Uma Interface Humano-Computador baseada em detecção e rastreamento de objetos

Cláudio dos Santos Fernandes
Curso de Ciência da Computação
Visão Computacional

24 de junho de 2010

Resumo

O objetivo deste trabalho foi desenvolver um método de rastreamento de objetos que possa ser aplicado ao problema de detecção e rastreamento de uma mão em uma Interface Humano-Computador. Um modo de usabilidade como este seria bastante útil em aplicações de edição gráfica, por exemplo, nas quais a utilização de um mouse nem sempre oferece ao usuário um nível adequado de liberdade de movimentação.

1 Pesquisas relacionadas

O artigo (PAVLOVIC; SHARMA; HUANG, 1997) discute as diversas técnicas empregadas por pesquisadores durante o processo de detecção, extração e rastreamento de padrões em imagens, bem como os maiores desafios da área e cita técnicas que visam contorná-los. Nele, é desenvolvido o conceito de sistema global de interpretação gestual baseado em visão computacional. De forma geral, estes sistemas requerem que um modelo matemático de gestos seja inicialmente estabelecido, como esquematizado na Figura 1.

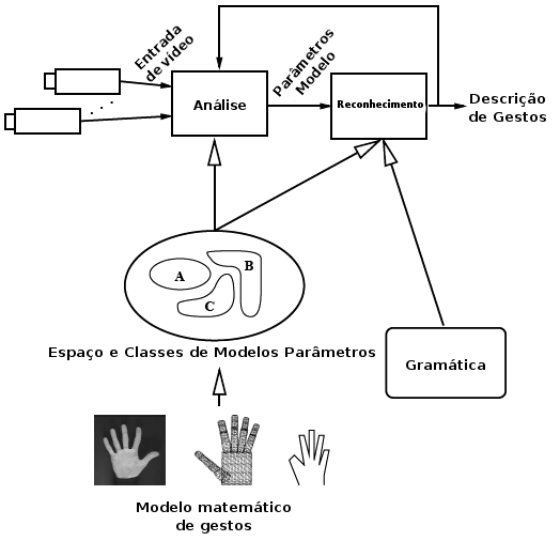

Figura 1: Diagrama do sistema de interpretação gestual definido em Pavlovic, Sharma e Huang (1997). Obtida e traduzido deste mesmo artigo.

Este artigo foca na questão do uso de técnicas de visão computacional aplicadas a problemas da área de Interfaces Humano-Computador. Por isso, seu desenvolvimento parte da definição de conceitos basicos, como gestos, a partir do ponto de vista desta área. Por exemplo, a seguinte proposição é feita acerca do conceito de gesto para ambientes IHC:

Em um ambiente IHC o seguinte conceito de regras determina a segmentação temporal de gestos:

1. O intervalo gestual consiste de três fases: preparação, movimento e retração.
2. A posição da mão durante o movimento segue um caminho classificável no espaço de parâmetros.
3. Gestos são confinados em um volume espacial especificado (chamado workspace).
4. Movimentos repetitivos da mão são gestos.
5. Gestos manipulativos tem tamanhos de intervalo gestual maiores do que gestos comunicativos.

Uma boa parte dos trabalhos realizados na área se baseiam na decomposição do corpo do usuário em objetos articulados. Esta abordagem tem dois grandes obstáculos: a grande quantidade de articulações a serem analisadas, e as dificuldades de obtê-las por meio de técnicas de visão computacional. Em técnicas de mapeamento tridimensional, elas são contornadas com o uso de premissas relacionadas às dependências entre juntas e seus limites em termos de ângulo máximo de rotação da articulação. Nas abordagens baseadas em modelos bidimensionais, a solução adotada é modelar gestos por meio da relação entre a aparência de qualquer gesto à aparência de um grupo de gestos pré-definidos em templates.

O processo de localização e extração das mãos em uma imagem é uma tarefa necessária e bastante complexa. É comum que algumas restrições sejam aplicadas para a simplificação deste processo. Tais restrições geralmente recaem sobre a escolha do background, do usuário e do sistema de captura de imagens. A primeira geralmente significa o uso de um background uniforme, enquanto a segunda se dá por meio do uso de luvas de cor e textura uniforme ou com marcadores em pontos específicos da mão. A abordagem de Kurata et al. () é inovadora em alguns aspectos. Foram utilizadas informações relacionadas às cores para detectar e rastrear a mão do usuário - ao invés de cores pré-definidas, os modelos de cor foram construídos dinamicamente. As premissas para detecção utilizadas foram: O usuário deve iniciar o processo de detecção apontando seu dedo para um "círculo guia", o que parametrizaria o processo de identificação de sua mão. A partir daí, o rastreamento é feito por meio do algoritmo *Mean Shift*.

Uma abordagem similar foi adotada por Bottino e Laurentini (2007). Nesta pesquisa, foram utilizados os algoritmos de *Mean Shift* para detecção da mão, e *CAMShift* para seu rastreamento. Foi utilizado um marcador referencial na cena analisada. A silhueta da mão foi inicialmente extraída da imagem. Um modelo 3D foi então adaptado à imagem segmentada com o objetivo de reconstruir a postura da mão e a posição do marcador. O reconhecimento da postura da mão foi baseado no casamento da projeção de modelos 3D (composição de elipsóides) em planos bidimensionais e a silhueta da mão, o qual foi feito por meio do algoritmo *ICondensation*.

Como pode ser notado, muitos trabalhos desenvolvidos nesta área dependem, em geral, de premissas que podem afetar negativamente a usabilidade do sistema. A necessidade de calibração inicial para determinação dos perfis de cores da mão do usuário, por exemplo, limita o uso do sistema a pessoas com perfis de pele similares, tornando proibitivo o uso de luvas ou objetos que cubram a sua pele.

Outras abordagens se baseiam no uso, por parte do usuário, de um material identificável pelo algoritmo de rastreamento - em geral uma luva com marcadores especiais, o que representa uma restrição de usabilidade do sistema.

2 Metodologia

2.1 Objetivos Detalhados

Como já foi explicado, este trabalho tem como objetivo desenvolver um modelo de rastreamento de objetos independente de marcadores instalados no corpo do usuário.

O processo de rastreamento desenvolvido passa pelas etapas de caracterização do *background*, extração das diferenças das imagens em relação ao *background*, readaptação do *background* e cálculo do fluxo óptico entre frames consecutivos. Cada uma destas etapas é discutida a seguir.

2.2 Caracterização do background

O *background*, ou imagem de referência, é uma imagem que armazena o fundo da cena sendo capturada pela câmera instalada. Nesta imagem, variações nas intensidades dos pixels são tipicamente decorrentes de dois fatores:

1. Ruído introduzido pela câmera.
2. Movimentação de objetos na cena.

Levando em conta estes fatores, uma caracterização inicial de background deve ser feita sem a presença de objetos que se espera verificar grande movimentação durante a utilização do sistema. Contudo, tal abordagem de caracterização poderia introduzir limitações indesejáveis por parte dos usuários do sistema. Sendo assim, a imagem de background pode ser determinada por meio do primeiro frame emitido pela câmera após a inicialização do sistema.

É esperado que esta imagem seja modificada com o tempo (por exemplo, se ocorrer a inserção ou remoção de um objeto na cena, ou até mesmo movimentações da câmera). Este é um problema a ser lidado na etapa de readaptação do *background*.

2.3 Extração da Matriz de Diferenças

A matriz de diferenças D_t armazena as diferenças entre o último *frame* I_t obtido e a imagem de referência M. Para lidar com ruídos, a imagem M deve passar por uma filtragem de ruídos - neste trabalho, foi utilizado o filtro gaussiano com janela de varredura de tamanho 7. Pequenas variações entre as imagens, em geral causadas por sombreamento no *frame* atual, podem ser eliminadas por meio de um *threshold* ϑ. Formalmente, temos a seguinte relação:

$$D_t(i,j) = \begin{cases} |M(i,j) - I_t(i,j)| & \text{Se } |M(i,j) - I_t(i,j)| > \vartheta \\ 0 & \text{Caso contrário} \end{cases}$$

A definição de ϑ deve ser feita de forma cautelosa. Seu valor deve pertencer ao domínio de 0 a 255. Em geral, sua escolha deve ser maior para ambientes onde se espera que a variação de *background* seja mais frequente.

2.4 Readaptação da Imagem de Referência

Este processo faz com que a imagem de referência M seja adaptada às variações no fundo do ambiente. Durante o cálculo da matriz de diferenças D_t, M é aproximada de I_t por uma taxa a ser definida pelo *threshold* ξ, conforme a equação (2.4).

$$M_t(i,j) = \begin{cases} M_{t-1}(i,j) + \frac{|M_{t-1}(i,j) - I_t(i,j)|}{\xi} & \text{Se } |M(i,j) - I_t(i,j)| > \vartheta \\ I_t(i,j) & \text{Caso contrário} \end{cases}$$

O valor de ξ é inversamente proporcional à variação esperada da imagem de referência, e deve ter um valor entre 1 e 255.

2.5 Cálculo do fluxo óptico

Fluxo óptico é uma técnica que permite estimar o deslocamento de pontos dadas duas imagens distintas de uma mesma cena onde possam haver objetos em movimento. Os detalhes deste algoritmo são apresentados em (LUCAS; KANADE, 1981).

Nesta abordagem, o cálculo de fluxo óptico é realizado entre matrizes de diferenças consecutivas D_{t-1} e D_t. A partir deste passo, são obtidos os vetores deslocamento para diversos pontos de interesse da imagem inicial D_{t-1}. Estes vetores são somados para o cálculo do vetor direção $\vec{V_d}$, que dá a direção do deslocamento médio para a imagem. Este vetor pode ser multiplicado por um escalar Φ fornecido por um eventual algoritmo de decomposição da mão do usuário. A idéia é que este valor deve ser inversamente proporcional à distância do objeto rastreado à câmera, e, portanto, é necessário ter uma noção desta distância a fim de calcular o módulo do deslocamento.

2.6 Execução das instruções requisitadas

Esta é a etapa final, onde ocorre a execução das instruções enviadas pelo usuário pelo sistema operacional. A arquitetura Linux possui a biblioteca Xlib (TRONCHE, 2005), que é uma interface de baixo nível de comunicação com o *X Window System* com muitas funcionalidades que permitem o controle de certos recursos do sistema, o que evita que este trabalho seja dependente do desenvolvimento de um *device driver* específico.

2.7 Testes e resultados

Foram realizados alguns testes qualitativos para verificar o comportamento do sistema em diversas ocasiões. Na primeira, foram provocados movimentos da câmera a fim de verificar o impacto de variações intensas na cena sobre a movimentação do mouse. Em geral, os efeitos foram sempre perceptíveis, contudo, toleráveis.

O segundo teste visou verificar o cálculo do vetor deslocamento em função de movimentações muito bruscas, conforme mostram as imagens 2. Neste caso, observaram-se ruídos durante o deslocamento do ponteiro do mouse. Contudo, estes dados não passaram por nenhuma filtragem, o que seria um aspecto a ser explorado em possíveis trabalhos futuros.

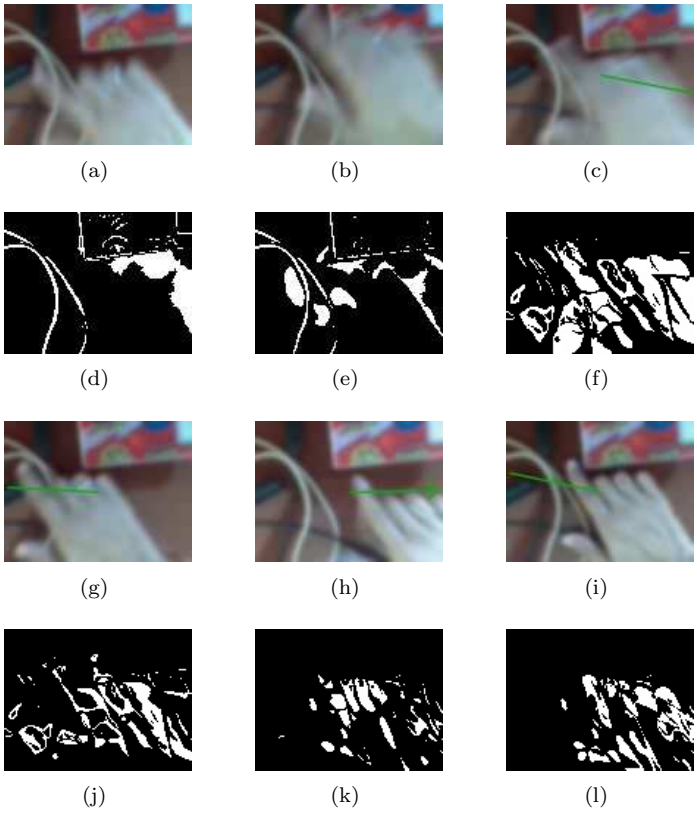

Figura 2: Imagens intermediárias produzidas para o algoritmo desenvolvido. As figuras em escala de cinza são correspondentes às matrizes de diferença, enquanto as demais exibem um vetor deslocamento associado ao cálculo do fluxo óptico.

2.8 Conclusões

A modelagem da extração da matriz de diferenças D_t por meio de um *threshold* não foi uma escolha que atendeu completamente a um dos aspectos que o trabalho se propõe a tratar: A aceitação de diversos tipos de fundo com um mínimo de calibração por parte do usuário. Isso ocorre pois o valor ideal para ϑ depende das condições do ambiente onde a câmera será instalada. Ruído no processo de aquisição das imagens é outro fator que influencia a determinação deste valor. Uma melhor forma de tratar o problema da eliminação de *outliers* poderia se basear na modelagem de como sombras interferem na intensidade de pixels, uma vez que a maior parte dos *outliers* são causados pelas sombras do usuário.

A aplicação do cálculo de fluxo óptico se mostrou adequada para aplicações em que o usuário está suposto a mover sua mão lentamente. Contudo, em aplicações onde as movimentações são frequentemente bruscas, uma abordagem diferente, como o algoritmo *CAMShift* (BRADSKI, 1998), seria indicada.

Referências

BOTTINO, A.; LAURENTINI, A. How to make a simple and robust 3d hand tracking device using a single camera. In: *ICCOMP'07: Proceedings of the 11th WSEAS International Conference on Computers*. Stevens Point, Wisconsin, USA: World Scientific and Engineering Academy and Society (WSEAS), 2007. p. 414–419. ISBN 978-960-8457-95-9.

BRADSKI, G. R. *Computer Vision Face Tracking For Use in a Perceptual User Interface*. 1998.

KURATA, T. et al. *The Hand Mouse: GMM Hand-color Classification and Mean Shift Tracking*.

LUCAS, B. D.; KANADE, T. An iterative image registration technique with an application to stereo vision. *Proceedings of Imaging understanding workshop*, p. 121–130, 1981.

PAVLOVIC, V. I.; SHARMA, R.; HUANG, T. S. Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Trans. Pattern Anal. Mach. Intell.*, IEEE Computer Society, Washington, DC, USA, v. 19, n. 7, p. 677–695, 1997. ISSN 0162-8828.

TRONCHE, C. *The Xlib Manual*. October 2005. `http://tronche.com/gui/x/xlib/`.