

ratoca—Mouse via Detecção de Marcadores por Câmera de Vídeo em Ambiente Não Controlado

A. G. Silva e S. C. Felipussi

Resumo—Este trabalho aborda uma alternativa de interface humano-computador utilizando uma câmera de baixo custo. Um protótipo de *mouse* via *webcam* é elaborado a partir da aquisição de imagens e processamento de marcadores simplificados definidos apropriadamente. Duas aproximações são propostas, uma baseada na análise dos componentes RGB para segmentação de marcadores circulares e coloridos; e outra baseada na detecção de listras horizontais ou verticais presentes em marcadores a partir da filtragem de Gabor. Pretende-se tornar o equipamento utilizável nos mais variados tipos de ambientes, com significativa tolerância à iluminação.

Palavras-Chave—Segmentação de marcadores, Detecção de objetos, Análise de formas, *Mouse* via câmera.

I. INTRODUÇÃO

A busca de uma interação mais natural com computadores vem se intensificando ao longo do tempo. Alguns métodos, por exemplo, baseiam-se na informação de eletromiogramas (EMG) [10] ou de expressões faciais [11]. Alguns equipamentos, como luvas e capacetes de realidade virtual ou cavernas virtuais, oferecem facilidades de interação, porém, em geral, com custo elevado para o usuário comum. Como alternativa, podem-se captar os movimentos das mãos por meio de fontes de luz infravermelha [1] ou do uso de uma câmera como sensor de entrada [2]. Reconhecimentos de gestos das mãos por visão computacional podem ser implementados utilizando-se, por exemplo, características multiescala e de probabilidade da cor da pele [12]. Marcadores grandes (40cm × 40cm), com desenho próprio, também são empregados e simplificam esta abordagem [4]. *Webcams* vêm se tornando populares para fins de entretenimento ou comunicação e podem também ser utilizadas neste sentido [3]. A idéia deste trabalho é acrescentar nova funcionalidade ao dispositivo *webcam* para tornar a interação humano-computador mais intuitiva. Ações semelhantes às de um *mouse* convencional por meio da detecção, por vídeo, de pequenos marcadores nos dedos do usuário, a partir de algoritmos simplificados e eficientes, são

propostas. A Fig. 1 mostra o diagrama de funcionamento do sistema batizado de *ratoca* (*rato* por *câmera*). A aquisição é feita com auxílio da implementação ARToolKit [5]-[9] e o posicionamento do cursor, com as bibliotecas do sistema de janelas utilizado. Este trabalho se concentra na forma como os marcadores são detectados e segmentados. O texto está organizado da seguinte forma: algumas definições de processamento de imagens são abordadas na seção II; os algoritmos propostos são descritos na seção III; resultados e comparações são apresentados na seção IV; conclusões são apresentadas na seção V.



Fig. 1. Sistema de posicionamento de *mouse* via câmera (*ratoca*).

II. DEFINIÇÕES PRELIMINARES

Um *video* corresponde a uma seqüência de quadros ou imagens. Uma imagem em nível de cinza consiste em uma grade retangular de *pixels*, ou seja, uma matriz de valores inteiros sem sinal, normalmente de 8 *bits*, representando intensidades luminosas (0 para preto, 255 para branco e valores intermediários para uma tonalidade de cinza). Define-se imagem binária quando se têm apenas duas intensidades, 0 ou 1. As imagens coloridas são formadas pela combinação de três destas matrizes (R-red, G-green e B-blue).

Seja $\mathcal{N}(x)$ a vizinhança de uma *pixel* x , define-se caminho de x_1 a x_n como uma seqüência $C=(x_1, x_1, \dots, x_n)$, em que $x_i \in \mathcal{N}(x_{i+1}), \forall i \in [1, n]$. Um componente conexo é um subconjunto CC , no qual há sempre um caminho $C, \forall x_a, x_n \in CC$.

A segmentação subdivide uma imagem em suas partes constituintes de forma a separar regiões de interesse (as quais recebem valor 1) do restante (que recebem valor 0), e pode ser realizada de forma manual, semi-automática ou automática.

A. G. Silva pertence ao Departamento de Ciência da Computação da Universidade do Estado de Santa Catarina, Brasil (e-mail: alexandre@joinville.udesc.br).

S. C. Felipussi pertence ao Departamento de Ciência da Computação da Universidade Federal de São Carlos, Brasil (e-mail: siovani@power.ufscar.br).

Uma técnica bastante simples e eficaz, em muitos casos, é a limiarização que, para uma certa intensidade t , todos os *pixels* com intensidades acima de t tornam-se 1 e todos os demais tornam-se 0.

A. Morfologia Matemática

Como ferramenta de segmentação, pode-se optar pela morfologia matemática, uma abordagem unificada baseada na teoria dos conjuntos¹ [6], aplicável em diversos tipos de problemas de processamento de imagens. Um elemento estruturante \mathcal{E} está relacionado à vizinhança considerada de um *pixel*. Supondo uma imagem binária I , a reconstrução $\mathcal{R}_{ec_{\mathcal{E}}}(I, S)$ consiste em, dentre várias regiões, selecionar apenas uma por meio de uma semente S (*pixel* ou conjunto de *pixels*) definida [6]. A dilatação $\mathcal{D}il_{\mathcal{E}}(I)$ consiste no cálculo do máximo dos vizinhos a cada translação, segundo um elemento estruturante \mathcal{E} refletido [6]. A erosão $\mathcal{E}ro_{\mathcal{E}}(I)$, por outro lado, obtém o mínimo dos vizinhos a cada translação de \mathcal{E} refletido [6]. A abertura $\mathcal{A}be_{\mathcal{E}}(I)$ é uma erosão seguida por uma dilatação usada para quebrar istmos estreitos e eliminar protusões finas [7]. O fechamento $\mathcal{F}ec_{\mathcal{E}}(I)$, de forma dual, preenche fendas em um contorno [7]. Em analogia, pode-se imaginar a abertura como uma enceradeira, por exemplo, circular no interior da região branca que não atinge alguns cantos, e o fechamento, como a mesma enceradeira no exterior da mesma região. A Fig. 2 exemplifica os quatro últimos operadores.

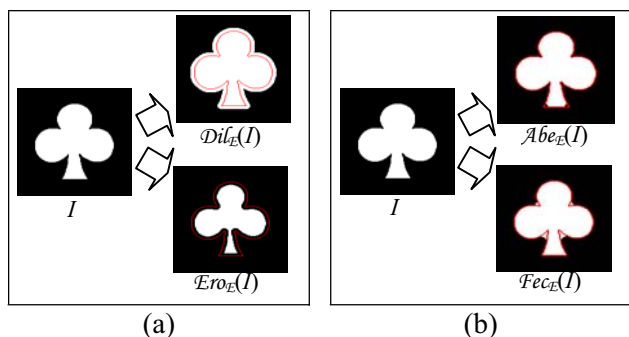


Fig. 2. Exemplos de operações morfológicas: (a) Dilatação e Erosão; (b) Abertura e Fechamento

B. Convolução

A convolução é uma técnica que pode ser usada em diversas operações em imagens como suavização, redução de ruído, detecção de bordas, realce de elementos, entre outras filtragens. É implementada pelo deslocamento de uma máscara m de pesos (contendo informação de vizinhança) por um *pixel* p de uma imagem I . O novo valor p' passa a ser a soma de todas as multiplicações de pesos pelo respectivo vizinho de p . Denota-se a convolução por $I * m$. Esta pode ser implementada no domínio da frequência usando a Transformada Discreta de Fourier $\mathcal{Z}(\cdot)$ pelo Teorema da Convolução [7]: $\mathcal{Z}(I * m) = \mathcal{Z}(I) \cdot \mathcal{Z}(m^e)$, onde m^e é a máscara aumentada com zeros ao seu redor de modo a ter as mesmas

dimensões da imagem I . No domínio da frequência, a filtragem é feita pela multiplicação pixel a pixel e pode ser mais eficiente para máscaras de tamanhos maiores, ainda mais se utilizada a Transformada Rápida de Fourier (FFT).

C. Filtragem de Gabor

Uma máscara de convolução com propriedades de frequência e orientação seletiva é desenvolvida com a seguinte forma geral [8]:

$$m_{\phi, f} = \exp \left[-\frac{1}{2} \left(\frac{x_{\phi}^2}{\delta_x^2} + \frac{y_{\phi}^2}{\delta_y^2} \right) \right] \cos(2\pi f x_{\phi}), \text{ sendo}$$

$$x_{\phi} = x \sin \phi + y \cos \phi \text{ e } y_{\phi} = x \cos \phi - y \sin \phi,$$

em que ϕ é a orientação, f é a frequência senoidal, e δ_x e δ_y são constantes de espaço do envelope gaussiano nos eixos. A Fig. 3 exemplifica essa máscara de tamanho 32×32 *pixels* e sua Transformada de Fourier, usando $\phi = 0$, $f = 1/16$, $\delta_x = 8$ e $\delta_y = 8$.

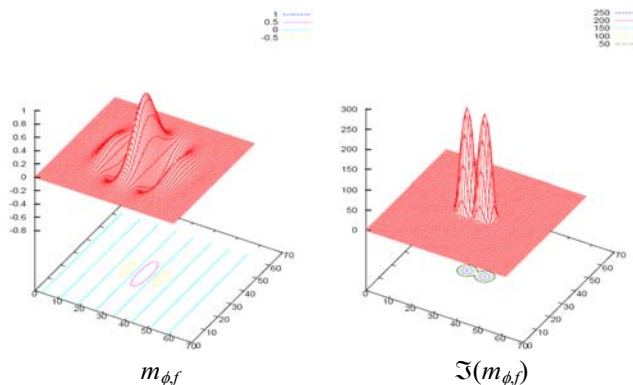


Fig. 3. Exemplo de máscara ou filtro de Gabor no domínio espacial e de frequência.

III. PROPOSTAS DE DETECÇÃO DE MARCADORES

Nesta seção é abordado o procedimento de aquisição de imagens, em que são propostos dois novos tipos de marcadores (o primeiro usando cores e o segundo usando texturas) e a forma como são detectados ou segmentados, além da exemplificação do mapeamento do centróide dos marcadores em posições de tela.

A. Aquisição de Imagens

Para aquisição de imagens é empregada uma *webcam* convencional com resolução de 320×240 *pixels*, com taxa de até 30 quadros por segundo. Para o acesso aos valores dos *pixels*, objetivou-se adotar um *software* livre, gratuito e multiplataforma. Embora diversos recursos do ARToolKit [9] sejam destinados à realidade aumentada, foi possível adequá-lo, de forma eficiente, para o propósito em questão, e dentre todos os passos necessários à escrita de uma aplicação nesta ferramenta, apenas a inicialização do caminho de vídeo e a leitura de quadros de entrada foram considerados neste trabalho.

¹ Também há a morfologia matemática para funções.

B. Segmentação Baseada em Cor

O primeiro modelo consiste na movimentação do cursor do *mouse* a partir da detecção de dois marcadores com duas colorações predefinidas, circulares, pintados ou colados nos dedos do usuário, conforme apresentado na Fig. 4.

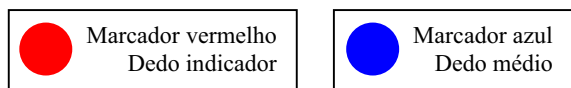


Fig. 4. Marcadores circulares pintados ou colados nos dedos com duas colorações.

As etapas do algoritmo para segmentação de marcadores coloridos (com boa iluminação) são descritas a seguir:

1. Normalização (entre $[0,255]$) de $|R-G|$.
2. Dilatação do complemento dos *pixels* entre intensidades (k_1, k_2) referentes à seleção da mão.
3. Remoção dos componentes conexos que tocam a borda da imagem, e seleção daqueles apenas com área entre 0.05% e 0.5% da área total, do passo 2.
4. Possíveis círculos: abertura morfológica de 3 por elemento estruturante em forma de disco de raio 4.
5. Sementes do vermelho: limiarização em k_2 para o valor normalizado de $|R-G|$.
6. Sementes do azul: limiarização em k_2 para o valor normalizado de $|B-G|$.
7. Reconstrução do passo 4, usando 5 como sementes.
8. Reconstrução do passo 4, usando 6 como sementes.
9. Dedo indicador: determinação do centróide de 7.
10. Dedo médio: determinação do centróide do passo 8.
11. Mapear os passos 9 e 10 em coordenadas de tela.

Pode-se constatar que esta proposta necessita da seleção inicial de dois limiares k_1 e k_2 de forma a adaptar a segmentação da mão às condições de aquisição das imagens. Neste trabalho, as condições de iluminação eram favoráveis e utilizou-se $k_1 = 50$ e $k_2 = 200$. A Fig. 5 ilustra cada um dos passos do algoritmo.

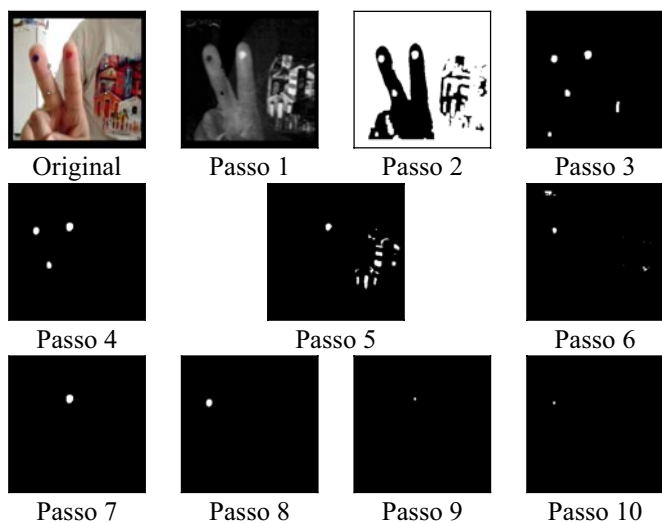


Fig. 5. Passos do algoritmo baseado em cor.

C. Segmentação Baseada em Textura

O segundo modelo proposto busca marcadores caracterizados por listras preto e branco, horizontais ou verticais, conforme mostrado na Fig. 6.

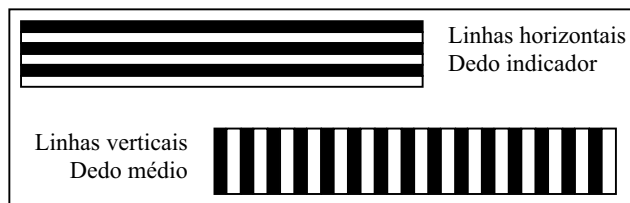


Fig. 6. Marcadores com listras em duas orientações que devem ser enroladas e afixadas em dois dedos.

Este método é possível por meio da filtragem de Gabor, que realça linhas em uma determinada orientação ϕ e frequência f . Esta última é inversamente proporcional à distância entre as listras ($f = 1/d$). Para melhorar o desempenho da convolução, o tamanho da máscara foi definida em 5×5 *pixels*. Dimensões superiores a este valor demonstraram ser mais rápidas no domínio da frequência. Quanto maior o tamanho da máscara, maior o realce e melhor a segmentação, porém, com custo computacional elevado para implementação em tempo real. A Fig. 7 ilustra a forma da máscara de Gabor 5×5 *pixels* no espaço e na frequência, tamanho considerado mínimo para segmentação correta e desempenho suficiente.

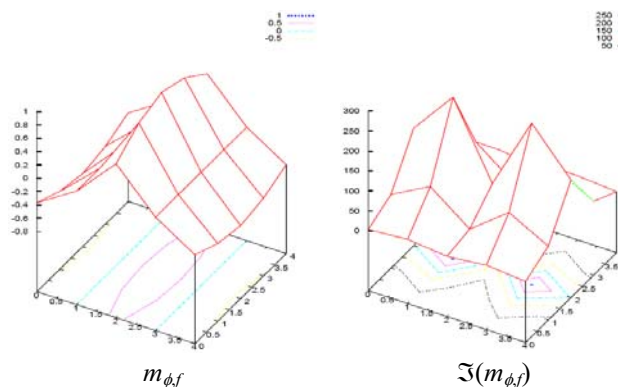


Fig. 7. Filtro de Gabor de tamanho mínimo (5×5 *pixels*) para o funcionamento adequado do algoritmo de segmentação baseado em textura.

As etapas do algoritmo para segmentação de marcadores com texturas (iluminação média) são listadas:

1. Transformação da imagem para níveis de cinza ($0,2989R + 0,587G + 0,114B$).
2. Filtragem de Gabor do passo 1 ($\phi = 0$ para horizontal ou $\phi = \pi/2$ para vertical, e $f = 1/4$).
3. Limiarização em 0 do passo 2.
4. Fechamento do passo 3 por um elemento estruturante em forma de quadrado de lado $2/f$;

5. Abertura do passo 4 por um elemento estruturante em forma de quadrado de lado $2/f^2$.
6. Atribuição de zeros a todos os *pixels* com distância $1/f$ da borda.
7. Seleção do componente conexo com a maior área;
8. Determinação do centróide, se a área do passo 7 for maior que 10% ou menor que 0,01% da área total da imagem (se não, não há marcador).

Os passos 2 ao 8 são processados duas vezes, uma para detecção do marcador horizontal ($\phi = 0$) e outra para o marcador vertical ($\phi = \pi/2$). A Fig. 8 mostra todos os passos para a localização do centróide do marcador com listras horizontais.

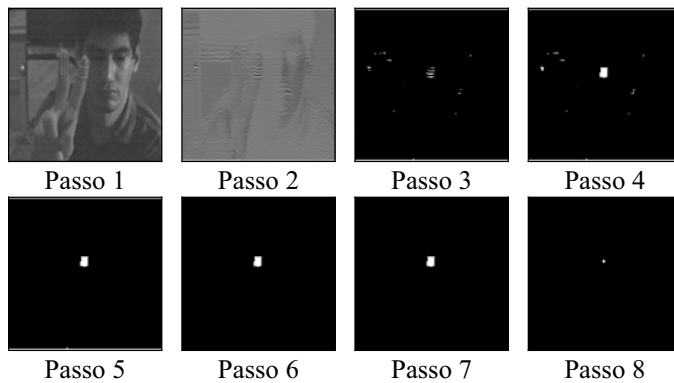


Fig. 8. Passos do algoritmo baseado em textura horizontal ($\phi = 0$).

D. Mapeamento em Posições de Tela

Considerando a determinação correta dos centróides dos marcadores, as coordenadas do *mouse* são mapeadas em tela da seguinte forma:

$$\begin{aligned} \text{MOUSE}_{\text{linha}} &= \text{CENTRÓIDE}_{\text{linha}} \cdot \text{TELA}_{\text{altura}} / \text{IMAGEM}_{\text{altura}} \\ \text{MOUSE}_{\text{coluna}} &= \text{CENTRÓIDE}_{\text{coluna}} \cdot \text{TELA}_{\text{largura}} / \text{IMAGEM}_{\text{largura}} \end{aligned}$$

A movimentação do cursor do *mouse* é dependente do sistema de janelas utilizado. Chamada semelhante a `SetCursorPos(MOUSElinha, MOUSEcoluna)` deve ser evocada visto que, para cada tipo de marcador, dois dedos podem ser detectados, podendo-se definir um dos centróides para retornar o movimento e o outro para desempenhar alguma funcionalidade como a de clique ou a de arrasto de elementos gráficos na tela.

IV. RESULTADOS

Nesta seção, serão apresentados alguns resultados de segmentação para ambos os modelos implementados (baseados em cor e textura). A Fig. 9 exemplifica alguns processamentos intermediários para oito quadros (Q^1 a Q^8) de vídeo no primeiro caso de detecção de marcadores coloridos. Para cada quadro, a primeira imagem refere-se à aquisição original. Os elementos em branco, da segunda e terceira imagens, são idênticos e consistem nos possíveis círculos da

mão (passo 4 do algoritmo). Em verde, na segunda imagem, estão possíveis sementes do marcador azul (passo 6), e, na terceira imagem, as possíveis sementes do vermelho (passo 5). Após a reconstrução (passos 7 e 8) usando tais sementes, pode-se determinar o círculo do marcador azul (contorno, em azul, na segunda imagem) ou o do marcador vermelho (contorno, em vermelho, na terceira imagem). Por fim, calcula-se o centróide dos dois círculos detectados (definidos pelos contornos anteriores em azul ou vermelho). Tais centróides são sobrepostos à aquisição original em níveis de cinza, com a respectiva cor, na quarta imagem. Observam-se falhas de detecção no quadro Q^7 (marcador vermelho não encontrado) e em Q^8 (posição incorreta do centróide do marcador azul).

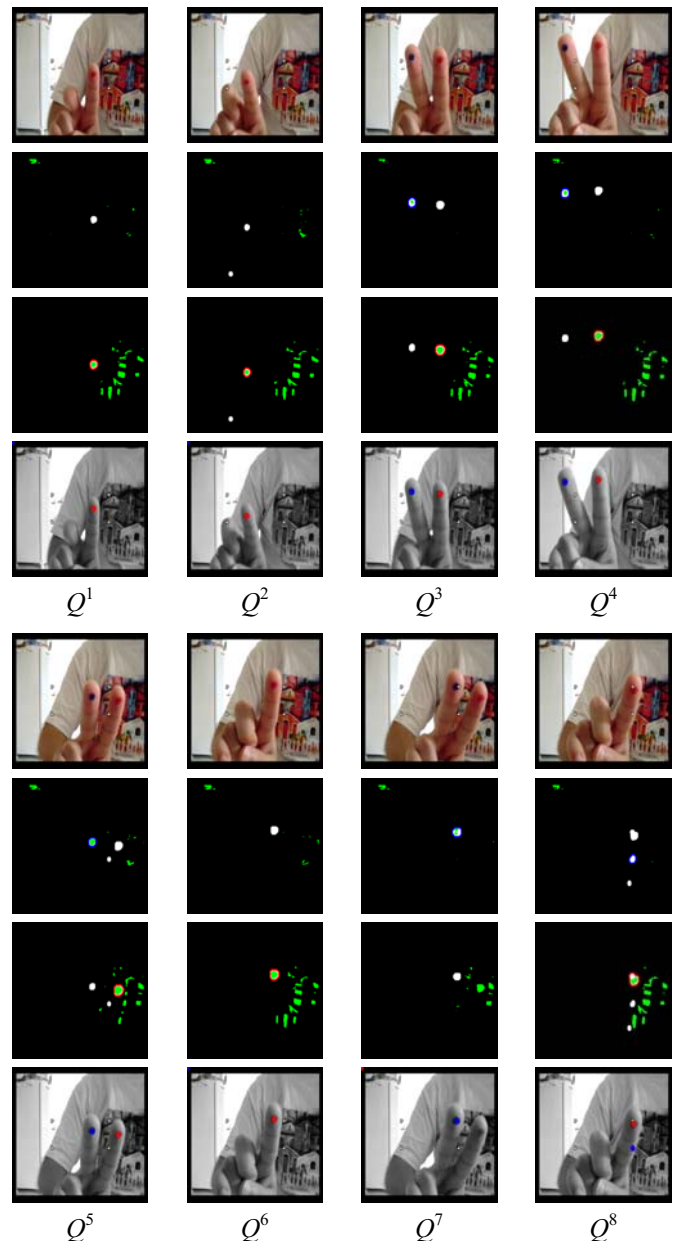


Fig. 9. Detecção pelo critério de cor.

A Fig. 10 exemplifica oito quadros (Q^1 a Q^8) de vídeo com a segmentação de marcadores proposta baseada em textura. Para cada quadro, a primeira imagem corresponde à filtragem de Gabor na horizontal (passo 2 para $\phi=0$). Os elementos em branco, na segunda imagem, referem-se à limiarização (passo 3 para $\phi=0$). O contorno do componente conexo selecionado (passo 7 para $\phi=0$) é sobreposto em vermelho na segunda imagem. As duas imagens subseqüentes correspondem às duas primeiras, mas agora com $\phi = \pi/2$. Os centróides dos elementos segmentados (contorno em vermelho) são sobrepostos à imagem original. Ocorre uma falha na detecção no quadro Q^5 (marcador horizontal não encontrado).

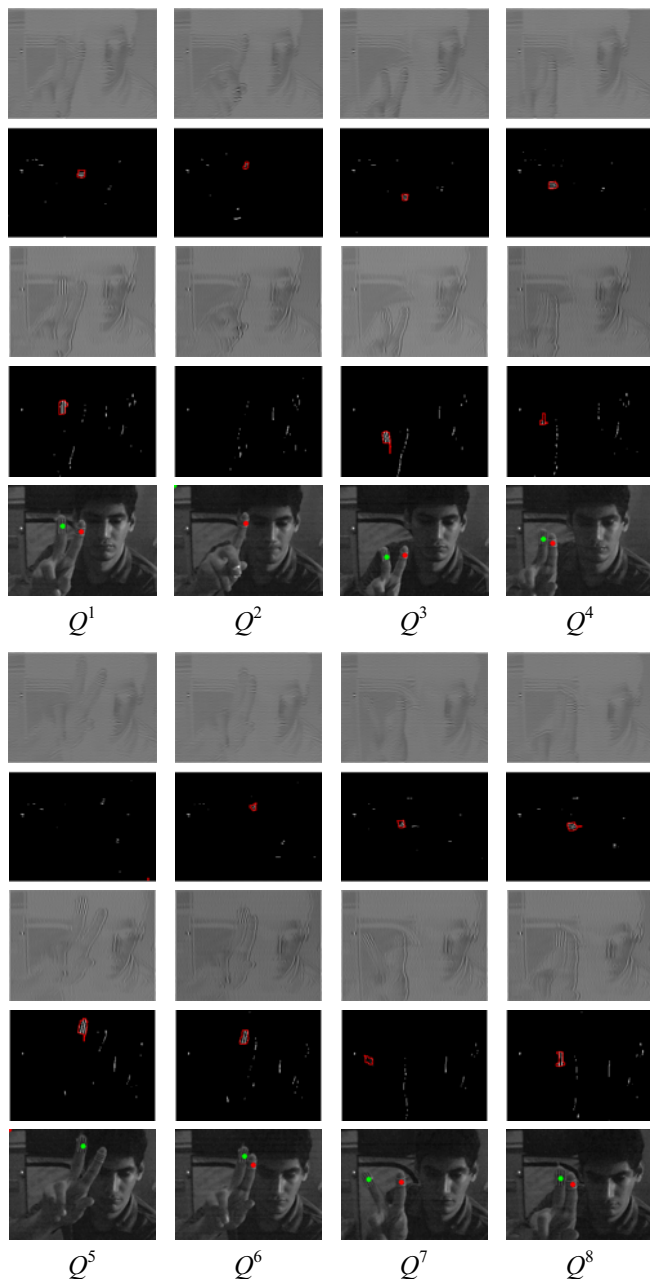


Fig. 10. Detecção pelo critério de textura.

Nos exemplos ilustrados, são apresentadas falhas de detecção de marcadores. A Tabela I exhibe a ocorrência dessas falhas em testes realizados. Falsa rejeição significa que um marcador existente não foi detectado; falsa aceitação ocorre quando um marcador inexistente é detectado; e a correta aceitação é o complemento da primeira, ou seja, quando um marcador existente é corretamente detectado. A aplicação utilizando cor revela-se aceitável somente com boa iluminação.

TABELA I
ÍNDICE DE ACERTO NOS EXPERIMENTOS REALIZADOS

Tipo	Iluminação	Falsa Rejeição	Falsa Aceitação	Correta Aceitação
<i>Cor</i>	Boa	13,79%	6,90%	86,21%
<i>Textura</i>	Qualquer	17,14%	0,00%	82,86%

V. CONCLUSÃO

As soluções propostas de detecção de marcadores e posicionamento do cursor do *mouse* apresentam imperfeições devido à complexidade das imagens, e também pelo fato de a iluminação e a textura do ambiente não serem controladas. Na primeira proposta, de segmentação de marcadores com cores, verificam-se dois problemas principais: influência de formas do ambiente (por exemplo: camisas com bolinhas coloridas) e iluminação insuficiente para distinguir as cores. Na segunda proposta, de segmentação de marcadores texturizados, outros dois problemas principais são considerados: influência de formas do ambiente (por exemplo: camisas ou paredes pintadas com quadradinhos listrados com mesma espessura das listras do marcador) e a inclinação da mão dentro de uma certa tolerância. Apesar de a segmentação baseada em cor apresentar uma elevada taxa de aceitação dos marcadores, é necessária uma iluminação adequada para a suficiente distinção de cores e, mesmo assim, ocorrem falsas aceitações, condição esta que pode gerar um comportamento não esperado do *mouse*. Na segmentação baseada em textura é possível constatar uma aceitação relativamente menor, mas o índice de falsa aceitação é praticamente zero (dificilmente tem-se um ambiente com texturas iguais e de mesmo tamanho em relação ao marcador), além de a tolerância à iluminação ser consideravelmente maior.

A detecção dos marcadores, mapeamento do cursor e seu deslocamento estão implementados. Porém, trabalhos futuros ainda são necessários como, por exemplo, tornar a segmentação mais robusta sem qualquer parâmetro inicial e, ao mesmo tempo, mais eficiente para que o *ratoca* possa ser aplicado em computadores de menor poder de processamento. A Tabela II exhibe um projeto de funcionalidade do *mouse* via *webcam* proposto, conforme a presença e combinação dos dois dedos (indicador e médio). Após as considerações destes aperfeiçoamentos, pretende-se estudar a viabilidade da utilização de lentes de contato texturizadas como marcadores. Desta feita, usuários tetraplégicos poderiam interfacear [13] com os computadores de forma rápida, confiável e com reduzido custo.

TABELA II
PROJETO DE FUNCIONALIDADE DO *MOUSE* POR *WEBCAM*

Ação dos dedos	Função
Apenas dedo indicador presente	Posicionamento
Breve ausência apenas do dedo indicador	Clique esquerdo
Apenas dedo médio presente	Posicionamento
Breve ausência apenas do dedo médio	Clique direito
Dedos indicador e médio presentes e unidos	Arrasto
Dedos indicador e médio presentes e separados	Rolagem

Projetos semelhantes são encontrados na literatura, porém, quase sempre requisitam ambientes controlados e têm um custo de aquisição elevado. A solução deste trabalho apresenta código-aberto e é disponibilizada livremente à comunidade para que haja contribuições no sentido de implementação de todas as funcionalidades projetadas para este *mouse*.

VI. AGRADECIMENTOS

Os autores agradecem ao aluno Eder Jonck de Souza pela implementação parcial, em tempo real, do protótipo de segmentação por textura.

VII. REFERÊNCIAS

- [1] S. Numazaki, A. Morishita, N. Umeki, M. Ishikawa, and M. Doi, "A Kinetic and 3D Image Input Device", in *ACM ISBN 1-581 13-028-7*, 1998, pp. 237-238.
- [2] K. Hinckley, M. Sinclair, E. Hanson, R. Szelisk, and M. Conway, "The VideoMouse: A Camera-Based Multi-Degree-of-Freedom Input Device", in *ACM 1-58113-075-9/99/11*, 1999, pp.103-112.
- [3] J. Segen and S. Kumar, "Simplifying human-computer interaction by using hand gestures", in *Communications of the ACM*, vol. 43, n.º 7, Jul. 2000, pp. 103-109.
- [4] E. Woods, P. Mason, and M. Billinghurst, "MagicMouse: an Inexpensive 6-Degree-of-Freedom Mouse", in *ACM ISBN:1-58113-578-5*, 2003, pp. 285-286.
- [5] P. Sinclair, K. Martinez, D. Millard, and M. Weal, "Augmented Reality as an Interface to Adaptive Hypermedia Systems", in *New Review of Hypermedia and Multimedia*, vol. 9, 2003, pp. 117-136.
- [6] E. Dougherty and R. Lotufo, "Hands on Morphological Image Processing", in *SPIE*, 2003.
- [7] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Addison-Wesley Publishing Company, 1992.
- [8] L. Hong, Y. Wan, and A. A. Jain, "Fingerprint Image Enhancement: Algorithm and Performance Evaluation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, n.º 8, Aug. 1998.
- [9] H. Kato, M. Billinghurst, and I. Poupyrev, "ARToolKit Version 2.33". *Human Interface Technology Laboratory*. University of Washington, Nov. 2000.
- [10] H. Jeong, J. Kim, and W. Son, "An EMG-based Mouse Controller for a Tetraplegic", *IEEE International Conference on Systems, Man and Cybernetics*, 2005.
- [11] P. Hong and T. Huang. "Natural mouse – a Novel Human Computer Interface", in *International Conference on Image Processing*, 1999. pp. 653-656.
- [12] L. Bretzner, I. Laptev, and T. Lindeberg, "Hand gesture recognition using multi-scale colour features, hierarchical models and particle filtering", in *Proc. 5th IEEE International Conference on Automatic Face and Gesture Recognition*, 2002.
- [13] H. Jeong, S. Kang, K. Jung, H. Kim, "Eye mouse: mouse implementation using eye tracking", in *International Conference on Consumer Electronics*, 2005, pp 207-208.

VIII. BIOGRAFIAS



Alexandre Gonçalves Silva nasceu em Apucarana, PR, Brasil. Obteve seu diploma de graduação em Engenharia de Computação na Universidade Estadual de Campinas (UNICAMP), Brasil, em 2001, e o título de mestre em Engenharia Elétrica também por esta universidade, em 2003. Desde 2004, é professor no Departamento de Ciência da Computação da Universidade do Estado de Santa Catarina (UDESC), em Joinville, SC, Brasil. Seus principais interesses estão na área de Processamento de Imagens e Visão Computacional. Atualmente, em seu doutorado, tem trabalhado com representações hierárquicas de imagens com o propósito de desenvolvimento de novos algoritmos de filtragem e segmentação.



Siovani Cintra Felipussi nasceu em São José do Rio Preto, SP, Brasil. Obteve seu diploma de graduação em Licenciatura em Matemática na Universidade Estadual Paulista (UNESP), Brasil, em 1995, o título de mestre em Matemática Aplicada também por esta universidade, em 1998, e, em 2003, obteve o título de doutor em Ciência da Computação pela Universidade Federal do Rio Grande do Sul (UFRGS). Trabalhou na Universidade do Estado de Santa Catarina (UDESC) e, desde 2005, é professor na Universidade Federal de São Carlos (UFSCar), em Sorocaba, SP, Brasil. Seus principais interesses estão na área de Processamento de Imagens e Visão Computacional.