RECONHECIMENTO DE OBJETOS COLORIDOS E MÃOS USANDO CORES E FORMAS

Victor Hugo Masutani[1], Almir Olivette Artero[2], Leandro Luiz de Almeida[1,3], Francisco Assis da Silva[1,3]

[1]Faculdade de Informática da Universidade do Oeste Paulista (UNOESTE), Presidente Prudente, SP. [2]Faculdade de Ciências e Tecnologia da Universidade Estadual Paulista (UNESP), Presidente Prudente/S. [3]Escola de Engenharia de São Carlos da Universidade de São Paulo (USP), São Carlos, SP. e-mail: chico@unoeste.br

RESUMO
Este artigo apresenta uma nova interface homem-computador, que através de um hardware simples e de baixo custo (webcam), captura imagens e realiza o reconhecimento de objetos coloridos ou de mãos a partir de informações de cores e formas, tornando possível o controle de aplicações no ambiente gráfico do sistema operacional.
Palavras-chave: Visão computacional, fecho convexo, modelo de cor YCrCb e HSV.

RECOGNITION OF OBJECTS AND COLOURED HANDS USING COLORS AND SHAPES

ABSTRACT
This paper describes a new human-computer interaction, which through a simple and low-cost hardware (webcam), captures images and performs the color objects recognition or hand recognition from informations of color and shapes, making possible the control of applications in the operational system graphical user interface.
Keywords: Computer vision, convex hull, color model YCrCb and HSV.

1. INTRODUÇÃO

O conceito de interface se refere à maneira em que dispositivos, como os computadores podem se comunicar e interagir. Ainda que o estudo da Interface Homem-Máquina tenha iniciado mais formalmente nos anos 70, a plataforma de computação estabilizou-se em um conjunto básico de interação com o computador (monitor, mouse e teclado) (MYERS et al., 2000). No entanto, mais recentemente, a sofisticação na parte de hardware evoluiu consideravelmente, trazendo novas formas de interfaceamento que podem trazer aos usuários uma forma mais simples e comum de comunicação, como por exemplo, usando a detecção de movimentos (gestos) e reconhecimento de voz. Este é o caso do videogame da MICROSOFT (2009), que é controlado através de detecções de movimentos, capturados por uma câmera e um sensor de profundidade, tornando a interação mais natural e imersiva.

A proposta deste projeto é fazer a captura de imagens através de uma webcam e realizar o reconhecimento de objetos coloridos ou das mãos e, a partir deste rastreamento, tornar possível o controle de aplicações no computador. O reconhecimento dos objetos é feito por meio da identificação da cor do objeto e o reconhecimento das mãos além da cor de pele também a forma das mãos é analisada.

As demais seções deste trabalho estão organizadas da seguinte maneira: na Seção 2 é apresentada uma introdução ao processamento digital de imagens, com ênfase nas técnicas usadas neste projeto; Na Seção 3 é apresentada uma introdução à visão computacional, com ênfase nas técnicas usadas neste trabalho, usando a biblioteca OpenCV; Na Seção 4 e 5 são apresentadas a metodologia empregadas no desenvolvimento deste trabalho, bem como experimentos; Por fim, a Seção 6 apresenta as conclusões.

2. PROCESSAMENTO DIGITAL DE IMAGENS

Segundo Gonzalez et al. (2000) uma imagem é definida como uma função bidimensional *f(x,y),* onde *x* e *y* são coordenadas espaciais e a amplitude de *f* é chamada intensidade da imagem no local. Quando *x*, *y* e a amplitude são valores finitos e discretos, diz-se que a imagem é digital. Caso contrário, tem-se uma imagem analógica.

Uma imagem digital pode ser considerada uma matriz, cujos índices de linhas e colunas identificam um ponto na imagem e o correspondente valor do elemento da matriz identifica o nível de cinza naquele ponto. Os elementos dessa matriz digital são chamados de elementos da imagem ou pixels (GONZALEZ; et al., 2000). Nas imagens digitais coloridas, cada pixel guarda mais de um valor, de acordo com o modelo de cor utilizado na representação da imagem.

2.1 Vizinhança de pixels

Como uma imagem digital é formada por pixels dispostos sobre uma malha retangular, cada pixel possui uma vizinhança. Assim, um pixel nas coordenadas x e y, possuem quatro vizinhos nas coordenadas (x+1,y), (x-1,y), (x,y+1) e (x,y-1). Esse conjunto de pixels é chamado de vizinhança-4 ou viz-4, como ilustrado na Figura 1(a), onde 1 indica um vizinho do pixel X e 0 os pixels que não estão em Viz-4. Em (b) tem-se a vizinhança-8 de X.

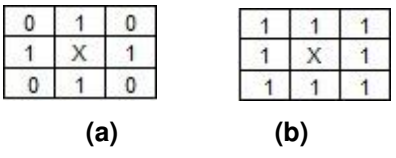


Figura 1. Vizinhança do pixel X.

2.2 Modelos de cor

A percepção e interpretação de uma cor é um fenômeno fisiopsicológico, que não é completamente entendido. As cores que os humanos e alguns animais percebem são determinadas pela natureza da luz refletida no objeto. A finalidade de vários modelos de cores é facilitar a especificação de cores em alguma forma padrão e de aceite geral. Cada cor deve ser representada em um sistema tridimensional por somente um ponto (GONZALEZ et al., 2000).

2.2.1 RGB

O modelo RGB (*red*, *green*, *blue*) utiliza as cores primárias vermelho, verde e azul e seu subespaço é o cubo mostrado na Figura 2. Cada eixo está normalizado e varia entre 0 e 1. O vértice que está na diagonal, liga o preto ao branco e, se refere aos tons de cinza neste modelo. Outras cores são produzidas quando as cores primárias se combinam de diferentes maneiras.

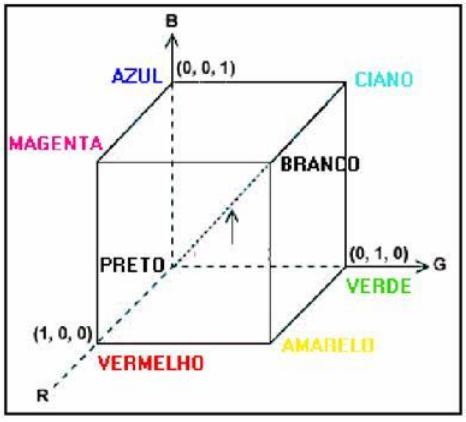


Figura 2. Cubo de cores do modelo RGB.

HSV

O modelo HSV (*hue, saturation, value*) representa as componentes de tonalidade, saturação e brilho. O subespaço do HSV é o hexágono mostrado na Figura 4. A saturação é dada pela distância a partir do eixo central (0 – 100), a tonalidade pela distancia do ângulo a partir do vermelho (360º) e o brilho varia de 0 a 1 a partir da parte de baixo. O modelo HSV é utilizado, quando é necessário identificar cores que são muito parecidas com outras cores (CARDANI, 2001). Pois, o determinante da cor no modelo HSV é a tonalidade, diferente do RGB que os determinantes da cor são todos os parâmetros (vermelho, verde e azul), mesmo utilizando a distância euclidiana entre duas cores no cubo RGB pode resultar em que não sejam realmente parecidas.

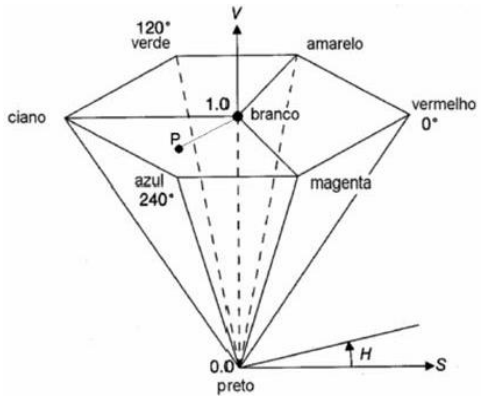


Figura 3. Hexágono de cores HSV.

2.2.2 Conversão de RGB para HSV

O modelo de cores utilizado pela maioria dos dispositivos de captura de imagem (incluindo o monitor de vídeo e a Webcam) é o RGB. Então é necessário fazer a conversão de uma imagem no modelo de cores RGB para o modelo de cores HSV. As equações de conversão são mostradas nas Equações 1.

$$H = \begin{cases} \theta & se\, B \le G \\ 360 - \theta & se\, B > G \end{cases}$$

com

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

(1)

$$S = 1 - \frac{3}{(R+G+B)}[\min(R,G,B)]$$

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

2.2.3 YCrCb

O modelo YCrCb representa usa uma componente de luminância (Y) e duas componentes de crominância (Cr e Cb) e, tem sido o modelo mais recomendado para realizar o reconhecimento de cor de pele. Uma vez que o tom de pele depende da iluminação, a transformação não linear do modelo de cor YCrCb faz com que o tom de pele não dependa da iluminação (HSU et al., 2002).

2.2.4 Conversão de RGB para YCrCb

A conversão das imagens capturadas no modelo RGB para o modelo YCrCb é realizada utilizando a Equação 2.

$$\begin{bmatrix} Y \\ Cr \\ Cb \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.288 & 0.436 \\ 0.615 & -0.514 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2)$$

2.3 Reconhecimento de cores

A partir de uma imagem capturada por um dispositivo de entrada, é possível o reconhecimento de cores. Cada ponto da matriz digital é processado e verificado se está próximo da cor a ser encontrada e, então, classificado como pixel da cor desejada. A diferença das cores dos pixels pode ser usada para que o reconhecimento de cores tenha uma tolerância na identificação, o que pode ser feito usando a distância euclidiana entre as duas cores C_1 = (R_1,G_1,B_1) e C_2 = (R_2,G_2,B_2). A distância euclidiana é dada pela Equação 3.

$$d = \sqrt{(R_1 - R_2)^2 + (G_1 - G_2)^2 + (B_1 - B_2)^2} \quad (3)$$

Para a identificação da cor de objetos coloridos, o modelo dado pela Equação 3 se mostra adequado, mas para a identificação da cor de pele, não tem se mostrado, apresentando resultados ruins, por causa da grande variação de tons de pele. Outro modelo mais eficaz (Equação 4) é apresentado por KOVAČ et al. (2003), que utiliza o modelo de cor YCrCb em seus cálculos.

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos\theta & sen\theta \\ -sen\theta & \cos\theta \end{bmatrix} \begin{bmatrix} Cb & cx \\ Cr & cy \end{bmatrix} \quad (4)$$

$$\frac{(x - e_{cx})^2}{a^2} + \frac{(y - e_{cy})^2}{b^2} \leq 1$$

onde: c_x=109.38, c_y=152.02, θ=2.53 radianos, e_{cx}=1.6, e_{cy}=2.41, a=25.39 e b=14.03.

2.4 Morfologia matemática

A morfologia matemática consiste em extrair componentes que são importantes para a representação e descrição de formas, bem como utilizado para pré- ou pós-processamento de imagens (GONZALEZ et al., 2000). A morfologia matemática possui algumas operações como: Dilatação, Erosão, Abertura e Fechamento; que são ilustradas na Figura 4.

2.5 Contornos, fecho convexo e defeitos convexos

O contorno de um objeto em uma imagem define importantes características deste objeto, sendo capaz de auxiliar a sua identificação. Em geral, é obtido através das bordas encontradas na imagem.

Importantes descrições dos objetos podem ser obtidas através do fecho convexo e dos defeitos convexos, que consistem respectivamente, no menor polígono convexo que contem os pontos do objeto (O'ROURKE, 1998) e sequência dos pontos do contorno entre dois vértices do fecho convexo (MANRESA et al., 2000).

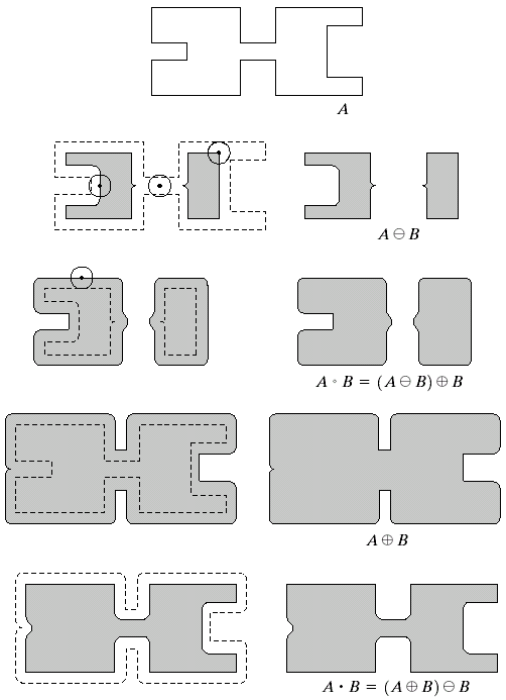


Figura 4. Processo das operações de Erosão, Fechamento, Abertura e Fechamento.

3. VISÃO COMPUTACIONAL

Visão computacional é a extração de dados a partir de imagens. O processo consiste em analisar uma imagem e a partir de suas características, extrair medidas qualitativas e quantitativas, como por exemplo, se existe uma pessoa nessa imagem (BRADSKI; KAEBLER, 2008).

3.1 Biblioteca de visão computacional OpenCV

A biblioteca OpenCV (*Open Source Computer Vision*) (Intel Corporation, 1999) possui uma grande variedade de funções para uso em visão computacional. Escrita na linguagem C/C++, esta biblioteca surgiu por causa do grande número de aplicações nesta área. A biblioteca OpenCV foi projetado para ser computacionalmente eficiente e poder trabalhar com aplicações em tempo real. Dentre as funções usadas neste trabalho estão:

cvNamedWindow – cria uma janela onde será capturado o vídeo;

cvCaptureFromCAM – função que retorna o dispositivo de captura esteja conectado ao computador, retornando falso caso contrário;

cvQueryFrame – função que retorna frames capturados. O retorno é uma imagem colorida, que possui um registro que armazena uma matriz com as dimensões [0..Altura, 0..Largura, [R,G,B]];

cvFlip – faz o espelhamento de uma imagem;

cvCvtColor – função que converte uma imagem de um modelo de cor para outro;

cvPtr2D – função que retorna as cores de um determinado pixel (x, y) da imagem;

cvErode – função que aplica a operação de erosão da morfologia matemática;

cvDilate – função que aplica a operação de dilatação da morfologia matemática;

cvMorphologyEx – função que aplica a operação de abertura ou fechamento da morfologia matemática;

cvFindContours – função que identifica contornos em imagens;

cvConvexHull – função que encontra o fecho convexo;

cvConvexityDefects - função que identifica os defeitos convexos dado um fecho convexo por parâmetro, bem como o ponto de maior distância.

3.2 Rotulação de componentes conexos

Rotulação de componentes conexos (*connected component labeling*) é uma técnica que analisa uma imagem e extrai dela grupos de pixels, baseado em sua conectividade e classifica cada agrupamento de pixels de uma cor. Ela funciona analisando a imagem pixel por pixel do topo até embaixo e da esquerda para a direita a fim de identificar os pixels adjacentes e compartilham a mesma cor, utilizando o viz-4 ou viz-8. Ele percorre sequencialmente a imagem a procura de cada pixel que não é fundo. Então, ele incrementa o rótulo chama a rotina auxiliar

recursiva que marca o corrente pixel com o corrente rótulo e procura por pixels ainda não rotulados em sua vizinhança. Se encontrar, a rotina chama a si própria, rotula o pixel com o mesmo rótulo e continua a procurar pixels ainda não rotulados em sua vizinhança. Se não encontra, a rotina volta para onde ela foi chamada e recomeça a procurar. A rotulação de componentes conexos é uma etapa indispensável e de fundamental importância em aplicações na área de Visão Computacional. Por meio da rotulação de componentes conectados é possível se faça o reconhecimento de um ou mais objetos em uma imagem, classificando cada agrupamento reconhecido com um rótulo (SANTIAGO, 2009).

A Figura 5 mostra em (a) os alvos reconhecidos e em (b) os alvos rotulados.

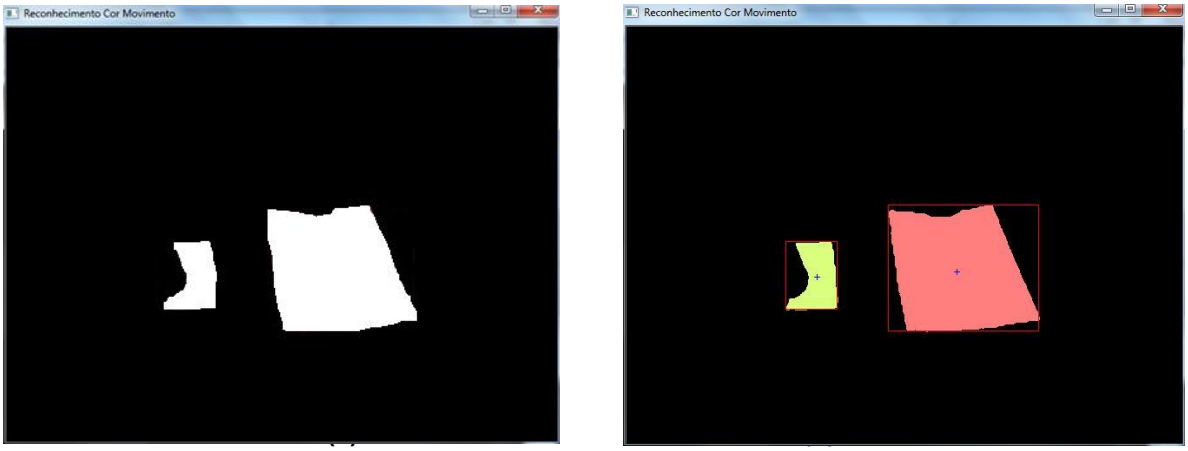


Figura 4. Sequência de execução para o reconhecimento de objetos coloridos.

4. RECONHECIMENTO DE OBJETOS COLORIDOS

O algoritmo proposto realiza a captura das imagens da Webcam e em seguida, as estas são convertidas de RGB para HSV. No próximo passo, são identificados os pixels que possuem cores similares ao alvo desejado (um objeto colorido usado pelo usuário, como uma caneta), com o uso da Equação 3. Na etapa seguinte, é realizado um tratamento para a retirada de ruídos (pixels isolados) e fechamento de orifícios na imagem segmentada, sendo aplicadas operações de Abertura e Fechamento da Morfologia Matemática. Na sequência, os pixels são rotulados, de modo a identificar os objetos na cena (determinação dos elementos conexos). Nesta etapa, é identificado o alvo desejado, o que é feito a partir de uma análise de transições de cenas, para que objetos com a mesma cor presentes na cena não sejam confundidos com o alvo. Por fim, os deslocamentos do alvo são tratados como movimentos do mouse, sendo usados para controlar aplicações no ambiente do sistema operacional. As etapas são mostradas na Figura 6.

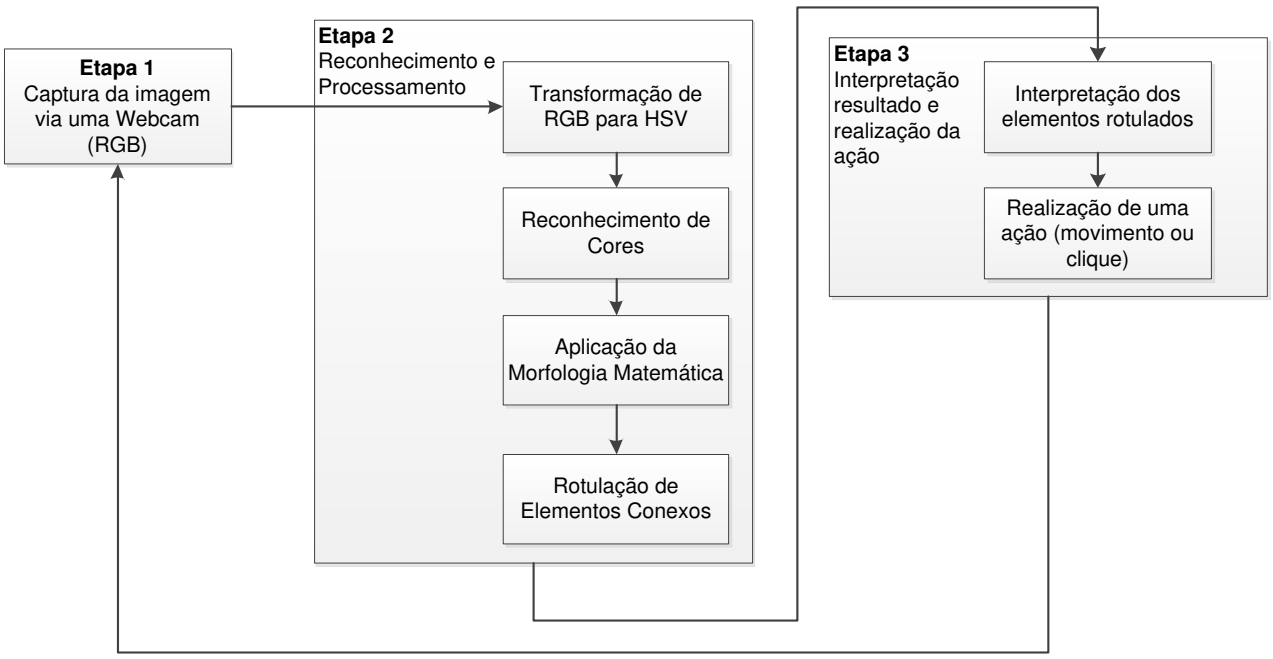


Figura 6. Sequência de execução para o reconhecimento de objetos coloridos.

A Figura 7 (a) mostra um exemplo de imagem capturada pela webcam e, em (b) tem-se a imagem com elemento reconhecido e rotulado (cor alvo).

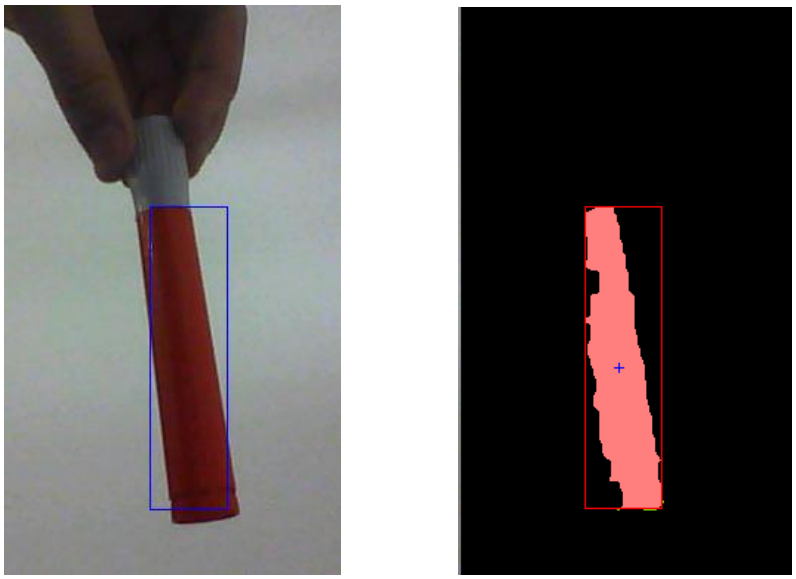


Figura 7. Imagem capturada pela webcam (a) e a cor vermelha sendo reconhecida (b).

5. RECONHECIMENTO DAS MÃOS

Com o objetivo de se eliminar o uso de um objeto como alvo foi realizado o reconhecimento da mão, o qual captura de imagens da Webcam e, em seguida, estas são convertidas de RGB para YCrCb, afim de reduzir o erro no reconhecimento da cor da pele. No próximo passo, os pixels são classificados como pele ou não pele, com a aplicação da Equação 4. Na etapa seguinte, é realizado um tratamento para a retirada de ruídos (pixels isolados), o que é feito usando as operações morfológicas de Abertura e Fechamento. Na sequência são identificados os contornos dos objetos na imagem e, o maior contorno define o objeto de interesse (mão), sendo aplicado a ele o algoritmo de fecho

convexo. Em seguida, são encontrados os defeitos convexos e, caso dois defeitos convexos consecutivos com ângulo entre 15º e 40º sejam encontrados, o objeto dentro do fecho convexo é confirmado como sendo uma mão. Por fim, os deslocamentos da mão são tratados como movimentos do mouse, sendo usados para controlar aplicações no ambiente do sistema operacional. Estas etapas são mostradas na Figura 8.

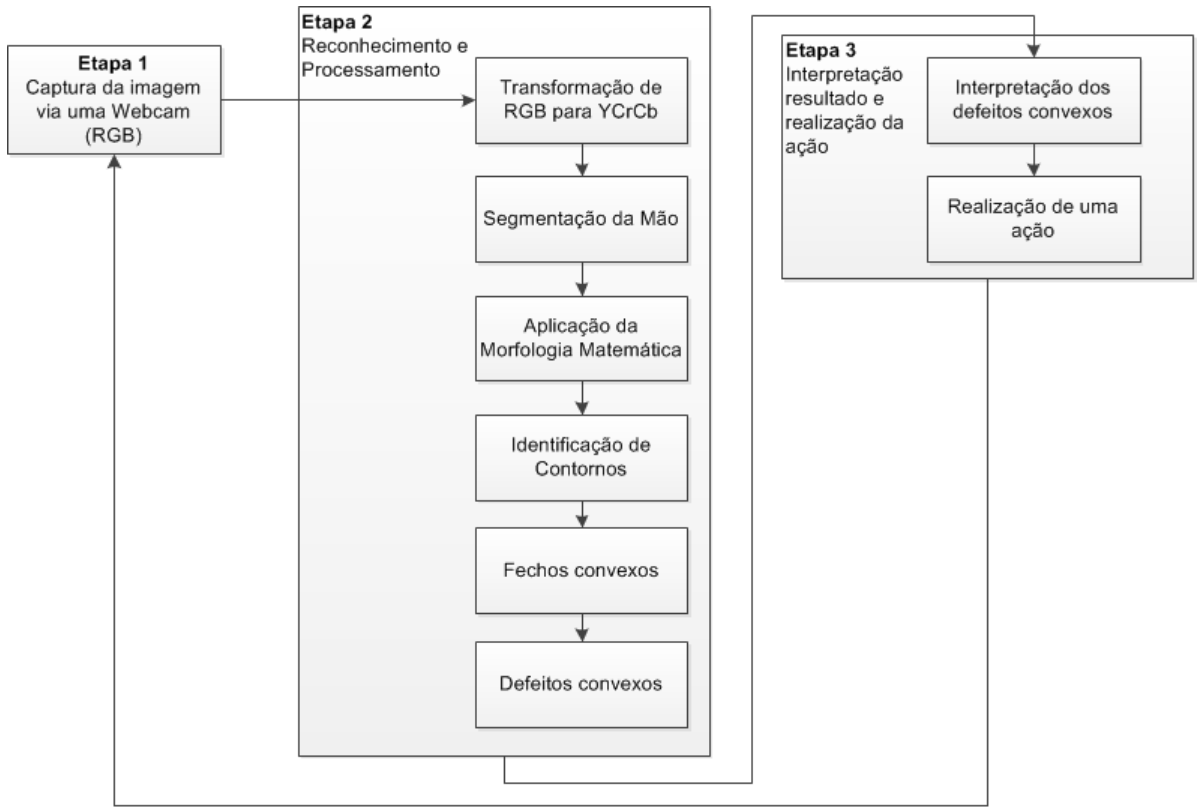
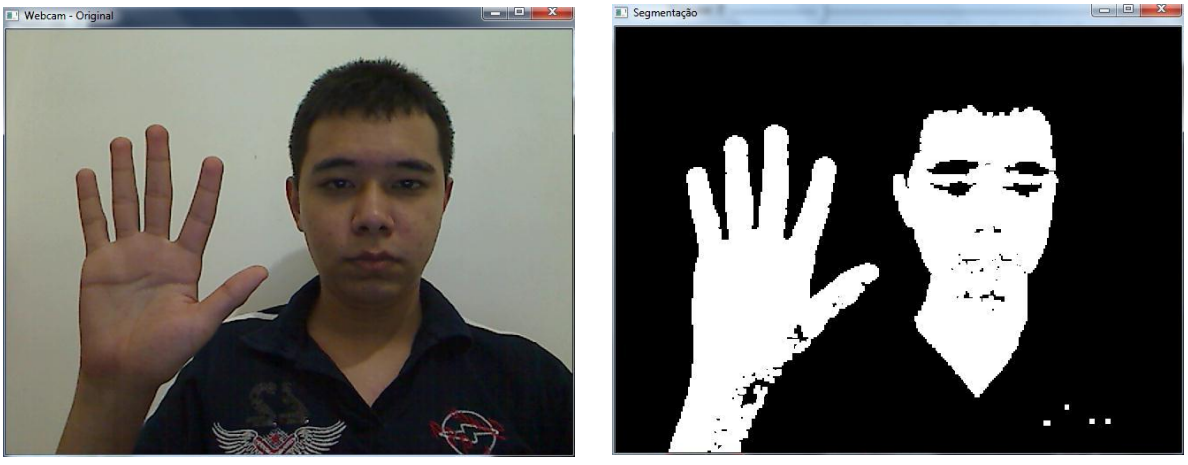


Figura 8. Sequência de execução para o reconhecimento da mão.

A Figura 9 (a) mostra um exemplo de imagem capturada pela Webcam e, em (b) tem-se a cor de pele sendo reconhecida. Para a realização deste teste foi utilizada uma webcam que captura quadros de imagens com a resolução de 640x480 pixels.



(a) **(b)**

Figura 9. (a) Imagem original; (b) Reconhecimento das áreas correspondentes à pele.

Embora tenha sido usada uma câmera de baixo custo, esta figura apresenta um resultado bem satisfatório para o reconhecimento da mão, o que é feito na Figura 10, que mostra a obtenção do fecho e dos menores pontos dos defeitos convexos, a partir da extração dos contornos, identificação do fecho convexo e reconhecimento dos defeitos convexos.

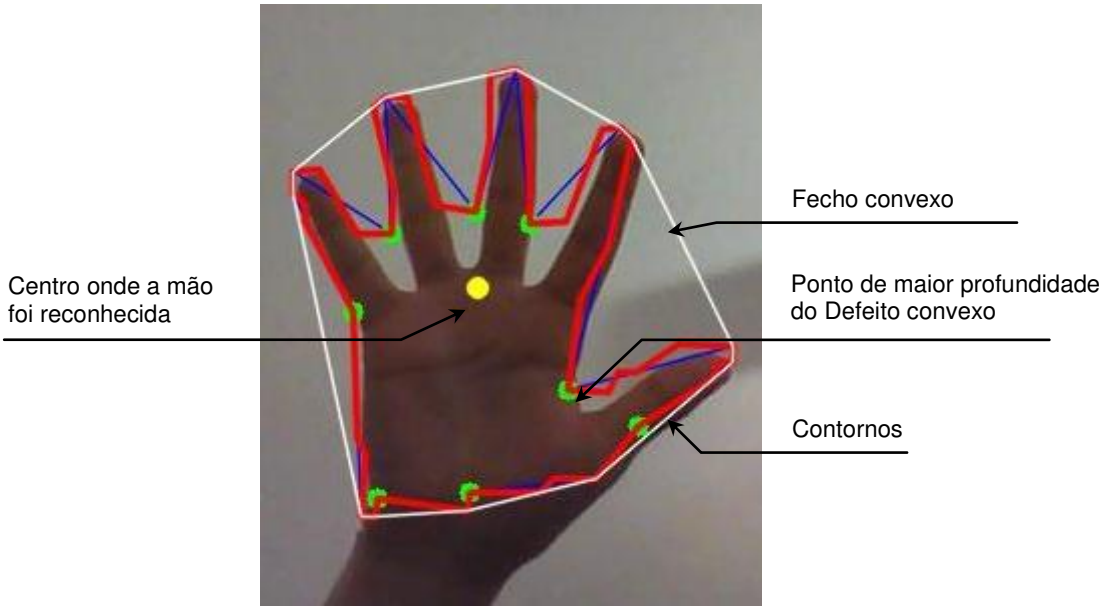
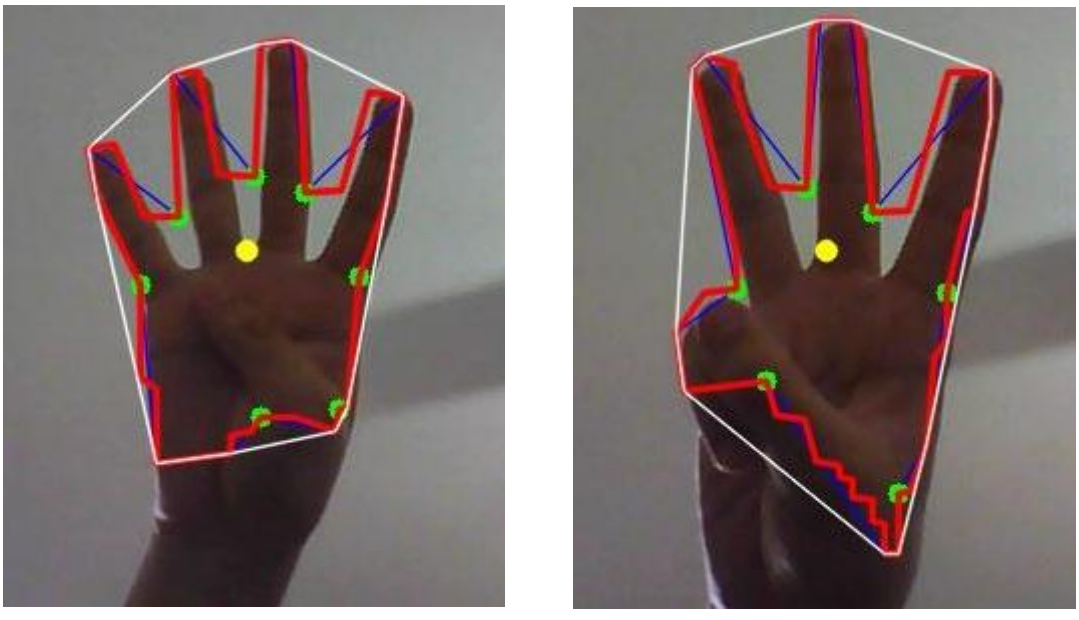


Figura 10. Contornos, fecho e defeito convexos.

5.1 Experimentos

A Figura 11 mostra que exibindo três (a) ou quatro (b) dedos, a mão continua sendo reconhecida, mesmo não havendo o mesmo número de pontos de profundidade reconhecidos pelo defeito convexo.



(a) **(b)**

Figura 11. (a) Quatro dedos; (b) Três dedos.

A Figura 12 mostra que exibindo apenas dois dedos, a mão não é reconhecida, pois existem insuficientes defeitos convexos para a sua caracterização, visto que são necessários dois pontos de profundidade consecutivos entre o ângulo desejado.

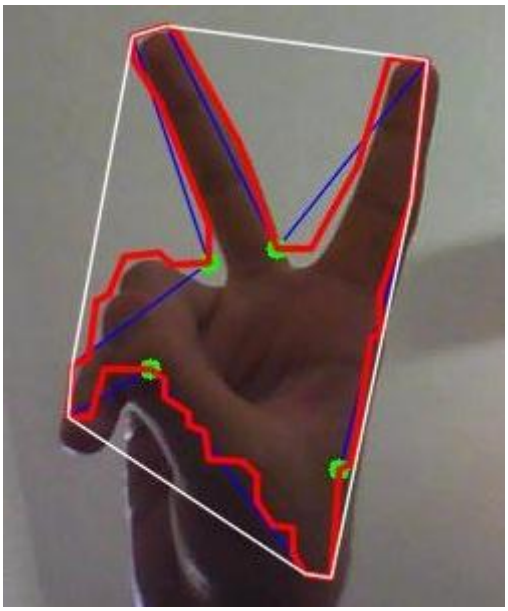


Figura 12. (a) Quatro dedos; (b) Três dedos.

A Figura 13 mostra esta interface sendo usada para controlar uma aplicação, através da movimentação da mão, que também poderia ser controlada utilizando um objeto colorido. Em (a) tem-se o computador, em (b) a Webcam, em (c) a mão do usuário e, em (d), a aplicação sendo executada.

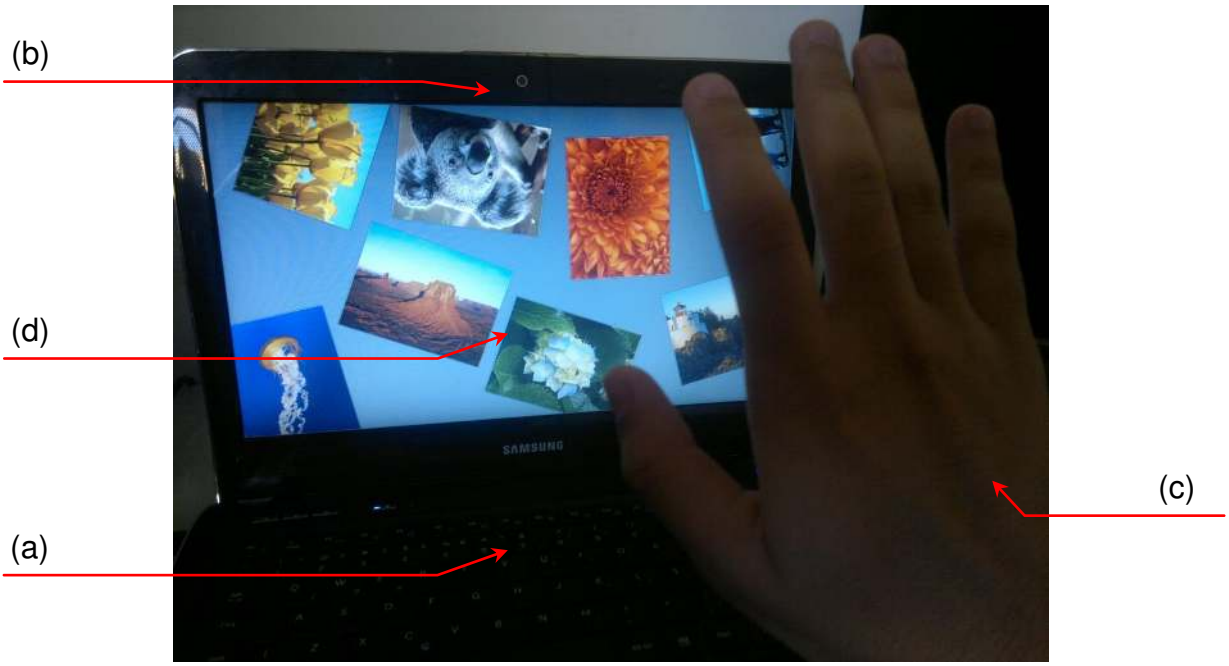


Figura 13. Aplicação utilizando o reconhecimento da mão.

6. CONCLUSÕES

Os resultados obtidos neste trabalho mostraram que, mesmo usando uma simples webcam, é possível interagir com o computador de uma forma bem mais intuitiva, além disso, o baixo custo deste equipamento torna possível a sua aplicação nas mais diversas situações, com o uso de objetos coloridos ou até a mão. Para o reconhecimento de objetos coloridos, o modelo de cores HSV mostra-se aceitável, mas não identificando cores que estão com uma baixa iluminação. Já o reconhecimento da cor de pele, embora o modelo de cores YCrCb adotado neste trabalho tenha sido proposto justamente para diminuir os efeitos da iluminação do ambiente, nota-se que este problema ainda persiste, não identificando toda a mão, porém, não chega a comprometer os resultados.

REFERÊNCIAS

BRADSKI, G.; KAEBLER, A. **Learning OpenCV, Computer Vision with the OpenCV Library**. Gravenstein Highway North, Canadá: OReilly, 2008.

CARDANI, D. **Adventures in HSV Space**, 2001. Disponível em: http://visl.technion.ac.il/labs/anat/hsvspace.pdf>. Acesso em 13 de março de 2011.

GONZALEZ, R. C.; WOODS, R. E.; CESAR JÚNIOR, R. M.; COSTA, L. F. **Processamento de imagens digitais**. São Paulo: E. Blucher, 2000.

HSU, R.-L.; ADBEL-MOTTALEB, M.; JAIN, A. K. **Face Detection in Color Images**. Dept. of Computer Science & Engineering, Michigan State University, 2002.

KOVAČ, J.; PEER, P.; SOLINA, F.. **Human Skin Colour Clustering for Face Detection.** University of Ljubljana, Slovenia, 2003.

MANRESA, C.; VARONA J.; MAS R.; PERALES F. J. **Real – Time Hand Tracking and Gesture Recognition for Human-Computer Interaction**, Janeiro, 2000.

MICROSOFT. Kinect for Xbox 360: **Science Fiction Comes to Your Living Room.** 2009. Disponível em: <http://www.microsoft.com/presspass/features/2010/nov10/11-03Kinect.mspx>. Acesso em 08 de março de 2011.

MYERS, B.; HUDSON, S.; PAUSCH, R. **Past, Present, and Future of User Interface Software Tools.** ACM Transactions on Computer-Human Interaction, Vol. 7, Número 1, Março 2000, p. 3–28.

O'ROURKE, J. **Computational Geometry in C.** 2.ed. Cambridge University Press, 1998. p. 63-65.

Recebido em: 05/06/2013

Revisado em: 25/06/2013

Aceito em: 01/07/2013