

**Michel Alain Quintana Truyenque**

**Uma Aplicação de Visão Computacional  
que Utiliza Gestos da Mão para Interagir  
com o Computador**

**DISSERTAÇÃO DE MESTRADO**

**DEPARTAMENTO DE INFORMÁTICA**

**Programa de Pós-Graduação em Informática**

Rio de Janeiro , março de 2005



**Michel Alain Quintana Truyenque**

**Uma Aplicação de Visão Computacional que Utiliza  
Gestos da Mão para Interagir com o Computador**

**Dissertação de Mestrado**

Dissertação apresentada como requisito parcial para obtenção do título de Mestre pelo Programa de Pós-Graduação em Informática da PUC-Rio.

Orientador: Prof. Marcelo Gattass



**Michel Alain Quintana Truyenque**

## **Uma Aplicação de Visão Computacional que Utiliza Gestos da Mão para Interagir com o Computador**

Dissertação apresentada como requisito parcial para obtenção do título de Mestre pelo Programa de Pós-Graduação em Informática da PUC-Rio. Aprovada pela Comissão Examinadora abaixo assinada.

**Prof. Marcelo Gattass**

Orientador

Departamento de Informática – PUC-Rio

**Prof. Paulo Cezar Pinto Carvalho**

Instituto de Matemática Pura e Aplicada – IMPA

**Prof. Raul Queiroz Feitosa**

Departamento de Engenharia Elétrica – PUC-Rio

**Prof. Anselmo Antunes Montenegro**

Instituto de Matemática Pura e Aplicada – IMPA

**Prof. Waldemar Celes Filho**

Departamento de Informática – PUC-Rio

**Prof. José Eugênio Leal**

Coordenador(a) Setorial do Centro Técnico Científico – PUC-Rio

Rio de Janeiro, 28 de março de 2005

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, do autor e do orientador.

### **Michel Alain Quintana Truyenque**

Graduo-se bacharel em Engenharia de Sistemas na UNSA (Universidad Nacional de San Agustín) em 2002. Ingressou no programa de mestrado do Departamento de Informática em 2003 e desde então participa em projetos de pesquisa no Tecgraf/PUC-Rio (Laboratório de Computação Gráfica) nas áreas de Visão Computacional e Visualização Científica. Atualmente participa do desenvolvimento de um sistema de visualização 3D de dados sísmicos realizado em parceria com a gerência de Tecnologia em Geofísica da Petrobrás

#### Ficha Catalográfica

Truyenque, Michel Alain Quintana

Uma aplicação de visão computacional que utiliza gestos da mão para interagir como o computador / Michel Alain Quintana Truyenque; orientador: Marcelo Gattass – Rio de Janeiro: PUC-Rio, Departamento de Informática, 2005.

100 f. : il. ; 30 cm

Dissertação (mestrado) – Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática.

Incluí bibliografia

1. Informática – Teses. 2. Visão computacional. 3. Interação Humano-Computador. 4. Gestos da mão. 5. Reconhecimento de Gestos 6. Interação baseada em visão. I. Gattass, Marcelo. II. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Informática. III. Título.

CDD:004

Com amor, a minha Família.

## **Agradecimentos**

Em primeiro lugar a Deus pelo caminho que me foi reservado até aqui.

A minha família, com todo amor, aos meus pais Yanina e Santos pelos ensinamentos, apoio constante e amor de sempre. As minhas irmãs Shirley e Karla pelo carinho, apoio constante e alegria de sempre.

A meu orientador Professor Marcelo Gattass pela confiança em mim depositada, pelos ensinamentos e a oportunidade de trabalhar a seu lado. Um agradecimento especial pelo apoio e compreensão nos momentos difíceis durante o curso do mestrado.

Ao Anselmo pelos conselhos e ajuda constante durante o desenvolvimento deste trabalho.

Aos meus professores do departamento, pelos ensinamentos e o aprendizado aqui desenvolvido.

A Maria Clara pela luz no final do túnel.

A Karla, amiga constante, mesmo de longe pelo carinho e apoio.

Aos meus colegas da pós-graduação, tanto do Mestrado quanto do Doutorado, pelo companheirismo, os momentos de diversão e por ter me recebido sempre de braços abertos. Aos funcionários do departamento, pelo carinho e atenção.

Aos colegas do Tecgraf pelo ótimo ambiente de trabalho, aos membros do grupo de visualização de dados sísmicos pela experiência aprendida e pela oportunidade de trabalharmos juntos.

A PUC-Rio, Tecgraf e a Capes pelo apoio financeiro sem o qual este trabalho não poderia ter sido desenvolvido.

## Resumo

Quintana Truyenque, Michel Alain. **Uma Aplicação de Visão Computacional que Utiliza Gestos da Mão para interagir com o Computador.** Rio de Janeiro, 2005. 100p. Dissertação de Mestrado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

A Visão Computacional pode ser utilizada para capturar gestos e criar dispositivos de interação com computadores mais intuitivos e rápidos. Os dispositivos comerciais atuais de interação baseados em gestos utilizam equipamentos caros (dispositivos de seguimento, luvas, câmeras especiais, etc.) e ambientes especiais que dificultam a difusão para o público em geral. Este trabalho apresenta um estudo sobre a viabilidade de utilizarmos câmeras Web como dispositivo de interação baseado em gestos da Mão. Em nosso estudo consideramos que a mão humana está limpa, isto é, sem nenhum dispositivo (mecânico, magnético ou óptico) colocado nela. Consideramos ainda que o ambiente onde ocorre a interação tem as características de um ambiente de trabalho normal, ou seja, sem luzes ou panos de fundo especiais. Para avaliar a viabilidade deste mecanismo de interação, desenvolvemos alguns protótipos. Neles os gestos da mão e as posições dos dedos são utilizados para simular algumas funções presentes em mouses e teclados, tais como seleccionar estados e objetos e definir direções e posições. Com base nestes protótipos apresentamos algumas conclusões e sugestões para trabalhos futuros.

## Palavras-chave

Visão Computacional; Interação Humano-Computador (IHC); Gestos da Mão; Reconhecimento de gestos; Interação Baseada em Visão.

## Abstract

Quintana Truyenque, Michel Alain. **A Computer Vision Application for Hand-Gestures Human Computer Interaction.** Rio de Janeiro, 2005. 100p. Dissertação de Mestrado - Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro.

Computer Vision can be used to capture gestures and create more intuitive and faster devices to interact with computers. Current commercial gesture-based interaction devices make use of expensive equipment (tracking devices, gloves, special cameras, etc.) and special environments that make the dissemination of such devices to the general public difficult. This work presents a study on the feasibility of using Web cameras as interaction devices based on hand-gestures. In our study, we consider that the hand is clean, that is, it has no (mechanical, magnetic or optical) device. We also consider that the environment where the interaction takes place has the characteristics of a normal working place, that is, without special lights or backgrounds. In order to evaluate the feasibility of such interaction mechanism, we have developed some prototypes of interaction devices. In these prototypes, hand gestures and the position of fingers were used to simulate some mouse and keyboard functions, such as selecting states and objects, and defining directions and positions. Based on these prototypes, we present some conclusions and suggestions for future works.

## Keywords

Computer Vision; Human-Computer Interaction (HCI); Hand Gestures; Gestures Recognition; Vision Based Interaction.



# Sumário

1	Introdução	17
1.1.	Motivação	17
1.2.	Objetivos	18
1.3.	Estrutura do Documento	19
2	Interfaces Baseadas em Visão	21
2.1.	Conceitos Básicos	21
2.2.	Requisitos Funcionais	23
2.3.	Requisitos Não-Funcionais	24
2.4.	Visão Computacional na Interação Humano-Computador em Tempo Real	25
2.4.1.	Vantagens dos Sistemas Baseados em Visão	25
2.4.2.	Desafios	26
2.5.	Trabalhos Relacionados	27
2.5.1.	Sistemas Baseados no Reconhecimento de Gestos	27
2.5.2.	Sistemas Baseados na Detecção dos Dedos	30
2.6.	Sistema de Interação Proposto	32
3	Subtração de Fundo	34
3.1.	Segmentação por Subtração de Fundo	34
3.2.	Requisitos para o Algoritmo de Subtração de Fundo	35
3.3.	Algoritmos Robustos de Subtração de Fundo	37
3.3.1.	Modelo de Cor	38
3.3.2.	Modelagem do Fundo	40
3.3.3.	Subtração ou Segmentação	42
3.3.4.	Seleção dos Limiares	47
3.4.	Resultados	50
3.5.	Conclusões	53
4	Detecção de Silhueta	55
4.1.	Motivação	55
4.2.	Refinamento da Segmentação	56

4.3. Detecção de Bordas	58
4.4. Detecção de Contornos	60
4.5. Seleção da Silhueta da Mão	63
4.6. Considerações Finais do Capítulo	66
5 Reconhecimento	67
5.1. Motivação	67
5.2. Detecção de Feições	69
5.3. Detecção dos Dedos	73
5.4. Reconhecimento de Gestos	75
5.5. Resultados Parciais	77
5.6. Considerações Finais do Capítulo	79
6 Aplicações	80
6.1. Detalhes da Implementação	80
6.2. Controlando o Cursor do Mouse	81
6.3. Controlando um Jogo de Computador	86
6.4. Interação Gestual	89
6.5. Conclusões Parciais	91
7 Conclusões	93
7.1. Trabalhos Futuros	94
Bibliografia	96

## Lista de figuras

- Figura 1 - Interfaces tradicionais e interfaces digitais. Extraído de Bérard [1]. 22
- Figura 2. Resultados do trabalho de Regh & Kanade [30]. O esqueleto é sobreposto na imagem original da mão. No lado direito, o modelo 3D é calculado a partir dos parâmetros encontrados no esqueleto. Extraída de Regh & Kanade [30]. 28
- Figura 3 – Métodos simples para interagir através da Visão apresentados no trabalho de Freemam [8]. (a) Orientação da mão para dirigir um robô. (b) Subtração de imagens para inferir direção do movimento. (c) Histogramas de orientação. 29
- Figura 4 – (a) Extração do contorno da mão com o modelo de curva rígida. Extraído de Heap [12]. (b) Posturas da mão para controlar funções básicas de um jogo de computador. Extraído de Segen [33]. (c) Funções de Gauss (*blobs*) para detectar a mão e os dedos. Extraído de Laptev [22]. 30
- Figura 5 - Sistema “FingerMouse”. Extraído de Queck [29]. 31
- Figura 6 – (a) Interação com o computador — posicionar a ponta do dedo num mesmo lugar por um determinado tempo ocasiona um clique. Extraído de Handenberg [11]. (b) Detecção dos dedos no trabalho de Oka [27]. Extraído de Oka [27]. 32
- Figura 7 – Detecção do contorno da mão apresentado no trabalho de MacKormick [26]. A ponta do dedo detectada é utilizada para desenhar. Extraído de MacKormick [26]. 32
- Figura 8 - Resultado dos algoritmos que lidam parcialmente com sombras; dificuldade na atribuição do limiar e impossibilidade de isolar as sombras; morfologia deformada do objeto. 37
- Figura 9 - Modelo de cor no espaço tridimensional RGB proposto em [14].  $E_i$  representa o valor de cor esperado do ponto  $i$  na imagem de referência,  $I_i$  representa a cor na imagem atual. A diferença entre  $I_i$  e  $E_i$  é decomposta em distorção de brilho ( $a_i$ ) e distorção de cor ( $CD_i$ ). (Extraída de [14]). 39
- Figura 10 - Algoritmo de subtração de fundo baseado na decomposição da diferença de cor em distorção de brilho e distorção de cor. Seguindo a

ordem das operações indicada pelos números, mostram-se as etapas de modelagem e subtração de fundo. A etapa de modelagem é feita uma vez, enquanto que a etapa de subtração é feita a cada imagem (*frame*). 44

Figura 11 – Regiões definidas pelo limiares no espaço bidimensional definido pela distorção normalizada de brilho  $\mathbf{b}$  e a distorção normalizada de cor  $\mathbf{d}$ . 46

Figura 12 - Algoritmo de geração de silhuetas ou de subtração de fundo, baseado no ângulo entre dois vetores.  $C_r$  é a cor atual e  $C_b$  é a cor de referência. 47

Figura 13 - (a) histograma da distorção normalizada de brilho ( $\mathbf{b}$ ), e (b) histograma da distorção normalizada de cor ( $\mathbf{d}$ ). Extraído de [14]. 49

Figura 14 - Resultado da segmentação do algoritmo baseado na medida do ângulo [3]. O limiar utilizado para o ângulo foi de  $\mathbf{q} = 17.49^\circ$ . Nas figuras à esquerda são mostradas cenas com a mão e nas figuras à direita são mostrados os resultados da segmentação. Nos resultados são apresentados a segmentação com as cores originais do objeto e a segmentação binária, onde é mais fácil perceber a morfologia do objeto segmentado. O número de quadros por segundo alcançado foi igual a 30.51

Figura 15 - Resultados do algoritmo baseado na medida da distorção de brilho e a distorção de cor [14]. Desvio padrão mínimo ( $\mathbf{s}' = 1.0$ ), distorção mínima de cor ( $\mathbf{CD}' = 0.75$ ), distorção mínima do brilho ( $\mathbf{a}' = 0.01$ ). Limiares de comparação utilizados na etapa de subtração:  $t_{CD} = 14.1$  para a distorção de cor,  $t_{b} = -80.5$  para a distorção de brilho. Número de quadros por segundo igual a 43. 52

Figura 16 - Teste do algoritmo em [14] aplicando uma janela de 4 pixels. Esta mudança aumenta consideravelmente o número de quadros por segundo sem alterar significativamente a segmentação do objeto. 60 qps. 53

Figura 17 - Máscaras utilizadas nos filtros morfológicos de Dilatação e Erosão; vizinhança 4 ( $N_4$ ) e vizinhança 8 ( $N_8$ ). Além da vizinhança definida pela máscara, mostram-se em preto os pontos que representam o fundo e em branco os pontos que representam os objetos. 57

Figura 18 - Resultados obtidos da aplicação dos filtros morfológicos. Na seqüência (a) imagem resultado da segmentação; (b) resultado da aplicação da Dilatação em (a); e (c) resultado da aplicação da Erosão em (b). Considera-se (c) como resultado final desta etapa. Número médio de

- quadros por segundo igual a 53 qps. 58
- Figura 19 - Obtenção de contornos a partir da subtração das imagens (b) e (c) da Figura 18. O objeto “mão” é representado apenas pelo seu contorno, o mesmo acontece com o ruídos e erros da segmentação. Número médio de quadros por segundo igual a 45 qps. 59
- Figura 20 - Alguns resultados da detecção de contornos realizada em diferentes orientações da mão, (a) presença de ruídos, (b) presença de contornos internos ou buracos. 45 qps. 60
- Figura 21 - Lista de Códigos (**Chain Codes**). Para cada ponto de referência (em azul) existem 4 ou 8 direções possíveis que se podem seguir na representação do contorno. 61
- Figura 22 - Passos do algoritmo de percurso de contornos. A cada passo, novos pontos da silhueta são detectados e alguns pontos “N” são descartados. As setas curvas indicam o sentido de busca, as setas que apontam para os pontos marcados com “X” indicam a direção inicial de busca, as setas que apontam para os pontos marcados com “P” indicam o próximo ponto. 62
- Figura 23 - Pequenos ciclos dentro dos contornos. Estes ciclos representam problemas para o algoritmo de percurso, pois como é indicado em (b) a detecção da silhueta pode acabar muito antes de percorrer o contorno inteiro, o que traz problemas na detecção. 63
- Figura 24 - Resultados do processo de detecção da silhueta da mão aplicados à imagem da Figura 19 em (a), e às imagens da Figura 20 em (b) e (c). Em (a) e (b) são removidos os ruídos e em (c) não é considerada a silhueta interna. Número médio de quadros por segundo igual a 35 qps. 65
- Figura 25 - Subamostragem dos pontos na silhueta detectada. A imagem à direita tem um quinto dos pontos da imagem à esquerda. 66
- Figura 26 - Feições consideradas para o reconhecimento dos gestos. As pontas dos dedos são chamadas de Picos (“P”) e as junções entre eles de Vales (“V”). 69
- Figura 27 - Medida de k-curvatura em cada ponto da silhueta. (a) Para cada ponto  $P_i$  é escolhido um ponto subsequente  $P_{(i+k)}$  e outro anterior  $P_{(i-k)}$  que formam dois vetores. A k-curvatura é o ângulo  $\alpha_i$  formado por esses dois vetores e  $k$  é um inteiro positivo. (b) O parâmetro  $k$  deve ser ajustado de maneira que as regiões de interesse sejam preferentemente detectadas durante o cálculo da k-curvatura. 71

Figura 28 - Resultado da aplicação do algoritmo de detecção de feições à silhueta da mão. As feições são os pontos ressaltados na silhueta. (a) e (b) Mostram duas posturas da mão com os dedos estendidos. Os valores utilizados para o teste foram  $k = 21$  e  $T_a = 60^\circ$ . (c) Mesmo com pequenas mudanças de escala, aproximando-se a mão da câmera, e mantendo os mesmos parâmetros que em (a) e (b) pode-se ver a detecção das feições. 72

Figura 29 - Resultado do processo de detecção dos dedos da mão. Os pontos estão ressaltados na silhueta. 73

Figura 30 - Cálculo da direção dos dedos. (a) A direção aproximada de cada um dos dedos é mostrada como sendo uma linha. (b) A linha que representa a direção do dedo é a que passa pelo ponto  $P_i$  e que melhor aproxima os pontos de referência. Os pontos de referência "x" são os pontos médios de cada par de pontos que estão ligados por uma linha. O ângulo  $q$  é o valor que representa a orientação da linha de direção com respeito à horizontal. 74

Figura 31 - Conjunto predefinido de gestos que são reconhecidos. Os 4 gestos são divididos em três classes: FECHADO, INDICE e ESTENDIDO. FECHADO é uma classe de gesto que não apresenta nenhum dedo estendido. INDICE representa uma classe com dois tipos de gestos: INDICE\_1 (dedo índice estendido) e INDICE\_2 (dedo índice e polegar estendidos). ESTENDIDO é uma classe de gesto que apresenta mais de dois dedos na mão estendidos. 76

Figura 32 - Autômato Finito Determinista que representa o processo de inferência dos gestos com base no número de dedos "d" detectados na silhueta da mão. O único símbolo do vocabulário é "d", que significa "dedo". Os estados de reconhecimento  $q_1$ ,  $q_2$  e  $q_3$  representam os gestos INDICE\_1, INDICE\_2 e ESTENDIDO, respectivamente. 77

Figura 33 - Mecanismo de movimento no cursor. Para cada ponto  $p_1$ ,  $p_2$  e  $p_3$ , a direção do movimento é dada pelas coordenadas  $(x,y)$  do vetor e a velocidade do movimento é dada pela norma do vetor. Em torno do ponto de referência  $O$  existe uma área de inatividade na qual os pontos que caem nessa área ( $p_2$ ) não ocasionam nenhum movimento. 82

Figura 34 - Mecanismo de movimento do cursor. O vetor formado entre a posição do dedo índice e o ponto de referência indica a direção do movimento. A cada quadro este vetor é somado à última posição do cursor. 83

- Figura 35 - Padrão de mudança do evento CLICK. Na nossa aplicação, o intervalo de tempo foi definido em número de quadros,  $T=20$ . Entre os gestos INDICE\_2 pode haver vários gestos INDICE\_1. A partir do gesto corrente (**atual**) é analisada a seqüência em busca do padrão de mudança requerido. 83
- Figura 36 - Evento ARRASTAR. O evento acontece enquanto o gesto INDICE\_1 é mantido. Para validar o evento, o intervalo de tempo em que o gesto INDICE\_1 é mantido deve ser maior do que o  $t$  predefinido. Na nossa aplicação o tempo foi  $t = T-5$ . O tempo  $t$  está relacionado com o tempo  $T$ . 84
- Figura 37 - Diagrama de fluxo do controle do cursor do mouse. Os eventos são validados dependendo do gesto que está sendo processado. O evento CLICK é validado para diferenciar o CLICK do evento MOVIMENTAR. O evento ARRASTAR é validado para diferenciar se o gesto INDICE\_1 indica o evento ARRASTAR o simplesmente faz parte da seqüência do CLICK. 85
- Figura 38 - Gesto ESTENDIDO sendo mapeado para o evento CLICK\_DIREITO do mouse no sistema Windows. Ao gerar o evento no arquivo selecionado, é mostrado o menu disponível no arquivo. 85
- Figura 39 - Gesto INDICE\_2 sendo utilizado para movimentar o ponteiro do mouse. Utilizando o gesto INDICE\_2 podem-se selecionar itens do menu. 86
- Figura 40 - Gesto INDICE\_1 sendo mapeado para o evento ARRASTAR, o qual é utilizado para desenhar letras no PAINT. 86
- Figura 41 - Mecanismo utilizado para calcular a direção do movimento. (a) Mecanismo utilizado para calcular as direções PARA\_FRENTE e PARA\_TRAS. (b) Mecanismo utilizado para calcular as direções DIREITA e ESQUERDA. O valor utilizado na nossa implementação foi  $\alpha = 75^\circ$ . 87
- Figura 42 - (a) Geração do evento ATIRAR utilizando o gesto INDICE\_1. A posição e a direção do dedo índice não indicam movimento. (b) Movimentação do helicóptero utilizando o gesto INDICE\_2. A posição e a direção de dedo indicam os eventos PARA\_FRENTE e PARA\_DIREITA. Neste caso não é gerado o evento ATIRAR. 88
- Figura 43 - (a) Utilizando o gesto INDICE\_1 para gerar o evento ATIRAR e ao mesmo tempo movimentar o helicóptero. A posição e a direção do dedo indicam a direção do movimento PARA\_TRAS e PARA\_ESQUERDA. (b) Gesto ESTENDIDO sendo utilizado para gerar o evento SOLTAR\_EXPLOSIVO. 89

Figura 44 - Estados da aplicação representados como sendo um autômato finito de quatro estados. Os estados **q2** e **q3** geram os eventos AVANÇAR e RECUAR, respectivamente. 90

Figura 45 - Geração do evento AVANÇAR. A mudança do gesto (a) FECHADO para o gesto (b) ESTENDIDO gera o evento AVANÇAR, que é refletido na (c) tela do monitor. 91

Figura 46 - Geração do evento RECUAR. A mudança do gesto (a) FECHADO para o gesto (b) INIDICE\_2 gera o evento RECUAR, que é refletido na (c) tela do monitor. 91



## Lista de tabelas

Tabela 1 - Resultado dos testes feitos para a avaliação do reconhecimento do número de dedos presente na mão. Para cada número de dedos é apresentada uma estatística indicando o número e a porcentagem de acertos e erros ao longo do teste. 78

Tabela 2 - Resultado dos testes feitos para a avaliação do reconhecimento dos gestos predefinidos. Para cada um dos 4 gestos é apresentada uma estatística indicando o número e a porcentagem de acertos e erros ao longo do teste. 79

Tabela 3 - Resultado do teste de reconhecimento dos eventos CLICK e CLICK\_DUPLO. 84