

# RASTREAMENTO DO MOVIMENTO DOS DEDOS DA MÃO HUMANA EM IMAGENS DE VÍDEO

Caroline Conti<sup>1</sup>, Adilson Gonzaga<sup>2</sup>

Laboratório de Visão Computacional (LAVI), Departamento de Engenharia Elétrica, Escola de Engenharia de São Carlos (EESC), Universidade de São Paulo (USP), 13566-590, São Carlos, SP, Brasil

<sup>1</sup>conti.carol@gmail.com, <sup>2</sup>agonzaga@sc.usp.br

**Resumo** Este artigo propõe uma metodologia de visão computacional para o rastreamento de imagens de mão humana na realização de gestos em frente a uma “webcam”, para que possa ser aplicado na interação homem-computador. O algoritmo foi baseado na subtração de fundo e segmentação do contorno da mão via aproximação de polígono mínimo. Para reconhecimento dos dedos, foi aplicado o algoritmo de *k*-curvatura. O resultado obtido em imagens de vídeo, demonstra a viabilidade de se utilizar a metodologia proposta na Interação Humano Computador, mesmo com câmeras e sistemas de baixo custo.

**Palavras-Chave:** rastreamento, segmentação de imagens de vídeo, gestos de mão.

**Abstract** This paper proposes a computer vision method for hand-fingers tracking from images captured by a low resolution webcam, in order to use as the basis of simple systems for human-computer interaction. The algorithm was based on segmentation by background subtraction, hand contour recognition from approximation for minimum-perimeter polygon and recognition of the fingertips by *k*-curvature algorithm. The system performance shows that it's possible to use the proposed methodology with a low cost hardware.

**Keywords** tracking, video image segmentation, hand gestures.

## 1. Introdução

A Interação Humano-Computador (IHC) é considerada um conceito novo, mas vem aumentando rapidamente e de forma constante nas últimas três décadas, atraindo profissionais de várias áreas e incorporando diversos conceitos e abordagens [1]. A inclusão digital bem como as grandes inovações nesta área abriu várias opções para que esta interação não ficasse limitada à utilização tradicional do monitor, teclado e *mouse*. Neste contexto, o conceito de visão computacional, rastreamento e reconhecimento de voz possibilitaram uma interação mais natural [2] e flexível entre o homem e o computador. A utilização da mão e de seus gestos é a maneira mais intuitiva para interação do homem com o computador. Atualmente os métodos mais eficientes utilizam alguns dispositivos baseados em sensores ópticos, magnéticos ou mecânicos ligados ao computador que são, em sua maioria, sofisticados (*data gloves*), necessitam de ambientes especiais para serem operados e são comercializados a um preço elevado. É possível e interessante que esta interação pelas mãos seja cada vez mais natural e independente de dispositivos que a pessoa deva vestir ou manipular, portanto, novas interfaces devem ser desenvolvidas. Recentemente há um crescente interesse em introduzir a visão computacional como meio de interação humano-computador. O rastreamento de gestos humanos representa importantes possibilidades e fatores como a proliferação atual de computadores de alta capacidade, as câmeras de boa qualidade a os baixos custos, além da crescente necessidade de análise automatizada de vídeo, geraram um maior interesse de desenvolver algoritmos de rastreamento [3]. O problema do rastreamento de um objeto pode ser definido como o problema de estimar a trajetória de um objeto no plano da imagem enquanto se move ao redor de uma cena [4].

No caso da mão humana, os principais desafios no rastreamento são:

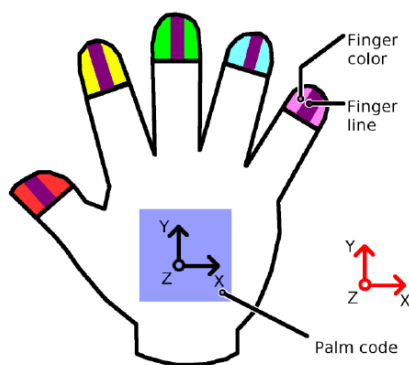
- a diferenciação entre as áreas de fundo e o primeiro plano;
- a diferenciação entre os objetos monitorados e outros objetos em movimento na mesma cena e extração de características relevantes ao rastreamento;
- mudanças na iluminação, que provocam mudanças nos objetos monitorados;
- normalização dos objetos que mudam com a distância da câmera;
- oclusões entre os objetos presentes na cena.

Este artigo propõe como objetivo principal, um método eficiente de reconhecimento e rastreamento de gestos dos dedos de uma mão humana a partir de imagens captadas por uma câmera (*webcam*) de baixa resolução, com o intuito de ser utilizado

como base para sistemas de Interação Humano-Computador. Neste contexto, o trabalho contempla um problema de visão computacional e sua solução desde a aquisição de imagens, segmentação, reconhecimento, rastreamento de gestos e interpretação dos resultados.

## 2. Trabalhos relacionados

Duca et al. [5] propuseram um algoritmo de rastreamento em tempo real dos movimentos dos dedos da mão em três dimensões. O algoritmo proposto é baseado no uso de apenas em uma *webcam* combinada a utilização de códigos para reconhecer a palma da mão e os dedos. Os dedos foram codificados por diferentes cores e com uma linha que divide cada dedo e identifica o eixo de cada um, como mostra a Figura 1. Dessa maneira, cada dedo é reconhecido facilmente em cada quadro. Utilizando uma figura quadrada na palma da mão e em sua face oposta eles conseguiram reconhecer o movimento de rotação da mão pela variação da área do quadrado. Esta solução, segundo os autores, tem a vantagem de rápido processamento em relação a outros algoritmos em três dimensões, e de necessitar de recursos mínimos, porém fica dependente da utilização destes marcadores, não sendo uma solução de grande usabilidade.



**Figura 1:** Modelo de codificação dos dedos para realização de rastreamento [5].

Letessier e Bérard [6] propõem um algoritmo para rastreamento dos dedos da mão baseado em segmentação por diferenciação da imagem (Image Differencing Segmentation) e filtros de rápida rejeição (Fast Rejection Filters). Eles utilizam critérios geométricos para caracterizar as pontas dos dedos e a partir de um filtro de forma (*shape filtering*) conseguem extrair a posição dos dedos. Cada ponta de dedo detectada em cada quadro é associada a uma ponta do quadro anterior utilizando como custo a restrição de velocidade máxima, isto é, as pontas correspondentes em cada quadro não podem se distanciar de uma região circular ao redor do objeto com um raio máximo definido. Segundo os autores este algoritmo possui a vantagem de ser insensível à orientação dos dedos e insensível a eventuais sombras existentes na imagem.

Fujii et al. [7] utilizaram uma câmera de infra-vermelho e uma estrutura contendo um espelho inclinado para detectar o movimento em três dimensões dos dedos. A utilização do infra-vermelho foi baseada no fato de que com esta câmera as condições de iluminação e de cor do objeto não são tão relevantes. Neste caso, o posicionamento dos dedos é dado pela combinação das imagens da mão captadas de forma direta e refletida. A extração do fundo da imagem é realizada de maneira simples pela utilização de um limiar.

Na maioria dos trabalhos propostos na literatura, no entanto, são necessários aparatos específicos para a aquisição da imagem ou para guiar o processamento e reconhecimento dos gestos. Nossa abordagem prevê o uso da “mão nua” e deverá ser capaz de acompanhar o movimento dos dedos em imagens de baixa resolução, como as geradas por simples *webcams* disponíveis em *notebooks*.

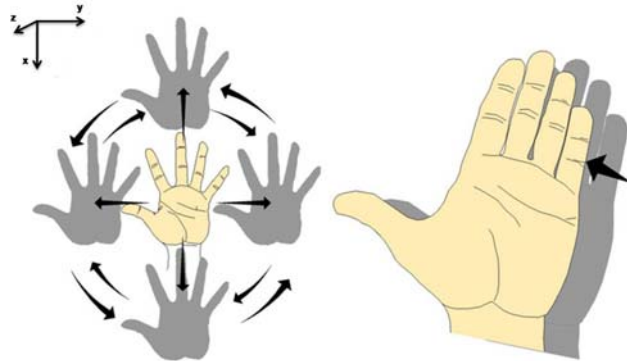
## 3. Material e Método

Para alcançar o objetivo proposto, algumas etapas foram definidas como base e serão abordadas separadamente na apresentação da metodologia. Estas etapas levaram em consideração alguns aspectos do trabalho realizado por Ribeiro [8],[9].

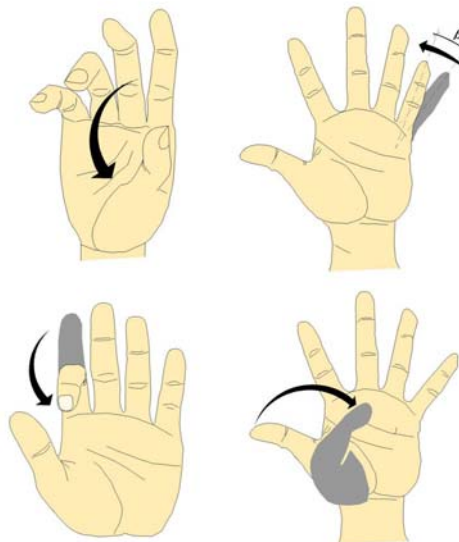
O sistema proposto possui uma *webcam* ligada a um computador para aquisição de imagens e tratamento dessas em tempo real com o intuito de rastrear o movimento de abrir e fechar a mão, o que significa abaixar e levantar os dedos frontalmente ao plano da câmera. O número de objetos presentes em cada quadro capturado é limitado a um, sendo este uma mão direita. O movimento da mão é limitado à translação em três direções (nas duas dimensões do plano da imagem e a distância da mão à *webcam* sendo o terceiro eixo). Os movimentos previstos da mão podem ser vistos na Figura 2. Nos movimentos dos dedos foram considerados os dois graus de liberdade, como mostrado na Figura 3. A *webcam* é fixa e a mão deve se movimentar a

uma distância máxima de um metro em relação à sua lente. Um painel de cor preta é colocado no fundo da cena com o intuito de criar um “background” simples.

O algoritmo foi desenvolvido utilizando MATLAB® e suas funções pertencentes ao *Image Aquisition Toolbox 3.0®* e *Image Processing Toolbox™*, além do *DIPUM Toolbox*, que acompanha o livro *Digital Image Processing Using MATLAB* [10].



**Figura 2:** Movimento previsto da mão no sistema proposto de rastreamento.



**Figura 3:** Movimento previsto dos dedos no sistema proposto de rastreamento.

Em relação à iluminação ambiente, foram considerados os casos de um ambiente fechado mas não isolado da iluminação externa, por exemplo um ambiente de laboratório com ou sem uma fonte de luz focalizada no objeto de interesse. O esquema de configuração física proposta pode ser visto na Figura 4.



**Figura 4:** Esquema de configuração física proposta para o rastreamento do movimento dos dedos.

Para o desenvolvimento do algoritmo e validação dos resultados, as imagens foram capturadas utilizando o espaço de cores RGB, com resolução de 352 x 288 pixels e com uma taxa de 15 a 22,5 quadros por segundo. A partir destas imagens, foram construídos arquivos de vídeos em formato AVI. Os vídeos foram capturados em ambiente com iluminação interna e superior à cena.

### 3.1 Segmentação da imagem

Foram avaliados seis métodos de segmentação baseados na cor da pele humana, para gerar a mão segmentada necessária para extração das características. O resultado desta avaliação é melhor discutido na Secção 4. A segmentação da mão foi realizada baseada na Subtração de Fundo (*Background*) [3] em espaço RGB. O método define como *background* a mediana de um vetor que acumula imagens no tempo, segmentando o movimento de uma cena. O *background* é encontrado pela mediana das imagens R, G e B separadamente. Esta imagem mediana de fundo é calculada utilizando-se os primeiros quadros do vídeo.

A cada quadro capturado, a imagem é separada em seus planos R, G e B, subtraída da imagem de fundo e a segmentação é realizada a partir de um limiar (*threshold*) escolhido empiricamente. Se a subtração resultar em um valor absoluto maior que o limiar definido, será classificado como objeto, caso contrário será classificado como fundo da imagem. O resultado é uma imagem binária. A Equação 1 mostra a condição de segmentação da imagem com limiar que obteve melhores resultados, na qual  $R_I$ ,  $G_I$  e  $B_I$  representam a imagem RGB a ser segmentada e  $R_B$ ,  $G_B$  e  $B_B$  a imagem de fundo.

$$\frac{(|R_I - R_B| + |G_I - G_B| + |B_I - B_B|)}{3} > 25 \quad (1)$$

A Figura 5 mostra uma mesma imagem segmentada com diferentes valores de limiar, na Figura 5a uma imagem segmentada com limiar menor que o limiar escolhido, na Figura 5b a imagem resultante da segmentação utilizando o limiar escolhido e na Figura 5c a imagem resultante de uma segmentação usando limiar maior.

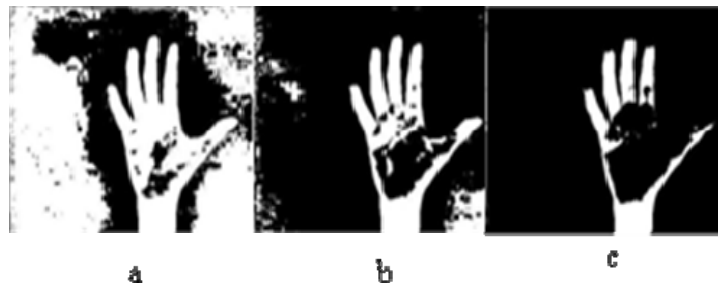


Figura 5: Imagens segmentadas por subtração de fundo com diferentes limiares.

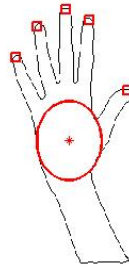
### 3.2 Pós-processamento

Após a segmentação de cada quadro, a imagem binária resultante pode conter “pixels falsos”, como por exemplo, quando o objeto da imagem binária contém buracos ou o fundo da imagem apresenta objetos ruidosos, como mostra a Figura 5a. É necessário corrigir ou amenizar estes efeitos indesejados. Para isso, é aplicado um filtro morfológico com operações de dilatação e erosão para melhorar a imagem segmentada.

Para refinar a segmentação, *blobs* de dimensões reduzidas contidos na mão são eliminados considerando-se como “objeto segmentado” apenas o de maior área. Este procedimento é um filtro mais seletivo que retira objetos que sejam maiores do que um determinado tamanho, mas que não fazem parte da mão. Após o tratamento com os filtros morfológicos a imagem não apresenta porções da mão separadas, ficando esta completamente conectada.

### 3.3 Extração das características para o rastreamento

Os parâmetros escolhidos para caracterizar a mão foram sua silhueta, as posições das pontas de cada dedo ao centro da mão, a distância do ponto de cada dedo ao centro da mão e o valor do raio da circunferência contida na palma da mão. A Figura 6 mostra uma mão com as características ressaltadas em vermelho.



**Figura 6:** Parâmetros para caracterização da mão.

O problema de encontrar os dedos de uma imagem da mão pode ser interpretado como o de encontrar pontos de picos e vales do contorno da mão, ou máximos e mínimos locais. A utilização de derivadas do contorno para encontrar máximos e mínimos locais não é uma boa solução na detecção de picos e vales, pois o contorno da mão segmentada é muito ruidoso, e a derivada gera uma grande quantidade de pontos [11]. Por isso utilizou-se a  $k$ -curvatura ou  $k$ -curva [11] [12]. O cálculo da curvatura em cada ponto do contorno é encontrado a partir do ângulo entre os dois vetores mostrados na Equação 2, na qual  $P(i)$  representa um pixel do contorno da mão,  $P(i-k)$  representa um pixel pertencente ao contorno anterior em  $k$  pontos ao ponto  $P(i)$  de referência e  $P(i+k)$  representa um ponto do contorno posterior em  $k$  pontos do ponto de referência.

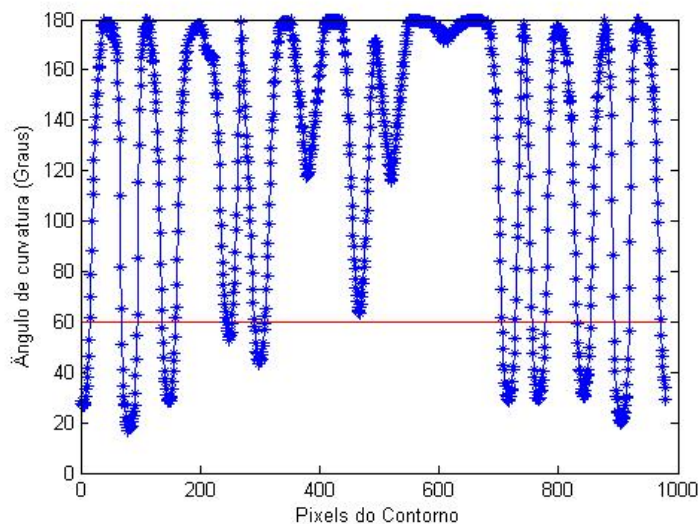
$$\begin{aligned}\vec{v}_1 &= [P(i-k), P(i)] \\ \vec{v}_2 &= [P(i+k), P(i)]\end{aligned}\quad (2)$$

Consegue-se então, ajustar pelo número de  $k$  pontos considerados a sua sensibilidade aos ruídos, já que se aumentar o intervalo de pontos, aqueles pertencentes às pequenas imperfeições na imagem não serão considerados, portanto sua curvatura não será encontrada.

O ângulo entre os dois vetores  $\vec{v}_1$  e  $\vec{v}_2$  pode ser encontrado através do produto escalar, segundo a Equação 3.

$$\phi = \arccos \left( \frac{\vec{v}_1 \cdot \vec{v}_2}{|\vec{v}_1| \cdot |\vec{v}_2|} \right). \quad (3)$$

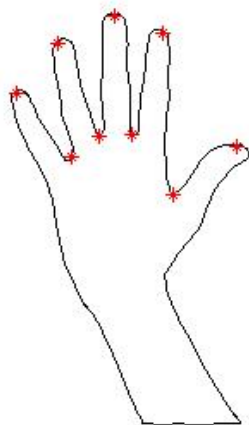
Se este ângulo for menor que um dado *threshold*  $T_\phi$ , o ponto é considerado como um ponto candidato à inflexão, isto porque todos os pontos pertencentes às pontas dos dedos ou às suas junções apresentam ângulo de curvatura próximos, como mostra a Figura 7. Nesta, pode-se ver em forma gráfica os ângulos de curvatura de todos os pontos pertencentes ao contorno (em azul) e uma reta (em vermelho) com o valor de *threshold*  $T_\phi$ . Nesta curva pode-se identificar nove porções abaixo do *threshold*, por isso nove conjuntos de pontos candidatos à inflexão.



**Figura 7:** Ângulo de curvatura de cada pixel do contorno da mão.

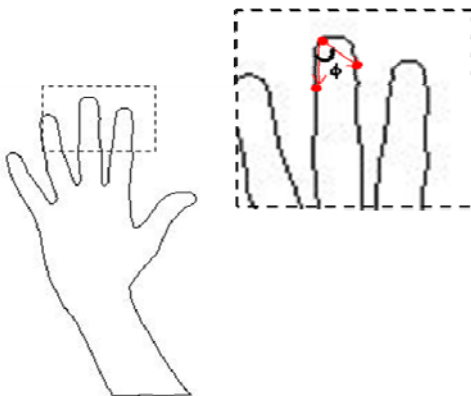
Para identificar apenas um ponto que caracterize cada inflexão do contorno, os pontos candidatos à inflexão que são adjacentes entre si são guardados juntamente com os seus respectivos ângulos de curvatura formando um mesmo conjunto, até que não haja mais candidatos adjacentes. O ponto de inflexão é definido como o ponto pertencente a um mesmo conjunto de pontos candidatos adjacentes e que possuam a menor curvatura. Observando a Figura 7, percebe-se que a última porção do gráfico foi unida ao grupo dos primeiros pontos, já que o algoritmo considera uma curva fechada.

Foram ajustados os valores de  $k$  e  $T_\phi$  em uma imagem contendo uma mão aberta para que se pudesse identificar todos os dedos chegando-se a um valor de  $k = 30$  e  $T_\phi = 60$  graus. A Figura 8 apresenta o contorno utilizado para encontrar estes valores de  $k$  e  $T_\phi$  com todos os pontos de inflexão.



**Figura 8:** Contorno da mão com picos e vales detectados.

Como os ângulos de curvaturas variam de zero a 180 graus, este algoritmo não diferencia os picos dos vales. Para que apenas os picos fossem identificados, incluiu-se a consideração da direção dos vetores  $\vec{v}_1$  e  $\vec{v}_2$ . A direção dos vetores em coordenadas cartesianas para valores de um pico terá sentido para aumento dos valores de  $x$  e no caso dos vales o vetor terá sentido para diminuição de  $x$ . Assim, pode-se considerar o valor do vetor em  $x$  e, se este for negativo localiza-se um vale, caso contrário um pico. Assim, apenas os pontos classificados como pico são considerados. A Figura 9 mostra a curvatura  $\phi$  da ponta de um dedo pela utilização de três pontos no contorno de uma mão.



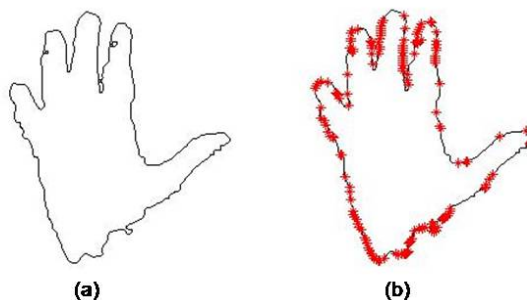
**Figura 9:** Ângulo de curvatura do contorno da mão utilizando k-curvatura.

Na Figura 10 tem-se a ação do algoritmo k-curvatura em uma imagem contendo a mão, onde apenas as pontas dos dedos são identificadas pelos retângulos em vermelho.



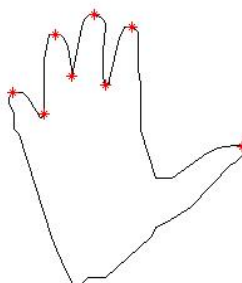
**Figura 10:** Contorno da mão com as cinco pontas dos dedos detectadas.

Para encontrar o contorno utilizou-se o Polígono de Mínimo Perímetro - MPP (*minimum-perimeter polygon*). Este algoritmo encontra um polígono aproximado de uma região ou um contorno de uma imagem, considerando apenas polígonos simples. Portanto, se a região tiver áreas internas, estas são desconsideradas. Logo, mesmo que existam buracos na imagem após passar pelos filtros morfológicos estes não serão considerados, segmentando-se apenas o contorno da mão. A qualidade do contorno possibilita maior confiabilidade na classificação de sua curvatura para encontrar os pontos de interesse. Como o contorno da mão é gerado utilizando os operadores de Sobel, a imagem de contorno resultante, em alguns casos, possui muitas imperfeições e é muito ruidosa, como mostra a Figura 11a.



**Figura 11:** Exemplo de contorno ruidoso com pontos detectados pela k-curvatura.

Quando aplicado o algoritmo k-curvatura em imagens ruidosas, o resultado é a identificação de muitos pontos de falsa inflexão, o que impossibilita o uso dos operadores de Sobel para encontrar o contorno, já que o algoritmo k-curvatura possui a característica de filtrar os ruídos menores. Entretanto, as imagens com imperfeições maiores são imunes a esta filtragem, e para que estes pontos sejam retirados, é necessário aumentar a seletividade do algoritmo de k-curvatura pelo aumento do parâmetro  $k$ , mas neste caso, corre-se o risco de o algoritmo ignorar também as pontas dos dedos. A Figura 11b exemplifica uma situação de contorno em que nos valores de limiar especificados de  $k = 30$  e  $T_\phi = 60$  graus, muitos pontos são considerados erroneamente como ponta dos dedos. A solução para este de problema foi encontrar o contorno através do conceito de MPP (*minimum-perimeter polygon*). A Figura 12 mostra a utilização da técnica no contorno da Figura 11a. Neste caso, os pontos que representam as inflexões do contorno, os asteriscos vermelhos assinalados na Figura 12, foram encontrados corretamente.



**Figura 12:** Contorno encontrado pela aproximação do polígono de mínimo perímetro (MPP).

Para que o rastreamento do movimento da mão seja relativamente independente da distância desta à webcam e também de diferentes tamanhos de mãos, localiza-se o centro da mão e o raio de uma circunferência interna com centro igual ao da mão. Para isso, utilizamos a Transformada da Distância Euclidiana (TDE), que converte uma imagem binária em outra imagem em que seu valor de intensidades contém a mínima distância de cada pixel ao limite do contorno da mão. O centro da palma pode ser estabelecido como sendo o pixel de maior intensidade na imagem resultante desta transformação, que representa o pixel

com maior distância em relação ao contorno. Esta distância é, então, definida como o raio de uma circunferência interna à mão. A Figura 13 exemplifica a imagem de intensidades gerada a partir da TDE.



**Figura 13:** Imagem de intensidades resultante da Transformada da Distância Euclidiana.

O algoritmo para definir o centro e o raio de uma imagem binária da mão segmentada é o seguinte:

1. Encontrar transformada da distância da imagem segmentada;
2. O raio é dado pelo máximo valor da transformada;
3. Encontrar ponto com máxima intensidade na imagem resultante da transformada;
4. O centro da mão terá posição deste ponto.

### 3.4 Algoritmo de Rastreamento

Após a detecção dos parâmetros que caracterizam a mão, o rastreamento consiste em encontrar a trajetória dos seus dedos para identificar o movimento de abaixá-lo e levantá-lo, portanto este movimento deve ser independente do movimento de translação da mão como um todo. Esta independência dos movimentos dos dedos em relação aos movimentos da mão considera cada posição em relação ao centro da mão. Ou seja, cada dedo é representado pelo ponto  $P_{dedo} = (x_D, y_D)$ , o centro da mão por  $P_{centro} = (x_C, y_C)$  sendo  $r$  o raio interno à mão. A posição do dedo normalizada é definida pela Equação 4, isto é, a posição de um dedo é definida pela distância euclidiana entre o ponto de pico que representa a sua ponta e o centro da palma da mão, normalizada pelo raio da circunferência interna a ela.

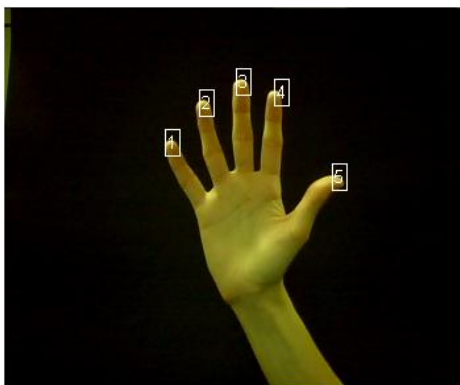
$$Posição_{dedo} = \frac{\sqrt{(x_D - x_C)^2 + (y_D - y_C)^2}}{r} \quad (4)$$

Para encontrar a correspondência do conjunto de pontos que localizam as pontas dos dedos de um quadro  $k$ , imagem  $I(k)$  com o conjunto do quadro posterior, imagem  $I(k+1)$ , definiu-se como custo a velocidade de cada ponto do quadro  $k$  em relação a todos os pontos do quadro  $k+1$ . Logo, sendo  $P_k = (x_k, y_k)$  um ponto do dedo de um quadro  $k$  a velocidade em relação a um ponto  $P_{k+1} = (x_{k+1}, y_{k+1})$  localizado no quadro  $k+1$  é dada pela Equação 5. A correspondência entre os pontos do quadro  $k$  e  $k+1$  é encontrada minimizando-se este custo.

$$Velocidade = \frac{\sqrt{(x_{k+1} - x_k)^2 + (y_{k+1} - y_k)^2}}{(k+1) - k} \quad (5)$$

Na inicialização do rastreamento dos dedos foi considerado um valor de referência com a mão totalmente estendida tal que todos os dedos fossem reconhecidos, para que posteriormente fosse possível comparar os movimentos. Portanto, o algoritmo considera este quadro, como o primeiro quadro a ser rastreado. Quando o primeiro quadro para o rastreamento é definido, um objeto é criado para guardar as informações que o caracterizam. Este objeto contém uma matriz em que cada linha representa um dedo, armazenando o nome do dedo, dado por um número de um a cinco, a posição deste na imagem e sua distância em relação ao centro da mão. As posições dos dedos são ordenadas e numeradas a partir do dedo anelar que se encontra o mais extremo à esquerda da imagem sendo nomeado como dedo "1". Isto se repete até que o dedo polegar é nomeado como dedo "5". A Figura 14 exemplifica um quadro que inicializa o rastreamento, reconhecendo os cinco dedos e com a mão completamente estendida.





**Figura 14:** Exemplo de quadro que inicializa o rastreamento dos dedos.

### 3.5 Tratamento de oclusões

A partir do primeiro quadro rastreado garante-se que sejam reconhecidos cinco pontos de picos no contorno pelo algoritmo de k-curvatura que representa as pontas dos cinco dedos da mão. Porém, do segundo quadro em diante pode acontecer que nenhum ponto ou um número menor que cinco pontos sejam reconhecidos, o que é explicado por duas hipóteses:

- Oclusão entre dois dedos, caso estes se aproximem o suficiente tal que não se consiga reconhecer o contorno entre eles. Este caso é exemplificado pela Figura 15 que mostra a imagem capturada onde dois dedos estão próximos o suficiente para não serem reconhecidos dois picos pelo algoritmo k-curvatura;
- Oclusão de um dedo ou todos os dedos com a palma da mão, que representa o movimento de abaixar os dedos o mais próximo possível da palma, ou fechar a mão.



**Figura 15:** Exemplo de oclusão entre os dedos com apenas dois pontos detectados pelo algoritmo de k-curvatura.

É essencialmente necessário que o algoritmo resolva a segunda hipótese, para caracterizar por completo o rastreamento ao abaixar e levantar os dedos. Se a posição dos dedos não é reconhecida ou considerada nos quadros em que não são encontrados picos no contorno da mão não é possível dar continuidade confiável ao movimento dos dedos. Esta solução é conseguida considerando-se que o rastreamento ocorre, primeiramente pela correspondência entre os pontos encontrados a partir da k-curvatura com os pontos do quadro anterior e estes são guardados na matriz com o nome que o equivale pela minimização do custo. Se o número dos pontos encontrados é menor que cinco, nas linhas da matriz onde o ponto não é reconhecido, as informações deste dedo são marcadas para que a parte do algoritmo de tratamento de oclusão reconheça esta situação.

O tratamento de oclusão reconhece os dedos marcados e encontra a distância entre a posição deste ponto no quadro anterior por todos os pontos pertencentes ao contorno da mão no quadro atual e considera a posição do ponto com a menor distância encontrada como sendo a posição mais provável deste dedo. Isto é, sendo a posição no quadro anterior,  $P_{k-1}$  de um dedo não reconhecido,  $P_i$  o  $i$ -ésimo ponto do contorno da mão e  $n$  a quantidade de pontos de contorno, sua posição  $P_k$  no quadro atual é dada pela Equação 6.

$$P_k = \min(|P_i - P_{k-1}|, i = 1..n) \quad (6)$$

Portanto, o algoritmo de rastreamento do movimento dos dedos poder ser resumido da seguinte maneira:

1. Encontrar no quadro atual pontos de picos pelo algoritmo de k-curvatura;

2. Achar equivalência destes pontos com os dedos encontrados no quadro anterior;
3. Se o número de pontos for menor que cinco pular para passo 5, senão próximo passo;
4. Encontrar dedos ainda não identificados pela menor distância de sua posição no quadro anterior ao contorno do quadro atual;
5. Guardar informações de cada dedo reconhecido com seu nome correspondente na matriz de objeto do quadro atual.

## 4. Resultados

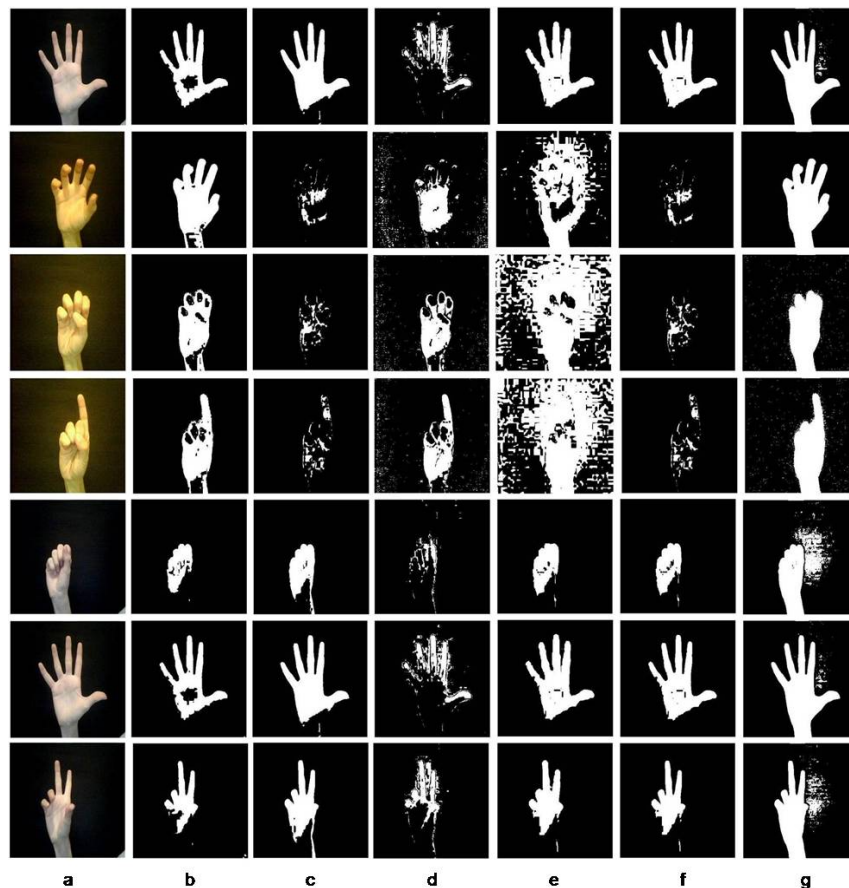
### 4.1 Avaliação dos métodos de Segmentação

Os resultados da análise quantitativa de cada um dos seis algoritmos considerados para avaliação de desempenho foi organizado na Tabela 1 e representam os valores médios encontrados de similaridade, taxa de detecção e taxa de falsos alarmes, bem como valor médio de tempo de processamento de cada algoritmo. Os métodos que apresentaram melhores desempenhos foram os baseados em *threshold* no espaço de cor RGB [13] e o nosso algoritmo baseado na subtração de *background*, já que apresentaram melhor taxa de detecção e maior similaridade com as imagens padrões e baixa taxa de falsos alarmes. Os métodos de segmentação em espaço de cor RGB normalizado [14] e HSV [15] apresentaram pequena taxa de falsos alarmes, porém observando seus baixos valores de taxa de detecção e de similaridade, percebe-se que este parâmetro não é suficiente para a segmentação. Estes valores baixos de similaridade e taxa de detecção podem prejudicar a correta segmentação em relação à mudanças de iluminação ou na detecção de peles de diferentes raças. O algoritmo que utiliza o espaço de cor YCbCr [16] apresentou um valor de taxa de detecção alta, por outro lado possui alta taxa de falsos alarmes o que explica o resultado baixo de similaridade.

**Tabela 1:** Tabela com os valores dos parâmetros de comparação e tempo de execução dos algoritmos de segmentação analisados.

MÉTODO	Similaridade Média	Taxa de Detecção Média	Taxa de Falsos Alarmes Média	Tempo Médio (s)
RGB[13]	0,6988	0,7691	0,1109	0,0190
RGB normalizado[14]	0,4680	0,5899	0,2053	0,0284
HSV[15]	0,4825	0,5343	0,1943	0,0895
YCbCr[16]	0,5476	0,7420	0,3219	0,0221
RGB normalizado e HSV[17]	0,4450	0,5123	0,1581	0,0507
Subtração de <i>Background</i> ( <i>proposto</i> )	0,7126	0,9717	0,2740	0,0254

Em relação aos tempos de execução, o algoritmo que opera no espaço HSV [15],[17] possui processamento mais lento possivelmente devido à necessidade da conversão da imagem do espaço RGB para HSV e, no caso do algoritmo que utiliza dois espaços de cores é esperado que seja mais lento já que realiza um passo a mais em seu processamento. A Figura 16 mostra um exemplo de segmentação com as seis metodologias avaliadas. Nas linhas, tem-se imagens com variação de iluminação.



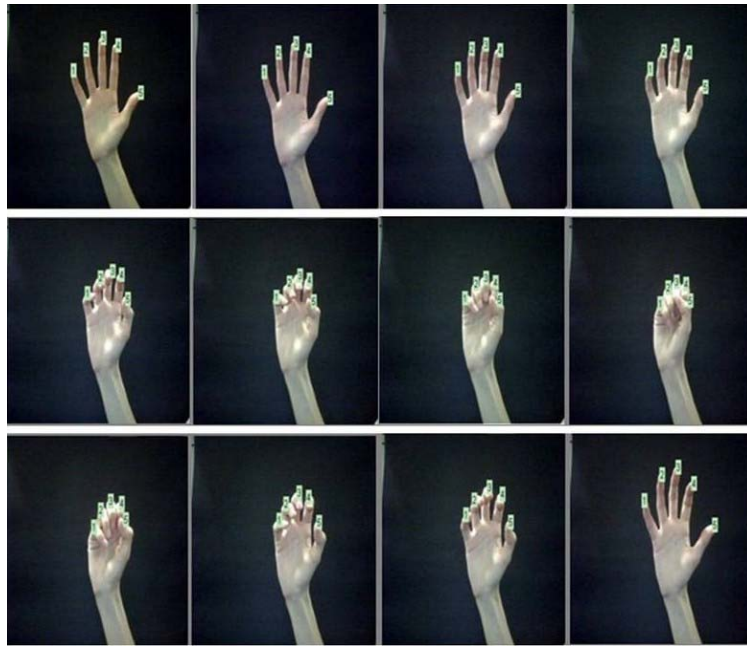
**Figura 16:** (a) Imagens originais com variação na iluminação. (b) Imagens correspondentes segmentadas pelo algoritmo baseado em limiar RGB[13]. (c) Imagens segmentadas pelo algoritmo baseado em espaço RGB normalizado[14]. (d) Imagens segmentadas pelo algoritmo no espaço de cor HSV[15]. (e) Segmentação no espaço de cor YCbCr[16]. (f) Segmentação nos espaços de cores RGB normalizado e HSV [17]. (g) Algoritmo proposto baseado em subtração de *background*.

Observa-se que o algoritmo proposto por [13] (Figura 16b) revelou-se mais robusto nos casos de variação de iluminação nas cenas ou para mãos de diferentes pessoas. Porém, de forma geral as imagens segmentadas pelo algoritmo de subtração de *background* apresentaram melhor definição na silhueta e sua resposta à variação de iluminação foi menos prejudicial para a detecção da mão.

## 4.2 Avaliação dos resultados do Rastreamento

Para a geração dos resultados, realizou-se uma análise visual utilizando-se um vídeo capturado com o movimento de fechar e abrir de uma mão. As imagens foram capturadas numa velocidade de 22,5 quadros por segundo e o rastreamento utilizou imagens a cada dois quadros. A Figura 17 apresenta alguns quadros capturados para exemplificar o movimento realizado no vídeo.

A cada quadro foram reconhecidos cada dedo com sua nomeação correta. Observou-se que se alguma parte da mão ficasse oclusa por muitos quadros, mais que 15 quadros, ocorriam erros no resultado do rastreamento devido a solução de oclusão. Um exemplo do tipo de erro observado é apresentado na Figura 18.



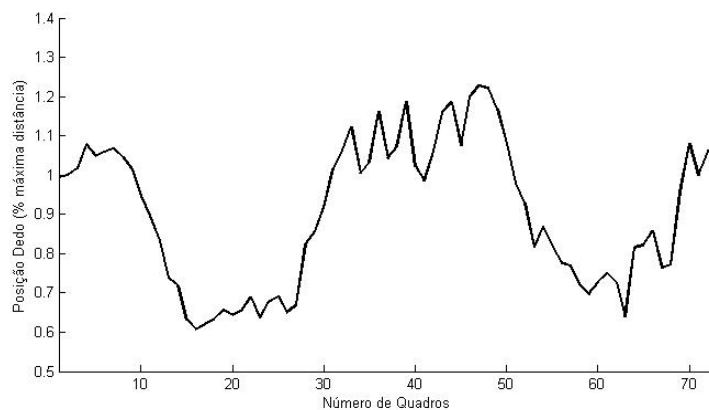
**Figura 17:** Exemplo de quadros de sequência de vídeo utilizada para analisar o algoritmo de rastreamento.



**Figura 18:** Exemplo de falha do tratamento de oclusão.

Como o tempo médio de execução do algoritmo em cada quadro de um vídeo com 1000 quadros é de aproximadamente 0,668 segundos, para a operação *online*, aproximadamente 1,5 quadros são analisados por segundo. Logo, problemas ocorrerão se o usuário ficar mais que 10 segundos com a mão fechada, por exemplo, o que demonstra que o algoritmo proposto responde consideravelmente bem no caso de oclusão.

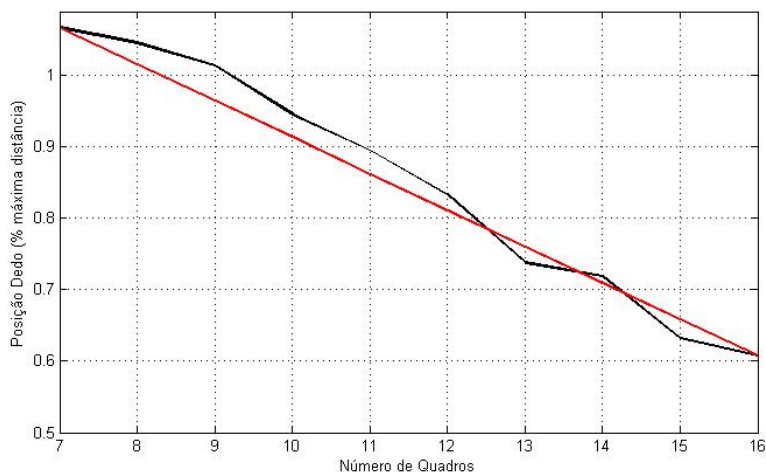
Para estimar a velocidade do movimento de um dedo a partir de sua posição em cada quadro, foi construída a curva de seu movimento apresentada na Figura 19. O vídeo utilizado para esta estimativa capturou imagens em uma taxa de 15 quadros por segundo e o algoritmo foi processado a cada dois quadros, portanto, o tempo real para realizar o movimento deverá ser a metade do calculado.



**Figura 19:** Curva do movimento de um dedo em uma sequência de vídeo.

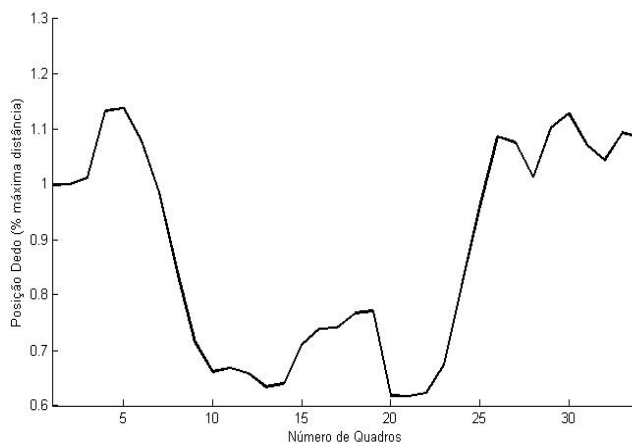
Observando-se esta curva do movimento percebe-se que o movimento de abaixar o dedo indicador ocorre aproximadamente entre os quadros 7 e 16. Aproximando-se esta curva neste intervalo a uma reta, como mostra a Figura 20, pode-se estimar o tempo em que o dedo faz o movimento, como mostra a Equação 7, na qual,  $Q_F$  representa o quadro onde o movimento cessou,  $Q_I$  o quadro onde o movimento iniciou e  $TQ$  a taxa de quadros capturados por segundo.

$$Tempo_{estimado} = \frac{(Q_F - Q_I) \cdot 2}{TQ} = 1,2s \quad (7)$$

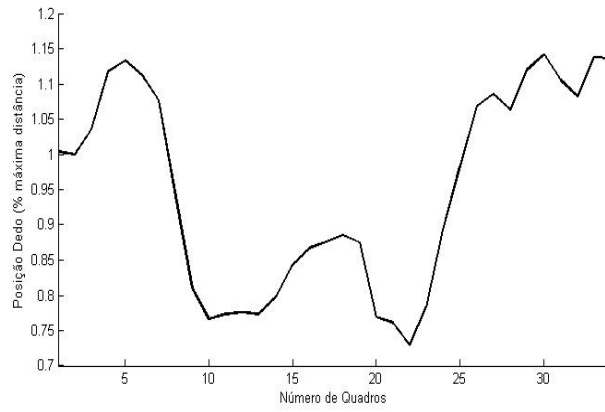


**Figura 20:** Aproximação por uma reta da curva de movimento de um dedo.

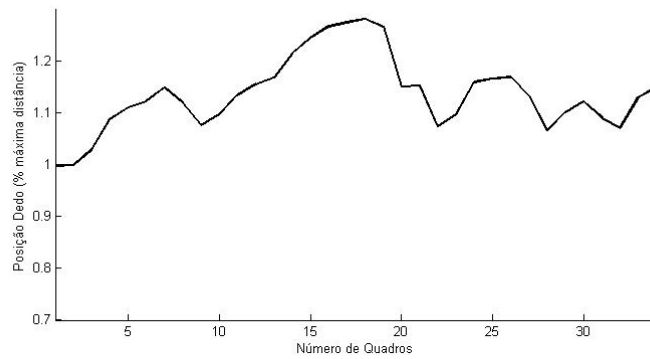
As Figuras 21 a 25 mostram o “tracking”, ou rastreamento de cada um dos dedos em um determinado movimento realizado dentro de um gesto em 40 quadros capturados. No eixo y destas figuras está a característica “distância ao centro da mão”. Como esta distância é normalizada pelo raio da mão, este movimento é sempre o mesmo, para o mesmo gesto independente da posição ou do tamanho da mão do usuário.



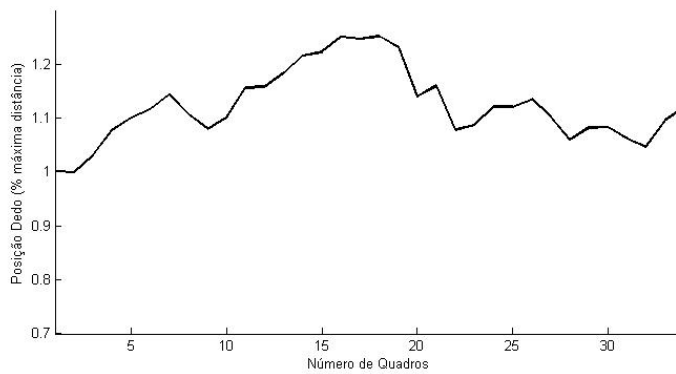
**Figura 21:** Curva do movimento do dedo mínimo em uma sequência de vídeo.



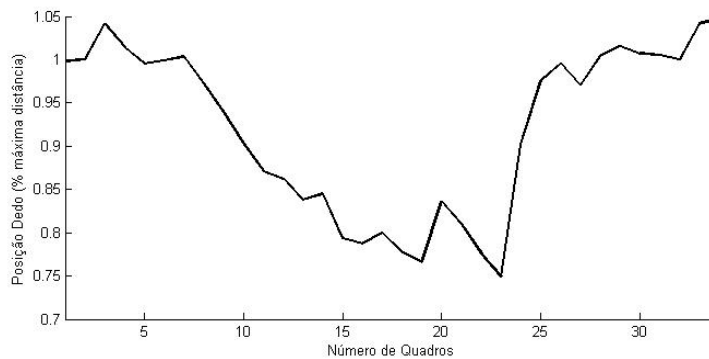
**Figura 22:** Curva do movimento do dedo anelar em uma sequência de vídeo.



**Figura 23:** Curva do movimento do dedo médio em uma sequência de vídeo.



**Figura 24:** Curva do movimento do dedo indicador em uma sequência.



**Figura 25:** Curva do movimento do dedo polegar em uma sequência de vídeo.

## 5. Conclusão

O método proposto foi considerado eficiente em suas etapas de segmentação e extração de características da mão, o que possibilitou realizar o rastreamento de todos os dedos em tempo compatível com o uso proposto, ou seja, para aplicações em IHC. É importante ressaltar que os testes realizados empregaram material de baixo custo, ou seja, um computador pessoal com “webcam” integrada, sem necessidade de iluminação especial. Nas cenas em que ocorrem oclusões a parte de tratamento de oclusão do algoritmo também obteve bons resultados nos casos em que a oclusão não persistiu por muitos quadros. Se a oclusão persistir por mais de 10 segundos, o algoritmo pode falhar no rastreamento dos dedos. Novos estudos deverão ser realizados para otimizar este tempo. Portanto, considerando-se o tempo de processamento do algoritmo, dentro de parâmetros compatíveis com a operação em tempo real, propõe-se sua utilização na Interface Humano-Computador por meio de comandos gestuais. Esta parte deverá ser motivo de pesquisa dos autores na continuidade do trabalho.

## Agradecimentos

Os autores agradecem à FAPESP pelo suporte financeiro e estrutura necessária para realização deste trabalho.

## Referências

- [1] Carroll, John M. **Encyclopedia entry on Human Computer Interaction (HCI)**. Disponível em: <[http://www.interaction-design.org/encyclopedia/human\\_computer\\_interaction\\_hci.html](http://www.interaction-design.org/encyclopedia/human_computer_interaction_hci.html)>. Acesso em: 25 Maio 2010.
- [2] Truyenque, M. A. Q. **Uma Aplicação de Visão Computacional que Utiliza Gestos da Mão para Interagir com o Computador**, Mestrado em Informática, PUC, Rio de Janeiro, 2005.
- [3] Ben-Israel, E. **Tracking of Humans Using Masked Histograms and Mean Shift**: Introductory Project in Computer Vision – Summary, [S.I.: s.n.], mar 2007. Disponível em: <<http://www.scribd.com/doc/4020604/Tracking-of-Humans-Using-Masked-Histograms-and-Mean-Shift>> . Acesso em: 15 de Fevereiro de 2010.
- [4] A. Yilmaz, M. Shah, O. Javed, “Object Tracking: A Survey”, *ACM Comput. Surv.*, Dec. 2006, pp.1-45.
- [5] Duca F.; Fredriksson, J.; Fjeld, M. Real-Time 3D Hand Interaction: Single Webcam Low-Cost Approach. In: **IEEE VR 2007 Workshop on Trends and issues in Tracking for Virtual Environments**, Gothenburg, Sweden, 2007.
- [6] Letessier, J.; Bérard, F. Visual Tracking of Bare Fingers for Interactive Surfaces. In: **ACM Symposium on User Interface Software and Technology (UIST)**, Santa Fe, New Mexico, USA, 2004.
- [7] Fujii, K.; Shimamura, J.; Arakawa K.; Arikawa, T. Three-dimensional finger tracking using direct and reflected infrared images. In: **Conference on Human Factors in Computing Systems**, Ft. Lauderdale, Florida, USA, p. 848 – 849, 2002.
- [8] H. L. Ribeiro, A. Gonzaga, Hand Image Segmentation in Video Sequence by GMM: a comparative analysis, In: **Proceedings of SIBGRAPI 2006**, xix Brazilian Symposium on Computer Graphics and Image Processing, 8-11 October 2006, Manaus, AM, Brazil, pp.357-364.
- [9] H. L. Ribeiro, “Reconhecimento de Gestos Usando Segmentação de Imagens Dinâmicas de Mãos Baseada no Modelo de Mistura de Gaussianas e Cor de Pele”, Escola de Engenharia de São Carlos, São Carlos, 2006, pp. 1-122.
- [10] R. C. Gonzalez, R. E., Woods, and S. L. Eddins, *Digital Image Processing Using MATLAB*, Prentice Hall, Upper Saddle River, NJ, 2004.
- [11] M. A. Q. Truyenque, “Uma Aplicação de Visão Computacional que Utiliza Gestos da Mão para Interagir com o Computador”, PUC-Rio, 2005.
- [12] E. C. Lopes, “Determinando a Posição e a Orientação da Mão Através de Imagens de Vídeo”, Porto Alegre, fevereiro de 2006.
- [13] Peer, P.; Kovac, J.; Solina, F. “Human skin colour clustering for face detection”. In: **EUROCON 2003 – International Conference on Computer as a Tool**, v.2, p.144-148, 2003.

- [14] Kühn, M. G.; Silva, M. S. “Nosso Hair: Sistema para simulação de coloração em tingimentos de cabelos.” **Faculdade Mater Dei Revista de Informática**, Pato Branco, v.1, n., 2005, p.37-44.
- [15] Sobottka, K.; Pitas, I. “Extraction of Facial Regions and Features using Color and Shape Information “. In: **Proceedings of the International Conference of Pattern Recognition (ICPR '96)**, 1996.
- [16] Chai, D.; Ngan, K.N. Face segmentation using skin-color map in video-phone applications, **IEEE Trans. Circuits Syst. Video Technol.**, v.9 n.4, 1999.
- [17] Wang, Y.; Yuan, B. “A novel approach for human face detection from color images under complex background.” **Pattern Recognition**, v. 34, p.1983-1992, 2001.