \theta) \leq \frac{\pi}{4})) \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

B. Réplica: detecção de cabelo

A mesma estratégia de aplicar uma normalização geral e normalização relativa pixel a pixel foram aplicadas na etapa de detecção de cabelo. Porém na comparação da equação (8) foram efetuadas modificações, tanto na comparação dos valores de *Hue* quanto na comparação dos valores de *I* e nas diferenças entre os canais $B - G$ e $B - R$. Primeiramente, os ângulos dos limites inferior e superior utilizados na comparação dos valores de *Hue* foram $\frac{\pi}{4}$ e $\frac{\pi}{2}$, 45° e 90° respectivamente. Diferentemente dos 20° e 40° mostrados na equação (8), foi empiricamente observado que os ângulos 45° e 90° resultavam em detecções melhores de regiões de cabelo.

Assim como na réplica da detecção de pele, os valores normalizados r, g, b , também serviram de entrada das primitivas *Hue*, *I* e θ utilizadas. Os mesmos também foram utilizados na computação das diferenças entre o canal azul e os canais verde e vermelho. Dado o fato de que os valores estão normalizados, os valores comparados com aqueles que as primitivas e as diferenças computam foram divididos por 255. Empiricamente foi observado que esta ação resultava em boas detecções de cabelo. A comparação resultante das alterações ficou como em Eq. (10).

$$HairDetect = \begin{cases} 1, & \text{if } ((I(r, g, b) < 0.313 \cap ((b - g) < 0.0588 \cup (b - r) < 0.0588)) \cup (\frac{\pi}{4} < Hue(b, g, \theta) \leq \frac{\pi}{2})) \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

C. Réplica: quantização de cabelo e pele

Uma vez obtidas as binarizações das imagens segundo a detecções de pele e cabelo, faz-se uma computação de componentes conexos junto com suas características básicas, a partir da chamada do procedimento *connectedComponentsWithStats(binaryImage)*, onde *binaryImage* é a imagem binarizada.

A única modificação quanto a etapa de quantização tanto de pele quanto de cabelo foi feita quanto a aplicação das operações morfológicas das imagens binarizadas. Para operação morfológica na binarização da detecção de pele o tamanho do elemento estruturante foi modificado de 5×5 para 10×10 . E foram efetuadas duas operações de erosão seguidas de duas operações de dilatação. Foi empiricamente observado que tais sequências de operações resultavam numa melhor remoção de ruído de detecção de pele.

Para binarização resultante da detecção de cabelo, o elemento estruturante foi modificado de 5×5 para 3×3 . Foi aplicada apenas uma operação de abertura. Semelhantemente, foi empiricamente observado que tais modificações resultavam numa melhor remoção do ruído de detecção de cabelo.

D. Réplica: união de características

Uma vez computadas as características dos componentes conexos da quantização de cabelo e face passa-se para a etapa de união de características. O algoritmo proposto em [1] considera que caso não haja intersecção então não foi detectada nenhuma face. Das características computadas a principal é o retângulo que contém o componente conexo que representa uma detecção de cabelo ou de face.

Na réplica, ao invés de simplesmente verificar a existência de intersecção entre retângulos a partir dos pontos. Foi efetuado o cálculo da área da intersecção entre os retângulos que contém os componentes conexos da detecção de cabelo e detecção de pele. Desse modo é possível distinguir entre intersecções maiores entre retângulos, e remover intersecções resultantes de detecções espúrias resultantes de ruídos que conseguiram passar pela etapa de quantização. O cálculo das posições (x, y) dos vértices do retângulo de intersecção foi feito segundo Eq.(11).

$$\begin{cases} x_1 = \max(hairL.x, skinL.x) \\ y_1 = \max(hairL.y, skinL.y) \\ x_2 = \min(hairR.x, skinR.x) \\ y_2 = \min(hairR.y, skinR.y) \end{cases} \quad (11)$$

Onde *hairL* e *skinL* são os vértices superiores mais a esquerda dos retângulos que contém os componentes conexos de detecção de cabelo e pele respectivamente. E *hairR* e *skinR* os vértices inferiores mais a direita.

IV. RESULTADOS

O algoritmo proposto é extremamente dependente das cores da imagem. Apesar do uso das cores como característica de detecção ser simples de utilizar, seu uso pode resultar em detecções espúrias. Principalmente em regiões que não pertencem a pele ou cabelo do indivíduo, mas são detectadas como tal pois possuem uma tonalidade de cor parecida com as que constam no intervalo de cores das primitivas implementadas.

Na figura 4 é possível notar detecções espúrias relacionadas a dependência de cores do algoritmo. Apesar do indivíduo detectado não ter cabelo, o algoritmo detectou a face do mesmo devido a intersecção de componentes de cabelo ao fundo da imagem e componentes de face que estão próximos.

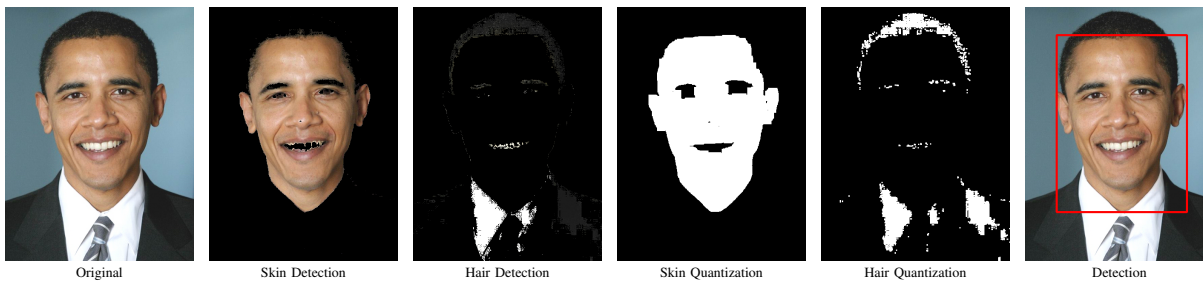


Fig. 3. Resultados de detecção de pele e cabelos segundo as etapas descritas na seção de metodologia

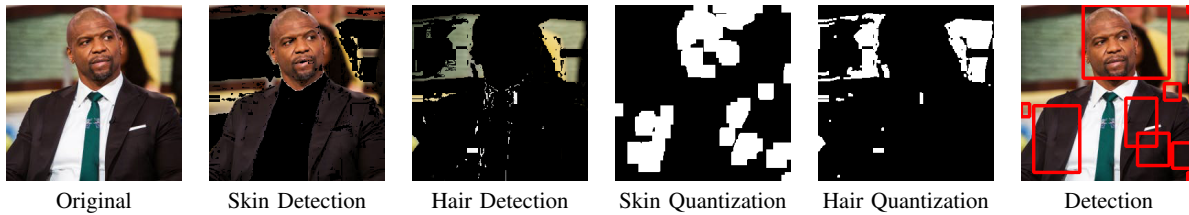


Fig. 4. Resultados espúrios de detecção de pele e cabelos segundo as etapas descritas na seção de metodologia

Algo similar acontece quando se trata de regiões que são identificadas como face simplesmente por ter uma tonalidade de cor cujo valor se encontra no intervalo das primitivas implementadas. Na figura 5 é possível observar que a detecção de uma região muito grande como se fosse um rosto. Também é possível notar a existência de detecções que não pertencem a um rosto. Mas que foram detectadas como tal devido a computação de interseções de regiões detectadas como cabelo estavam muito próximas das mesmas.

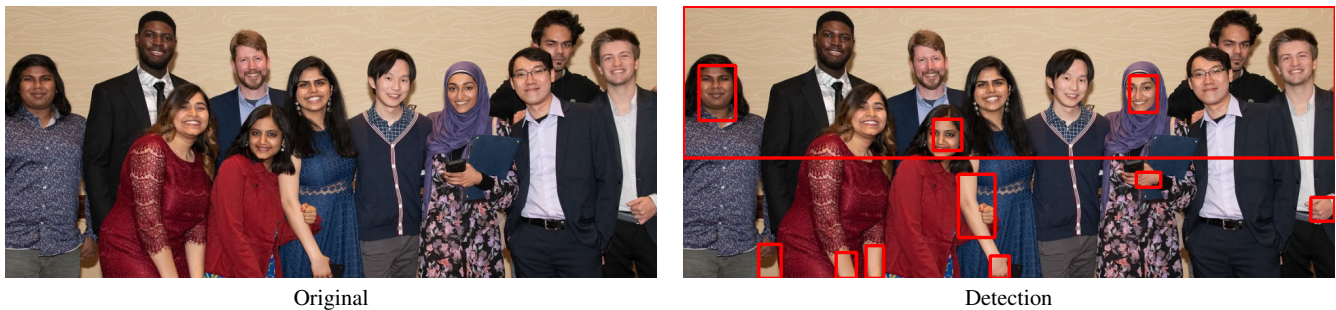


Fig. 5. Resultados espúrios de detecção multiface

V. TRABALHOS FUTUROS

O algoritmo proposto em [1] tem como objetivo detectar faces de acordo com o mínimo possível de características. Porém por ser altamente dependente das cores presentes na imagem, mesmo pode acabar resultando em outliers muito distantes do que seria uma detecção de face ideal. Porém tal algoritmo pode ser como uma heurística para melhorar algoritmos de aprendizagem de máquina que efetuam detecção de face. Os algoritmos mais simples geralmente efetuam uma varredura completa na imagem para poder propagar os mapas de características ao longo das camadas. Um possível trabalho futuro seria investigar se utilização deste algoritmo para definir regiões de interesse deixaria outros algoritmos de aprendizagem de máquina para detecção de face mais eficientes e precisos.

REFERENCES

- [1] Y.-J. Chen and Y.-C. Lin, "Simple face-detection algorithm based on minimum facial features," in *IECON 2007-33rd Annual Conference of the IEEE Industrial Electronics Society*. IEEE, 2007, pp. 455–460.
- [2] N. A. Ibraheem, M. M. Hasan, R. Z. Khan, and P. K. Mishra, "Understanding color models: a review," *ARNP Journal of science and technology*, vol. 2, no. 3, pp. 265–275, 2012.
- [3] M. Loesdau, S. Chabrier, and A. Gabillon, "Chromatic indices in the normalized rgb color space," in *2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. IEEE, 2017, pp. 1–8.
- [4] M. Soriano, B. Martinkauppi, S. Huovinen, and M. Laaksonen, "Using the skin locus to cope with changing illumination conditions in color-based face tracking," in *IEEE Nordic Signal Processing Symposium*, vol. 38, 2000, pp. 383–386.