

PESQUISA E DESENVOLVIMENTO DE UMA INTERFACE MULTITOQUE

Ricardo da Silva Werneck

Universidade Federal de Juiz de Fora
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Ciência da Computação
Orientadora: Prof^a. Jesuliana Nascimento Ulysses



Juiz de Fora, MG
Dezembro de 2009

Pesquisa e Desenvolvimento de uma Interface Multitouch

Ricardo da Silva Werneck

Monografia submetida ao corpo docente do Departamento de Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Juiz de Fora, como parte integrante dos requisitos necessários para a obtenção do grau de Bacharel em Ciência da Computação.

Aprovada pela banca constituída pelos seguintes professores:

Prof^a. Jesuliana Nascimento Ulysses - orientadora
M.Sc. em Computação,
UFF/2003

Prof. Dr. Marcelo Bernardes Vieira
Pós-Dr. em Ciência da Computação,
IMPA/2005

Prof. Dr. Michel Heluey Fortuna
D.Sc. em Eng. Sist. e Computação,
UFRJ/2008

Juiz de Fora, MG
Dezembro de 2009

Agradecimentos

Agradeço minha família pelo carinho, apoio e compreensão nesta fase.

Sumário

Lista de Figuras	v
Resumo	vii
Capítulo 1 – Introdução	1
1.1 Divisão da Monografia	1
Capítulo 2 – Sistemas Multitoques	3
2.1 Histórico	3
2.2 Tecnologias de Toque	9
2.3 Uso Comercial dos Sistemas Multitoque	13
Capítulo 3 – Segmentação	16
3.1 Limiarização	18
3.2 Crescimento de Regiões	20
Capítulo 4 – Sistema Desenvolvido	22
4.1 Hardware	22
4.2 Software	25
Capítulo 5 – Testes	31
5.1 Testes de Hardware	31
5.2 Testes de Software	32
5.3 Teste da Interface	37
Capítulo 6 – Conclusão	38
Referências Bibliográficas	39

Lista de Figuras

Figura 2.1.1 - PLATO IV	4
Figura 2.1.2 - Video Place / Video Desk	4
Figura 2.1.3 - Sensor Frame	5
Figura 2.1.4 - Simon	5
Figura 2.1.5 - Diamond Touch – Gestos	6
Figura 2.1.6 - Perceptive Pixel	7
Figura 2.1.7 - iPhone da Apple	7
Figura 2.1.8 - Surface da Microsoft	8
Figura 2.1.9 - Windows 7	8
Figura 2.2.1 - Refração e Reflexão	9
Figura 2.2.2 - Refração e Reflexão - Ângulo Crítico	9
Figura 2.2.3 - Estrutura da Fibra Ótica	10
Figura 2.2.4 - FTIR – Toque modificando o índice de reflexão do acrílico	10
Figura 2.2.5 - Surface – Reflexão de luz traseira	11
Figura 2.2.6 - Difuse Illumination System	11
Figura 2.2.7 - Tela de toque com indutores e receptores	12
Figura 2.2.8 - Tela de toque por pressão	12
Figura 2.3.1 - Jeff Han	13
Figura 2.3.2 - iPhone da Apple – Interface	13
Figura 2.3.3 - Microsoft Surface – Mobile Connection	14
Figura 2.3.4 - iBar System	15
Figura 3.0.1 - Máscara genérica 3x3	16
Figura 3.0.2 - Máscara para detecção de pontos isolados	17
Figura 3.0.3 - Máscara para detecção de linhas	17
Figura 3.1.1 - Influência do valor limiar sobre a Limiarização	18
Figura 3.2.1 - Crescimento de Regiões - Posicionamento de Pixels	20
Figura 3.2.2 - Divisão e Fusão	21
Figura 4.1.1 - Acrílico e leds - Visão superior	22
Figura 4.1.2 - Ligação dos leds - Visão inferior	23
Figura 4.1.3 - Câmera utilizada	23
Figura 4.1.4 - Hardware desenvolvido - Visão interna	24
Figura 4.1.5 - Conectores externos	25
Figura 4.2.1 - Código condensado de captura	26
Figura 4.2.2 - Código fonte para binarização	27
Figura 4.2.3 - Chamada da função recursiva de crescimento de regiões	28
Figura 4.2.4 - Função recursiva de crescimento de regiões	28
Figura 4.2.5 - Aparência do Applet desenvolvido	29
Figura 4.2.6 - Aparência do Applet desenvolvido em funcionamento	30
Figura 5.1.1 - Posicionamento dos leds e acrílico	31
Figura 5.2.1 - Imagem da câmera com um ponto de toque	32
Figura 5.2.2 - Imagem da câmera com dois pontos de toque	33
Figura 5.2.3 - Imagem da câmera com um ponto de toque na borda da interface	33
Figura 5.2.4 - Imagem da câmera com pontos de toque variados	33

Figura 5.2.5 - Imagem da câmera e segmentada com um ponto	34
Figura 5.2.6 - Imagem da câmera e segmentada com múltiplos pontos	35
Figura 5.2.7 - Software em funcionamento com um ponto	36
Figura 5.2.8 - Software em funcionamento com três pontos	36
Figura 5.3.1 - Interface produzida em funcionamento	37

Resumo

Sistemas e Interfaces de Toque estão em evidência nas nossas vidas, desde celulares, terminais de auto-atendimentos de bancos, e computadores. Embora esta tecnologia esteja recebendo grande atenção atualmente, trata-se de uma pesquisa que já nos anos setenta apresentava resultados. Hoje, o grande nome que se lembra ao falarmos em Tela de Toque é o do aparelho celular *iPhone*, que deu origem a uma gama de outros aparelhos concorrentes e também a computadores pessoais. A interação intuitiva entre sistemas multitoque e o usuário é a grande motivação para o desenvolvimento de interfaces homem-máquina. A popularização das tecnologias multitoque ainda está por vir com o lançamento do Windows 7, que promete dar suporte a elas.

Palavras-chave: Telas de Toque, Sistemas Multitoque, Interfaces Homem-Máquina.

Capítulo 1

Introdução

Esse trabalho tem como objetivo apresentar o desenvolvimento de uma interface multitoques. Trata-se de um tema muito discutido atualmente principalmente pelo surgimento de grande número de aparelhos de telefones celulares com telas de toque, que reconhecem um único toque através de sistemas de pressão ou sensores eletrônicos, e multitoque, que combinam interfaces de toque e sistemas de reconhecimento de vários toques.

Estes sistemas, também, estão presentes em terminais de auto-atendimento de bancos, onde para iniciar uma operação basta pressionar a tela sobre o item desejado. Em outros dispositivos, o uso de uma caneta plástica substitui o dedo. Nos computadores, as telas de toque podem substituir o mouse para se tornarem mais simples.

Sistemas que utilizam telas sensíveis a toques tornam-se mais intuitivos para o usuário, já que dispensam outros dispositivos apontadores. Para nós, é comum utilizarmos de gestos para nos comunicar. A passagem de gestos para uma interface que os interpreta torna esta interface mais natural para o nosso uso. Este é um dos motivos que atraíram a atenção na interface do *iPhone*, gestos como arrastar o dedo lateralmente na tela para rolagem lateral, nos lembra a passagem de páginas de um livro. Outro exemplo é o uso de dois dedos girando sobre uma imagem que a faz rotacionar.

Sistemas multitoques são compostos de *hardware* e *software* para proporcionar um controle maior de eventos. Neste trabalho foi produzido um protótipo de *hardware* e desenvolvido um *software* com a finalidade de simular um sistema multitoque.

1.1 Divisão da Monografia

O trabalho está organizado em 6 capítulos, incluindo a introdução.

O segundo capítulo trata de Sistemas Multitoque, apresenta conceitos, motivações e um breve histórico desta tecnologia, além de relatar sistemas conhecidos e suas características.

No terceiro capítulo, aborda-se a técnica de segmentação de imagens, conceitos e aplicações. A segmentação foi usada no desenvolvimento do *software* para poder segmentar a

região do toque. As técnicas de segmentação utilizadas foram limiarização (*Threshold* ou binarização) e crescimento de regiões.

No quarto capítulo, é apresentado o sistema desenvolvido, envolvendo *hardware* e *software*. Demonstra-se a estrutura do *hardware* e trechos do código produzido com uma breve explicação.

No quinto capítulo, os testes são apresentados. Para este trabalho os testes estão restritos a captura, segmentação da imagem e identificação de pontos na imagem resultante.

Com base nos resultados obtidos, a conclusão é relatada no sexto capítulo juntamente com a proposta para trabalhos futuros.

Capítulo 2

Sistemas Multitoques

Telas sensíveis a toques ou *Touch-Screens* são interfaces que combinam monitores e mecanismos apontadores. Assim como nos computadores o mouse é usado para clicar sobre ícones e opções de menus que são visíveis no monitor, nas telas de toque é necessário apenas pressionar o dedo sobre a opção desejada, dispensando o mouse. Mas estas telas, muito utilizadas em terminais de auto-atendimento de bancos, são construídas para a utilização de apenas um ponto de contato por vez.

Entende-se por Sistemas Multitoques as interfaces que surgem da combinação de telas sensíveis ao toque, com sistemas que reconhecem múltiplos pontos assinalados nestas superfícies. Seja o toque de muitos dedos ou de outros objetos, esta interação aumenta muito a complexidade no controle de ações.

2.1 Histórico

Sistemas de toques estão datados de antes dos anos 70, quando as telas de toque começam a ser desenvolvidas pela IBM, na Universidade de *Illinois* e *Ottawa* no Canadá. Em 1971, diferentes técnicas haviam sido divulgadas, como o uso de *Touchpad*, semelhantes aos usados em notebooks atuais para substituir o mouse. Estes *Touchpads* foram utilizados em sintetizadores de música, mas cada *Touchpads* era de um único toque. [11]

Logo em seguida, em 1972, foi implantado o PLATO IV- *Touch Screen Terminal*, um terminal de pesquisa educacional assistida da Universidade de Illinois. A tecnologia usada era semelhante ao uso de infravermelhos de hoje, porém continha uma matriz restrita de 16x16 posições possíveis para o toque. [11]



Figura 2.1.1 PLATO IV. [11]

Em 1983, Myron Krueger apresenta o *Video Place / Video Desk*, que inova por utilizar vídeo para detectar posição de dedos e mãos, mesmo de várias pessoas. Outro ponto que se destaca neste sistema é o fato de possuir uma vasta biblioteca de gestos. Combinava não apenas o toque, mas também o tempo do toque para certos eventos. Foi uma das primeiras interfaces que tinha configuração horizontal, ou seja, uma interface mais ergonômica para o usuário, e remete ao uso de uma mesa.



Figura 2.1.2 Video Place/Video Desk. [11]

Apresentado em 1985, *Sensor Frame*, trabalho de Paul McAvinney, utilizava sensores ópticos nos cantos, como molduras para identificar os pontos. A tecnologia de sensores foi utilizada, pois ainda era difícil a utilização de câmeras com tamanho reduzido. O *Sensor Frame* podia detectar até três dedos ao mesmo tempo sem que houvesse interferência entre eles. Mais tarde, com incentivo da NASA, surgiu o *Sensor Cube*, dispositivo semelhante ao *Sensor Frame*, que podia identificar também o ângulo que os dedos “entravam” na tela. [11]



Figura 2.1.3 Sensor Frame. [11]

O primeiro telefone celular com tela de toque foi o *Simon*, desenvolvido pela IBM e *Bell South*, em 1992, utilizava apenas de um toque, mas já era possível vislumbrar o futuro dos *Smart-Phones* sem os teclados mecânicos.



Figura 2.1.4 Simon. [11]

Durante anos, muitas dessas tecnologias já mencionadas foram aprimoradas, outras foram substituídas, dando origem a vários projetos que utilizavam de conhecimento gestual para atribuição de comandos.

É o caso da *Diamond Touch*, 2001, produzido pela *Mitsubishi Research Labs* em *Cambridge*, que era capaz de distinguir dedos e mãos de pessoas, bem como a localização e a pressão dos gestos. Ele proporcionou um repositório gestual mais rico. [11]



Figura 2.1.5 Diamond Touch - Gestos. [11]

Nos anos de 2005 e 2006, Jeff Han apresenta uma tecnologia baseada no fenômeno físico de Perturbação da Reflexão Total Interna (FTIR - *Frustrated Total Internal Reflection*) em superfícies como o acrílico. Este fenômeno será mais detalhado neste trabalho, no capítulo referente ao *hardware* desenvolvido (Seção 4.1). Associando câmeras, luzes infravermelhas e uma superfície acrílica, Han lançou *Perceptive Pixel*, uma interface de projeção traseira acompanhada por aplicativos e técnicas bem desenvolvidas. O criador fundou a empresa *Perceptive Pixel* para abranger o mercado privado.



Figura 2.1.6 Perceptive Pixel. [7]

Em 2007 foi lançado o *iPhone*, aparelho celular da *Apple* que utiliza da tecnologia multitoque em sua tela., O controle do aparelho é feito, quase exclusivamente, através de toque. Utiliza uma biblioteca de gestos bem adaptada para suas funções e tamanho de tela.



Figura 2.1.7 iPhone da Apple. [10]

Também lançada, em 2007, a *Surface* da *Microsoft*, uma interface em forma de mesa, com tela sensível a multitoques, e diversas implementações específicas como editar arquivos de fotos com o toque dos dedos, ou conectar dispositivos móveis apenas apoiando-os sobre a superfície.



Figura 2.1.8 Surface da Microsoft. [6]

Em 2009, o Windows 7, sistema operacional da Microsoft, prevê o suporte a multitoque. Trata-se de uma tentativa da Microsoft de abranger, com o sistema, a opção de utilizar periféricos com suporte a multitoque como monitores.



Figura 2.1.9 Windows 7 [21].

2.2 Tecnologias de toque

Muitos *hardwares* multitoques baseiam-se no fenômeno físico de Perturbação da Reflexão Total Interna (FTIR - *Frustrated Total Internal Reflection*).

FTIR é o fenômeno óptico de luz que ocorre quando um feixe de luz atravessa de um meio para outro em um determinado ângulo, acima do ângulo crítico, forçando a luz a ficar totalmente refletida no meio de entrada. Este ângulo crítico ocorre apenas quando o meio de saída do feixe tem índice de refração menor do que o meio de entrada, como por exemplo, do vidro pra o ar. [17]

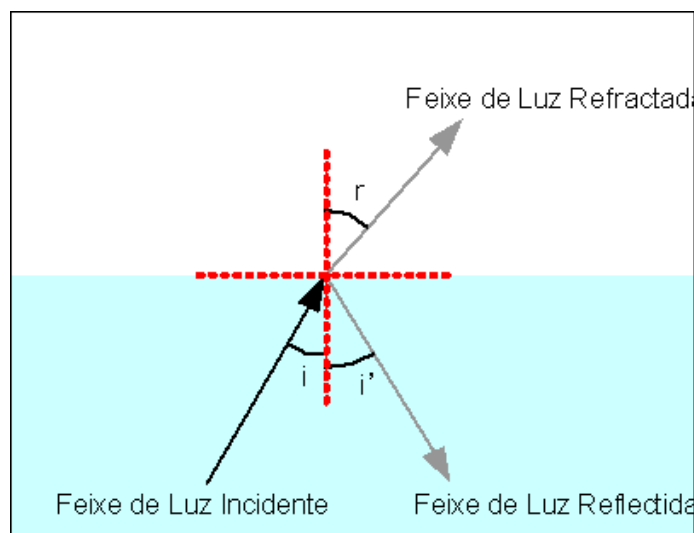


Figura 2.2.1 Refração e Reflexão. [23]

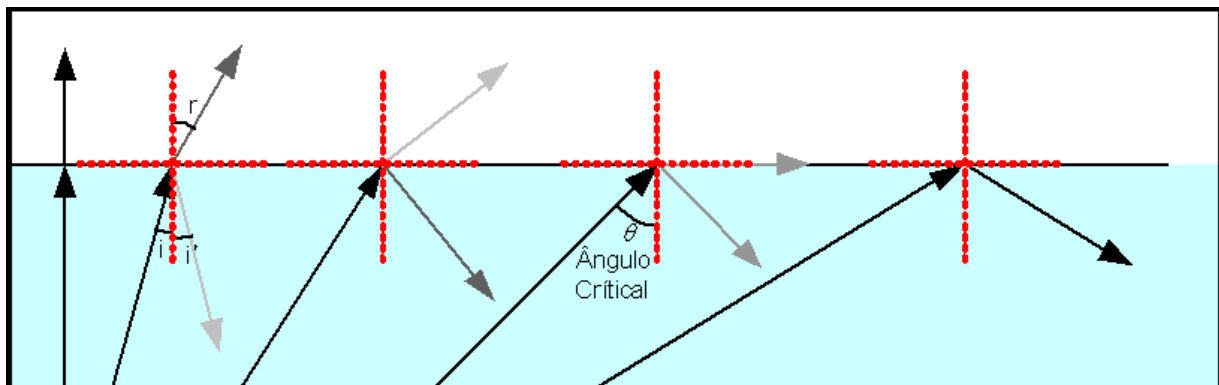


Figura 2.2.2 Refração e Reflexão – Ângulo Crítico. [23]

A fibra-ótica é um exemplo onde a FTIR é utilizada. Na camada interna da fibra o feixe de luz está contido, envolto em uma camada refletora e coberto por uma camada plástica. Tais camadas funcionam como um isolamento do meio para evitar perdas.

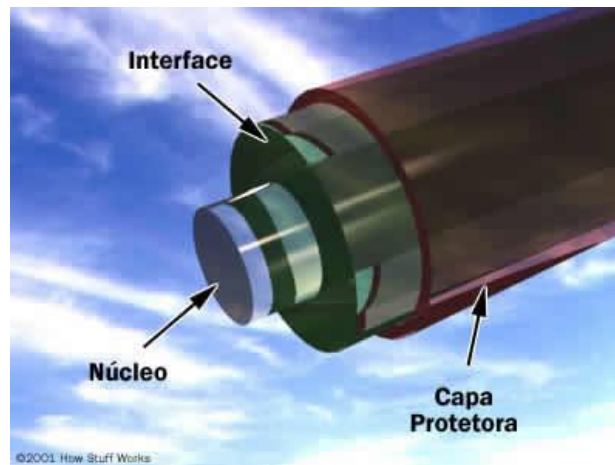


Figura 2.2.3 Estrutura da Fibra Ótica. [18]

Utilizando-se destes princípios, quando ocorre a mudança no índice de refração na fronteira do meio, o ângulo do feixe de luz se modifica e em uma condição específica o feixe de luz sai do meio que estava aprisionado.

Em *hardwares* multitoques baseados em FTIR, o feixe de luz é proveniente de *leds* infravermelhos posicionados perpendiculares a espessura de uma placa de acrílico. Durante o funcionamento do *hardware*, o feixe de luz infravermelha fica retido dentro da placa de acrílico até que se haja um toque na superfície da placa e mude o índice de reflexão fazendo com que a luz escape.

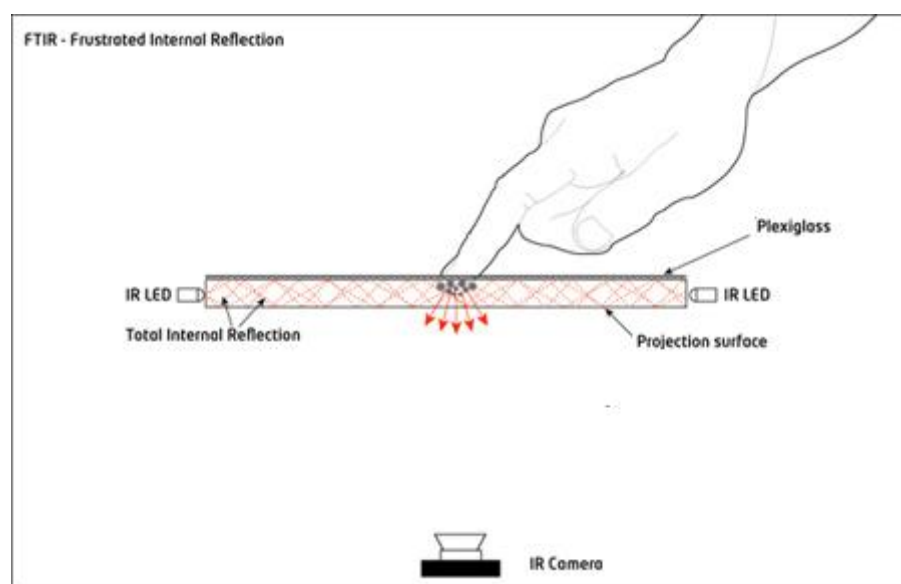


Figura 2.2.4 FTIR – Toque modificando o índice de reflexão do acrílico. [9]

O uso de acrílico, *leds* infravermelhos e câmera são a base para o *hardware* de muitas interfaces multitoques baseadas em captura de luz, como no projeto *Surface* da *Microsoft* que não utiliza FTIR. Na *Surface*, um feixe de luz é jogado diretamente contra a superfície onde ocorre o toque provocando a reflexão da luz em direção a câmera (Figura 2.2.5).

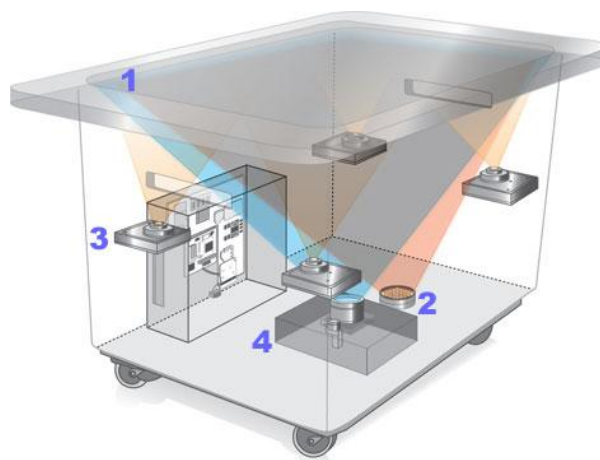


Figura 2.2.5 Surface – Reflexão de luz traseira. [24]

Este tipo de interface possui iluminação difusa (DI), e a principal diferença com FTIR é a capacidade de identificar objetos próximos a interface, mesmo que não esteja em contato com a superfície [25]. Nestas interfaces o uso de vidro, em contrapartida ao acrílico, não interfere em seu funcionamento. Na figura 2.2.6 verificamos a estrutura de sistemas DI.

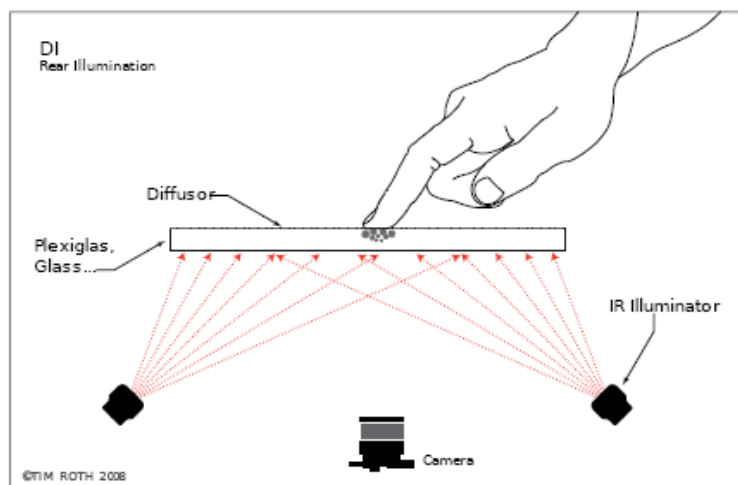


Figura 2.2.6 Difuse Illumination System. [25]

Em aparelhos portáteis como celulares e *palms*, a tela é composta de sensores e indutores eletrônicos para detecção de toques por pressão (figura 2.2.7).

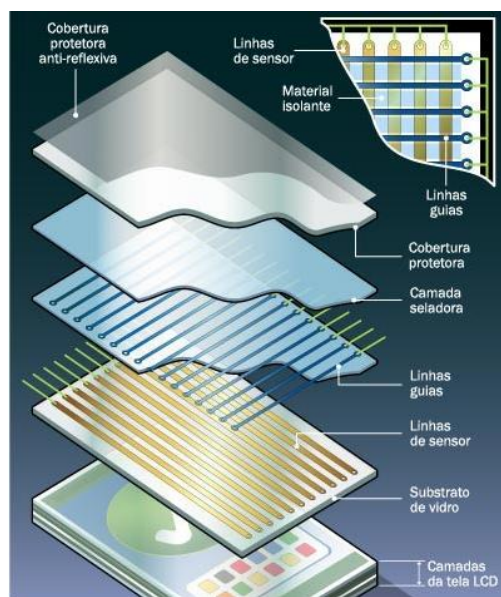


Figura 2.2.7 Tela de toque com indutores e receptores. [5]

Tais sistemas consistem em duas superfícies condutoras separadas por uma camada isolante, geralmente composta por minúsculos pontos de silicone, e um controlador elétrico que gera correntes nas camadas condutoras. Quando o usuário toca na tela é medida a corrente na horizontal e na vertical para determinar o ponto. As vantagens desta tecnologia é o consumo baixo de energia e a possibilidade do toque com *stylus*, caneta para estes aparelhos, ou com os dedos. [25]. A estrutura durante o toque pode ser verificada na figura 2.2.8.

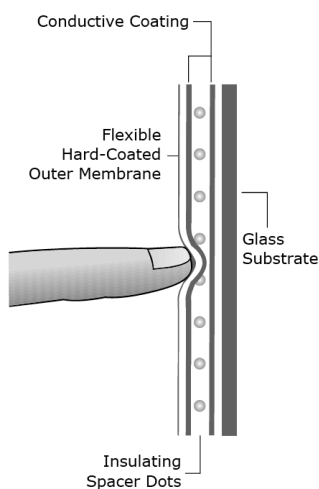


Figura 2.2.8 Tela de toque por pressão. [25]

2.3 Uso comercial dos sistemas multitoque

Jefferson Han fundou a empresa *Perceptive Pixel* para registrar a sua tecnologia, a idéia fundamental é utilizar luz projetada sobre uma superfície acrílica que captura os movimentos das mãos e os transformam em reflexos interpretados por sensores, que traduzem em imagens. Uma idéia simples que tornam essas telas fácil de usar. A intuitividade é o marco nessa tecnologia.



Figura 2.3.1 Jeff Han. [22]

Segundo Han: “O toque é uma das coisas mais intuitivas do mundo”. Para o criador dessa tecnologia de multitoque: “Em vez de usar mouse ou o teclado, você pode manipular imagens diretamente” [22].

Quando o *iPhone* foi lançado em 2007 e o público pode testar sua tela de toque, essa tecnologia foi amplamente divulgada e procurada. Novos aparelhos surgiram para competir pelo consumidor.



Figura 2.3.2 iPhone da Apple - Interface. [5]

A tela sensível a toque trouxe aos aparelhos celulares a oportunidade de reduzir seu tamanho, se comparado um aparelho que utiliza teclado tradicional e outro que possui tela de toque, é visível a diferença principalmente na sua espessura. Em um aparelho tradicional as peças mecânicas que compõem o teclado ocupam grande parte do espaço na carcaça, isso limita a miniaturização dos aparelhos. Para uma tela de toque, embora o tamanho de visualização seja importante, o uso de funções de zoom facilita a exposição de mais informações na tela, aquela informação que o usuário deseja pode receber mais destaque enquanto outras são ocultadas ou suprimidas, sem que uma área seja dedicada ao teclado.

Com o uso dessa tecnologia os aparelhos ganham no apelo visual, já que na tela pode ser projetada qualquer configuração visual, possibilitando uma elaborada interface que também pode ser personalizada pelo usuário.

Em computadores a tela sensível a toque pode ser considerado o maior avanço de interfaces desde o uso do mouse. A redução de periféricos pode ser drástica, uma tela de toque pode substituir monitor, mouse e teclado. Embora ainda haja uma resistência na substituição do teclado, os avanços na pesquisa de materiais e desenvolvimento do *software* para sistemas multitoques prometem reverter essa desconfiança.

Para não perder tempo a Microsoft lança a *Surface*, já citada nas seções 2.1 e 2.2, uma mesa composta por uma tela de 30 polegadas, câmeras infravermelhas que detectam os movimentos próximos a tela e um *software*, que pode ser considerado embarcado, pois só alguns aplicativos específicos estão disponíveis. Segundo Bill Gates, a intenção é futurista com uma interação mais natural por voz, mãos ou canetas.



Figura 2.3.3 Microsoft Surface – Mobile Connection. [6]

A conectividade com outros aparelhos como celulares, merece destaque, bastando apenas apoiar o aparelho sobre a mesa e esta se conecta automaticamente ao aparelho por infravermelho ou *wireless*. Quando essa conectividade acontece, ao redor do aparelho é disposto as funcionalidades do mesmo, como por exemplo transferência de fotos do aparelho para a mesa, que se dá com o arrastar da foto desejada para fora da região delimitada ao celular.



Figura 2.3.4 iBar System. [8]

Uma das possibilidades do uso comercial de interfaces multitoque é o iBar, trata-se de um balcão que reconhece objetos sobre a sua superfície, e projeta imagens sobre a superfície nos pontos onde estes objetos tocam. Composta por vários projetores interagindo com a superfície multitoque, e algumas animações modificadas conforme a movimentação dos objetos.

Capítulo 3

Segmentação

Segmentação é a operação que objetiva particionar uma imagem em um conjunto de regiões que correspondam a elementos de interesse na cena representada pela imagem. [15]

Cada região deve ser homogênea e uniforme respeitando propriedades, como tom e textura. O nível que gera a divisão das regiões depende da aplicação, já que os valores nestas propriedades são diferentes entre regiões vizinhas. A segmentação chega ao fim quando a imagem obtida atende aos interesses.

Os algoritmos de segmentação são geralmente baseados nas propriedades de descontinuidade ou similaridade de valores dos níveis de cinza da imagem. A abordagem por descontinuidade consiste em dividir a imagem em áreas onde ocorrem mudanças bruscas nos níveis de cinza para a detecção de pontos isolados, linhas e bordas na imagem. Utilizando similaridade a imagem é segmentada em regiões similares de acordo com um algum critério predefinido. Métodos mais utilizados nesta abordagem são limiarização, crescimento de regiões e divisão e fusão de regiões [15].

Deteção de descontinuidades consiste em varrer a imagem, utilizando se de uma máscara (figura 3.0.1), para comparar os tons de cinza entre regiões vizinhas. Cada região da máscara possui seu coeficiente w , representando o valor do pixel e seu nível de cinza. A comparação é definida em relação a posição central.

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

Figura 3.0.1 Máscara genérica 3x3. [15]

Para a detecção de pontos isolados a máscara utilizada possui na região central um coeficiente maior do que nas regiões vizinhas (figura 3.0.2). Conhecido como Operador Laplaciano, este coeficiente define que o nível de cinza do ponto central é completamente diferente dos outros pontos na máscara.

-1	-1	-1
-1	8	-1
-1	-1	-1

Figura 3.0.2 Máscara para detecção de pontos isolados. [15]

Em um nível maior de complexidade a detecção de linhas em uma imagem pode ser determinada utilizando-se das máscaras de operador Sobel, figura 3.0.3. Estas máscaras identificam posições na imagem onde os níveis de cinza estão alinhados seja horizontal, vertical ou em 45 graus.

-1	-1	-1	-1	-1	2
2	2	2	-1	2	-1
-1	-1	-1	2	-1	-1
Horizontal			+45°		
-1	2	-1	2	-1	-1
-1	2	-1	-1	2	-1
-1	2	-1	-1	-1	2
Vertical			-45°		

Figura 3.0.3 Máscara para detecção de linhas. [15]

3.1 Limiarização

Limiarização ou *Thresholding* pode ser entendida com a conversão de uma imagem de entrada $f(x,y)$ em uma imagem binária $g(x,y)$, ou seja, após a escolha de um limiar T , todos os pixels que possuem valor menor que T são transformados em 0 e os outros pixels recebem valor 1. Neste caso tem-se a limiarização simples definida na equação 1.

$$g(x,y) = \begin{cases} 1 & \text{se } f(x,y) \geq T \\ 0 & \text{se } f(x,y) < T \end{cases} \quad (1)$$

Está técnica é muito usada quando a imagem é composta de objetos homogêneos em um fundo uniforme.

Caso se tenha dois limiares tem-se a limiarização múltipla [15], que admite dois limiares T_1 e T_2 , e é definida na equação 2:

$$g(x,y) = \begin{cases} 1 & \text{se } T_1 \leq f(x,y) \leq T_2 \\ 0 & \text{nos demais} \end{cases} \quad (2)$$

A limiarização pode ser feita com os mesmos valores de limiar para toda a imagem (limiarização global), ou podem ser definidos valores de limiar diferentes para áreas distintas (limiarização local).

Um dos problemas nesta técnica é a escolha do limiar T , que pode ser automático ou manual. Há vários métodos para calculo do limiar que podem influenciar muito no resultado, o mais utilizado é o baseado na análise do histograma.[15]

Na figura 3.1.1 apresenta-se a limiarização da imagem da figura 3.1.1 a) com diferentes valores para o limiar.



a) Imagem Original



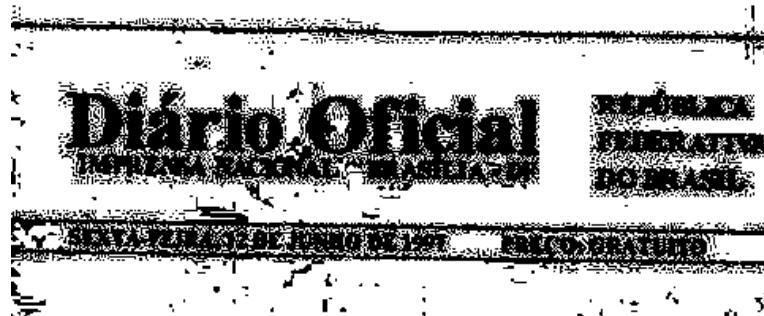
b) Histograma



c) Limiarização com valor 50



d) Limiarização com valor 150



e) Limiarização com valor 235

Figura 3.1.1 Influência do valor do limiar sobre a Limiarização. [13]

Os valores de limiar utilizados nas figuras 3.1.1-c, 3.1.1-d e 3.1.1-e foram retirados do histograma da figura 3.1.1-b.

3.2 Crescimento de Regiões

Crescimento de regiões é o método de agrupamento de pixels em regiões, ou de regiões a outras maiores. Começa a partir de um ponto, geralmente um pixel “semente” e cresce buscando as condições necessárias para se agrupar a uma região. São condições para um pixel: ser vizinho da região, não participar de outra região e manter a homogeneidade da região.

Duas formas de implementar o algoritmo de crescimento de regiões é o uso de função recursiva ou o uso de pilhas. Neste trabalho foi escolhido o uso de função recursiva e é este que será exemplificado.

Os algoritmos de crescimento de regiões são baseados na comparação com os primeiros quatro vizinhos.

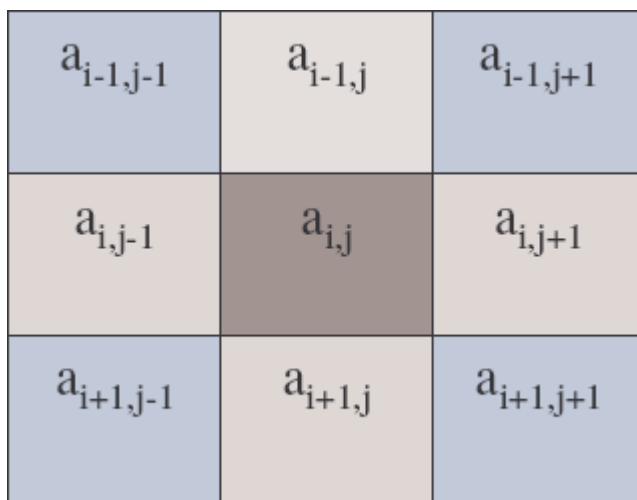


Figura 3.2.1 Crescimento de Regiões – Posicionamento de Pixels.[16]

O algoritmo de crescimento de região parte de um pixel base definido “semente” e faz a comparação com seus vizinhos conforme o critério de agregação. Se o critério for aceito este vizinho é incorporado e passa a ser o pixel base. Quando um vizinho é escolhido, os outros devem “ficar em espera”, para retornar quando necessário. Se não for aceito volta-se para os vizinhos em “espera” até que todos sejam verificados.

Uma vez que todos os vizinhos tenham sido verificados e o algoritmo retornado ao pixel “semente”, o critério de parada foi aceito e os pixels foram adicionados a região.

Durante este processo é importante que os pixels utilizados sejam marcados de alguma forma para que não voltem a ser reutilizados, caracterizando um loop infinito no algoritmo.

Uma alternativa ao procedimento de crescimento de regiões a partir de um pixel, que não foi utilizada neste trabalho, é a divisão e fusão de regiões em uma imagem. Consiste em dividir a imagem em regiões iguais e verificar se uma condição é satisfeita; se for negada a região é dividida novamente. Quando todas as regiões satisfazem a condição, passa-se a fundir regiões com características semelhantes até que se formem regiões com características distintas. Esse procedimento é ilustrado na figura 3.2.2.

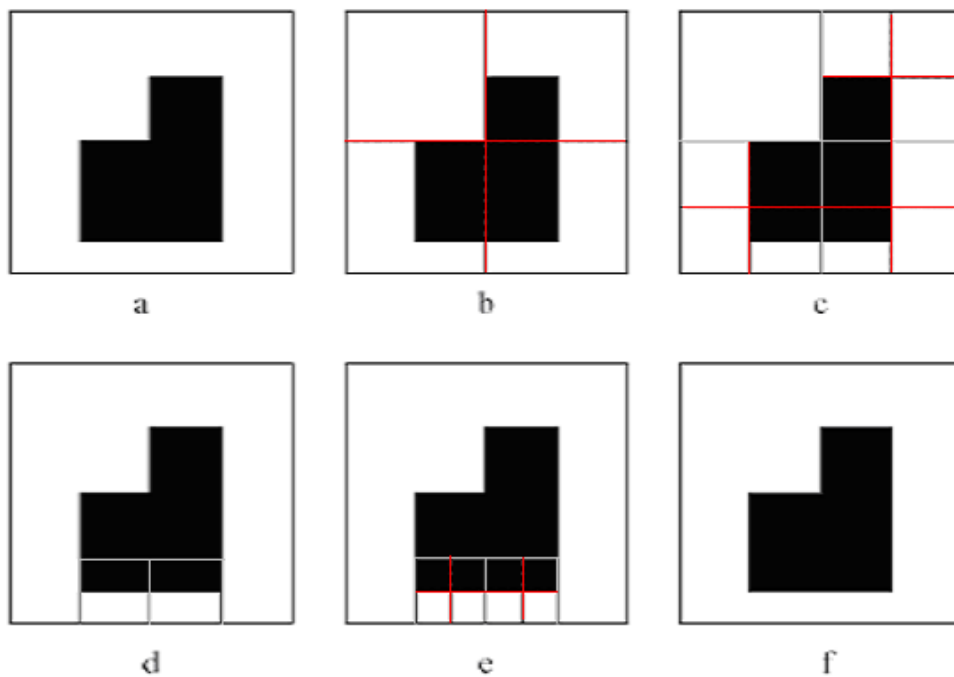


Figura 3.2.2 Divisão e Fusão. [15]

Na Figura a técnica utiliza a estrutura de dados denominada *QuadTree* para codificar imagens, que consiste na montagem de uma árvore onde a raiz é a imagem inicial e os nodos “filhos” são os quatro quadrantes após a divisão. A característica analisada é a cor única em uma área, ou seja, se uma área é totalmente branca ou preta. A figura 3.2.2-a, representa a imagem inicial que será dividida em quatro quadrantes, figura 3.2.2-b; aqueles quadrantes que não apresentarem uma única cor serão subdivididos em outros quatro, figura 3.2.2-c; que, se necessário, serão subdivididos em quatro quadrantes, figura 3.2.2-e; até que resultem na identificação da área de cor branca e na área de cor preta, figura 3.2.2-f.

Capítulo 4

Sistema Desenvolvido

4.1 Hardware

O *hardware* desenvolvido para este trabalho consiste em uma mesa com as seguintes dimensões, 90 cm de altura, 70 cm de comprimento por 60 cm de largura. Para a construção da base foi utilizado madeira.

A superfície de toque consiste em uma placa de acrílico de 8 mm de espessura, 65 cm de comprimento e 55 cm de largura. Tal acrílico foi posicionado sobre uma moldura de madeira onde foram fixados *leds* infravermelhos (Figura 4.1.1).

Foram utilizados 72 *leds* infravermelhos de 5 mm de espessura, com voltagem aproximada de 3v e amperagem de 0,3 μ A. Os *leds* foram conectados a um circuito misto, composto por 18 grupos paralelos de 4 *leds* em série, o que proporcionou uma corrente de 12 volts. Este circuito contém também dois *coolers*, para resfriamento interno, conectados paralelamente aos *leds* e com 12 volts de corrente. Para alimentar o sistema utiliza-se uma fonte de 12 volts dc com 800 μ A.



Figura 4.1.1 Acrílico e *leds* – visão superior.



Figura 4.1.2 Ligação dos leds – Visão inferior.

No interior do *hardware* encontra-se um projetor multimídia *Epson* modelo *PowerLite S5+*, que projeta sua imagem em direção a um espelho, posicionado em ângulo, que reflete a imagem até um anteparo de papel vegetal, posicionado abaixo da placa acrílica.

A câmera utilizada neste *hardware* consiste em uma *webcam* de marca *XPC*, sensor *CMOS VGA 1/3*, com resolução máxima *640 x 480 pixels* e conexão *USB*. Neste trabalho a câmera está utilizando resolução de *320 x 240 pixels* e taxa de 30 quadros por segundo.

A fixação da câmera é feita sobre uma haste de madeira, sobre o projetor, centralizada em relação ao acrílico. Para melhor fixação foi retirada a carcaça da câmera como pode ser vista na figura 4.1.3.



Figura 4.1.3 Câmera utilizada.

Para melhoria da imagem a câmera foi modificada trocando-se o filtro original, que bloqueava os raios infravermelhos, por filtros de filme fotográfico para reduzir a passagem da luz e permitir somente a passagem de luz infravermelha. Também foi alterada a lente da câmera para se obter um ângulo maior de visão em um espaço mais reduzido, para isso foi utilizado lente angular, utilizada em câmeras de segurança, com distancia focal mínima de 30 centímetros.

Na figura 4.1.4 é possível identificar o projetor, o espelho, a câmera, um dos *coolers* de exaustão e as conexões eletroeletrônicas internas.



Figura 4.1.4 Hardware desenvolvido - Visão interna.

Além das estruturas comentadas, conectores de vídeo, de energia, USB e interruptores foram instalados externamente ao *hardware* para melhor conectividade (figura 4.1.5).



Figura 4.1.5 Conectores externos.

4.2 Software

O *software* desenvolvido neste trabalho tem como etapas principais a captura da imagem, a segmentação desta imagem com limiarização (*Threshold*) e crescimento de regiões, e finalizando com a identificação e representação dos pontos de toque.

Além dos códigos destas etapas principais, que serão explicados neste capítulo, implementou-se um *Applet* simplificado onde é definido uma área para exibir a entrada de vídeo da câmera sem tratamento, outra área para a imagem segmentada e uma terceira área ilustrando a relação dos pontos identificados com a área de trabalho no computador.

Para auxiliar no desenvolvimento foi utilizado um *framework* de media, *Java Media Framework* (JMF), que estende o Java 2 *Standard Edition*, habilitando para desenvolvedores multimídia um controle maior na captura, reprodução, transmissão e decodificação de mídias.

Na etapa de captura foi implementado o trecho de código, que segue condensado, na figura 4.2.1.

```
32     public static Player player = null;
55     MediaLocator medLocat = new MediaLocator("vfw://0");
57     player = Manager.createRealizedPlayer(medLocat);
58     player.start();
76     FrameGrabbingControl fgc = (FrameGrabbingControl)
77         player.getControl("javax.media.control.FrameGrabbingControl");
78     Buffer buf = fgc.grabFrame();
79     BufferToImage bufToImg = new BufferToImage((VideoFormat) buf.getFormat());
80     Image img = bufToImg.createImage(buf);
```

Figura 4.2.1 Código condensado de captura.

Neste trecho é criado um *Player*, que recebe a entrada de mídia ‘vfw://0’ referente ao vídeo da câmera USB utilizada no *hardware*. Em seguida é criado um *FrameGrabbingControl*, que é um controlador para o *Player*.

Este controlador é utilizado para obter um frame do vídeo que será transformado em dados para um *Buffer*. Este buffer armazena os dados do frame capturado até que seja passado para outro buffer específico de imagem, *BufferToImage*. Neste buffer de imagem, os dados recebem formatação correspondente a do vídeo, como largura, altura e resolução. Com os dados já formatados cria-se a imagem que será segmentada.

A primeira parte da etapa de segmentação é a limiarização da imagem, ou seja, a passagem da imagem original para imagem binarizada. O código fonte para esta binarização é apresentado na figura 4.2.2.

```

84 int[] pixels = new int[wid*hei];
85 bufImg = (BufferedImage) bufToImg.createImage(buf);
86 int red = 0;
87 int green = 0;
88 int blue = 0;
89
90 PixelGrabber pg = new PixelGrabber(bufImg,0,0,wid,hei,pixels,0,wid);
91 try {
92     pg.grabPixels();
93 } catch (InterruptedException e1) {
94     e1.printStackTrace();
95 }
96
97 for(int p=0;p<pixels.length;p++){
98     blue = (pixels[p] & 0xff);
99     green = (pixels[p] >> 8) & 0xff;
100    red = (pixels[p] >> 16) & 0xff;
101    if(blue > 200 && green > 200 && red > 200){
102        pixels[p] = Color.BLACK.getRGB();
103    }
104    else{
105        pixels[p] = Color.WHITE.getRGB();
106    }
107 }
108 bufImg.setRGB(0, 0, wid, hei, pixels, 0, wid);
109 Image image = Toolkit.getDefaultToolkit().createImage(bufImg.getSource());
110 imgpanel.setImage(image);

```

Figura 4.2.2 Código fonte para binarização.

Para iniciar a binarização é criado um vetor de inteiros para armazenar os pixels da imagem, tal vetor tem o tamanho exato para o número total de pixels da imagem. Em seguida é criado um *PixelGrabber* para passagem dos pixels do buffer para o vetor de inteiro. E também são criadas três variáveis do tipo *int* para comparação.

O processo ocorre dentro de um loop no tamanho do vetor de pixels. Para cada passagem nesse loop atribuem-se às variáveis os valores referente a cada cor dos pixels.

Neste ponto foi definido o limiar como sendo a composição do valor RGB para azul, verde e vermelho maiores que 200, significando uma cor próxima ao branco total. Este valor de 200 foi definido por experimentação para um resultado satisfatório na sensibilidade do toque sobre a superfície do acrílico.

Quando o valor da cor do pixel é próximo ao branco, significa que este pixel pertence a uma região onde ocorreu um toque. Então este pixel recebe o valor correspondente a cor preta, caso contrário recebe a cor branca.

Ao final deste processo todos os pixels do vetor têm a cor branca ou preta, e com esse vetor é gerada uma imagem binarizada, que em seguida é atribuída a um painel da interface *Applet*.

Finalizada a limiarização passa-se a segunda parte da segmentação, com o processo de crescimento de regiões. Neste processo começa-se a definir os pontos de toque.

Começa-se o processo percorrendo todos os pixels da imagem binária verificando se estes são de cor preta. Em caso afirmativo o código da figura 4.2.3 é executado para verificar se o pixel já foi usado em uma região. Se ele não pertence a nenhuma região é chamada a função *calcRegion*, função esta que contém o algoritmo de crescimento de regiões implementado como na figura 4.2.4.

```
116     int[] position = new int[2];
117     position[0] = i;
118     position[1] = j;
119     if(isUsed(position) == false){
120         calcRegion(i,j);
121         int[] flag = new int[2];
122         flag[0] = -1;
123         flag[1] = -1;
124         v.add(flag);
125     }
```

Figura 4.2.3 Chamada da função de crescimento de regiões.

```
148 public void calcRegion(int i,int j){
149     int[] point = new int[2];
150     point[0] = i;
151     point[1] = j;
152     if(isUsed(point) == false){
153         v.add(point);
154         if((i<bufImg.getWidth()-1)&& (bufImg.getRGB(i+1, j)==Color.BLACK.getRGB()))
155             calcRegion(i+1,j);
156         if((j<bufImg.getHeight()-1)&& (bufImg.getRGB(i, j+1)==Color.BLACK.getRGB()))
157             calcRegion(i,j+1);
158         if((i>0)&& (bufImg.getRGB(i-1, j)==Color.BLACK.getRGB()))
159             calcRegion(i-1,j);
160         if((j>0)&& (bufImg.getRGB(i, j-1)==Color.BLACK.getRGB()))
161             calcRegion(i,j-1);
162     }
163 }
```

Figura 4.2.4 Função recursiva de crescimento de regiões.

Estas partes do código utilizam um vetor de pares ordenados que armazenam as posições dos pontos pertencentes a alguma região. Durante a função recursiva, os pontos pertencentes a uma mesma região ficam agrupados no vetor e, ao final, é adicionado um marcador para separá-los de outros pertencentes a regiões diferentes.

Terminado a verificação de todos os pixels da imagem, o vetor de pares ordenados contém todos os pontos pertencentes a alguma região. É neste vetor que se aplica uma função para cálculo das coordenadas centrais das regiões, que percorre o vetor procurando os pontos extremos de uma região e calculando a média.

Utilizando as coordenadas centrais das regiões, são desenhados, na imagem que representa a área de trabalho, pontos vermelhos nas coordenadas correspondentes. Estes pontos são meramente ilustrativos e possuem tamanho fixo.

A aparência do *Applet* desenvolvido é mostrada na figura 4.2.5. E na figura 4.2.6 o *Applet* em funcionamento.

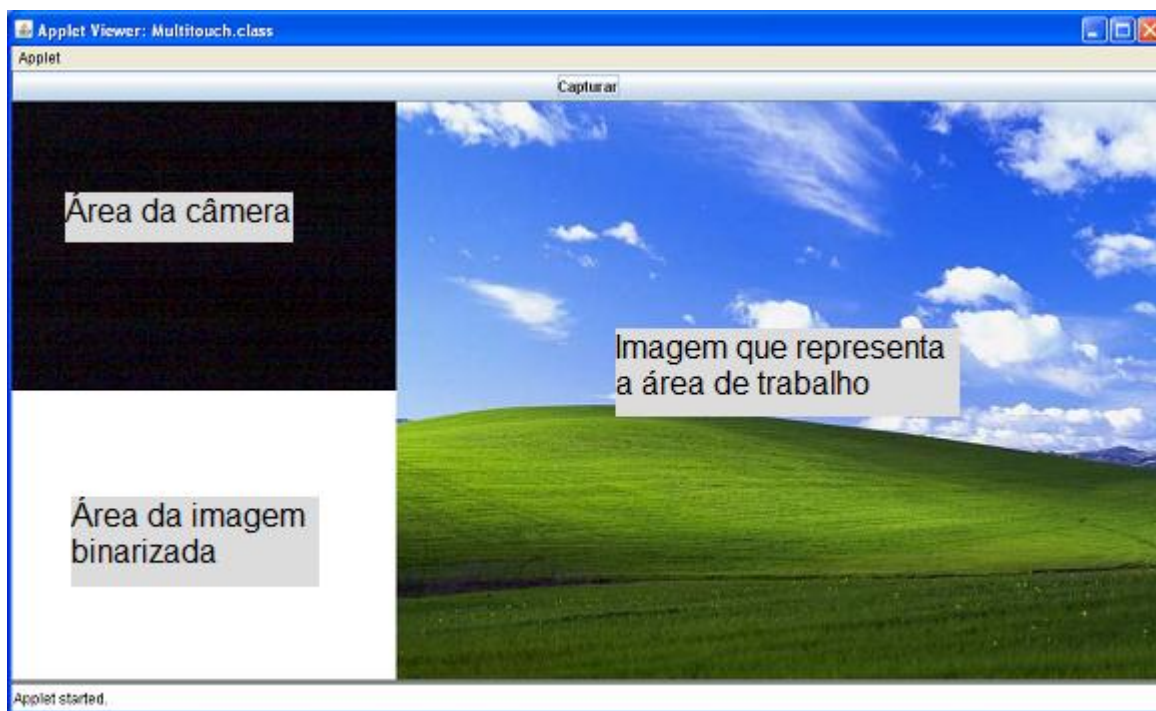


Figura 4.2.5 Aparência do Applet desenvolvido.

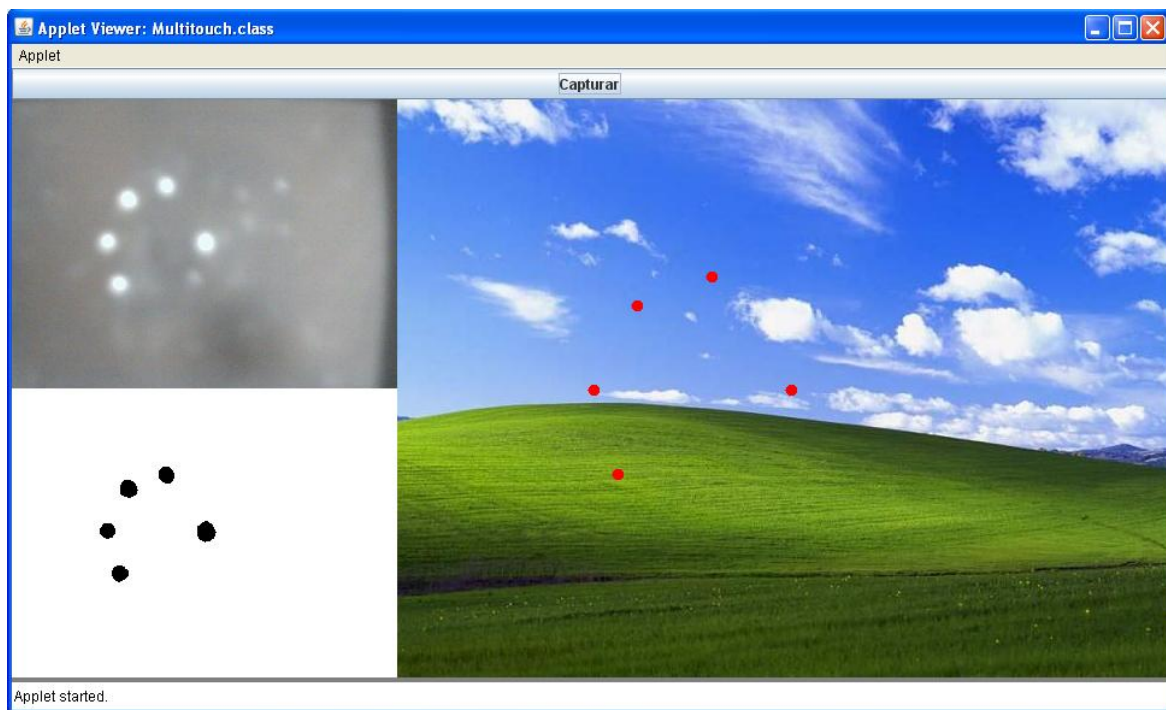


Figura 4.2.6 Aparência do Applet desenvolvido em funcionamento.

Capítulo 5

Testes

Todo o processo de desenvolvimento do *hardware* neste trabalho foi realizado de forma experimental. Durante este período de experimentação vários testes foram realizados em busca de resultados aceitáveis e são relatados neste capítulo.

Para o *software*, os testes foram realizados em duas fases. Na primeira foi testada, separadamente, cada etapa da programação descrita na seção 4.2, o que facilitou verificar se estavam funcionando corretamente e corrigir prováveis erros. Na segunda fase o teste foi realizado no software terminado, utilizando o *applet* construído para gerenciar as tarefas necessárias do sistema.

5.1 Testes de hardware

Na construção do *hardware* foi testada a utilização de vidro temperado como superfície de toque. Este teste foi importante para a tentativa de redução do custo do projeto. Porém, foi constatado que o vidro temperado tem propriedades que alteram a incidência da luz infravermelha no seu interior e prejudica a propriedade física FTIR, base para o projeto.

Quando na utilização do vidro, a reflexão interna total é dificultada por possuir uma dispersão maior de luz. Em substituição ao vidro foi utilizado acrílico que atendeu às propriedades desejadas.



Figura 5.1.1 Posicionamento dos leds e do acrílico.

A câmera utilizada no *hardware* foi modificada com filtros para passagem apenas da luz infravermelha e substituição da lente em busca de maior ângulo de captura. Foram testadas lentes utilizadas em câmeras de segurança para ambientes pequenos, que possuem um ângulo de abertura maior e distancia focal menor.

A câmera posiciona-se imediatamente abaixo e centrada em relação ao acrílico, posição ideal, porém a substituição da lente original buscando maior ângulo de abertura causou uma deformidade na imagem que, embora influencie no resultado da imagem capturada, será desconsiderada neste trabalho inicial.

Outros testes complementares no *hardware* dizem respeito ao funcionamento dos conectores externos, ligações do circuito de *leds* e partes elétricas internas.

5.2 Testes de software

O primeiro teste realizado no *software* foi relacionado a captura feita pela câmera. Com este teste foi decidido pelo uso do JMF API que possui funções para controle de mídias, facilitando a implementação. Também é possível testar a qualidade dos pontos identificados na imagem.

O resultado é visto na figura 5.2.1 com a imagem de um ponto, na figura 5.2.2 com dois pontos, na figura 5.2.3 com ponto na borda da interface e 5.2.4 com vários pontos de toque.



Figura 5.2.1 Imagem da câmera com um ponto de toque.



Figura 5.2.2 Imagem da câmera com dois pontos de toque.



Figura 5.2.3 Imagem da câmera com um ponto de toque na borda da interface.



Figura 5.2.4 Imagem da câmera com pontos de toque variados.

Ainda na figura 5.2.4 é possível verificar vários pontos de toque com formas diferentes e pressão de toque diferente.

O teste referente a segmentação pode ser observado na figura 5.2.5 com a segmentação de um ponto de toque e na figura 5.2.6 que contém múltiplos pontos de toque.

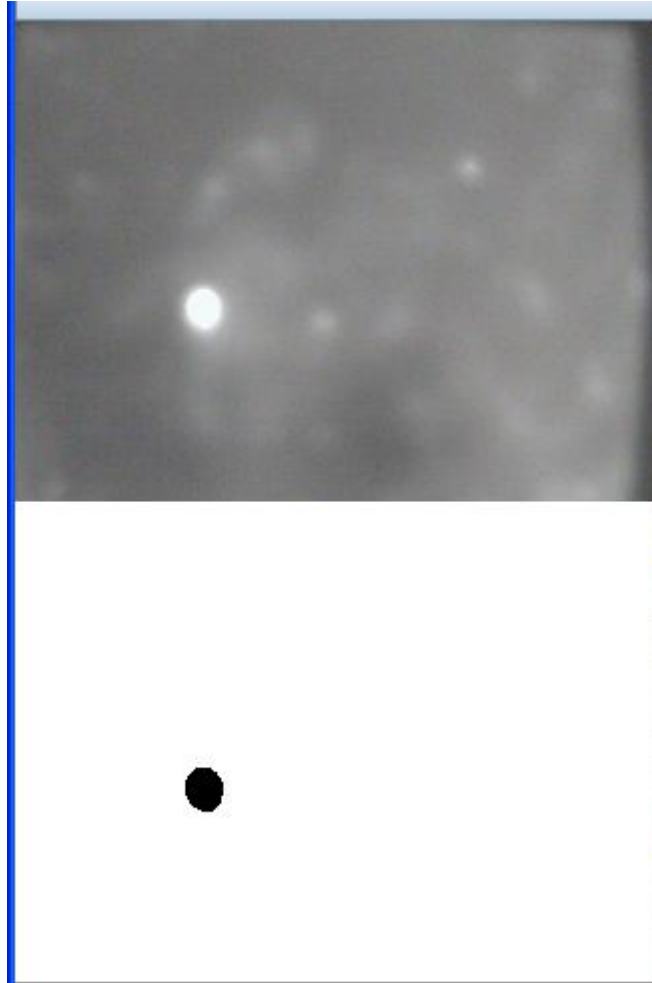


Figura 5.2.5 Imagem da câmera e segmentada com um ponto.

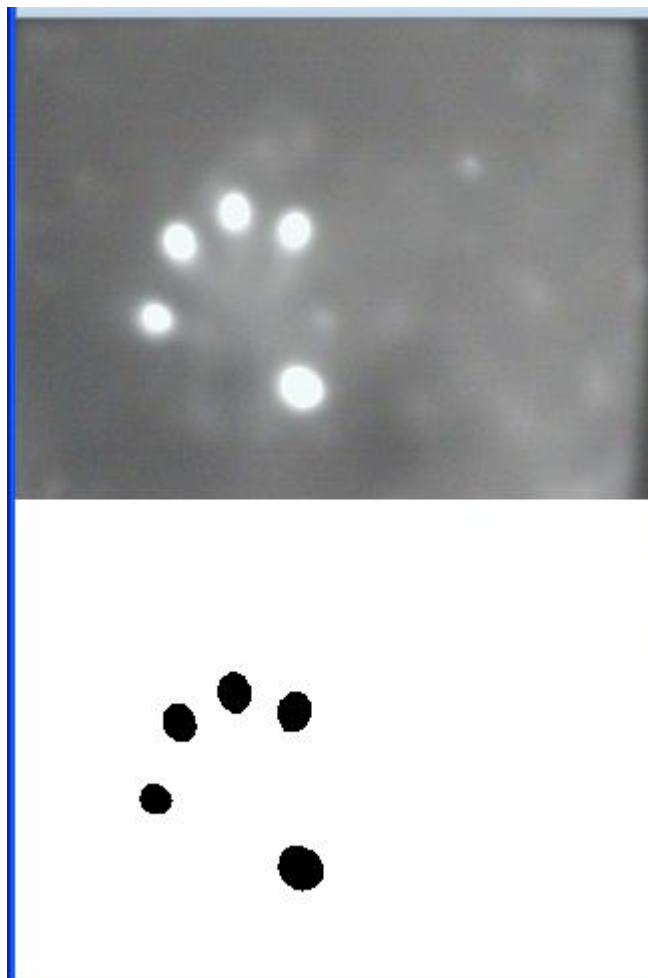


Figura 5.2.6 Imagem da câmera e segmentada com múltiplos pontos.

Durante os testes da segmentação foi possível detectar erros no algoritmo de crescimento de regiões, quando o toque ocorria próximo a borda da imagem. Foi corrigido incorporando ao algoritmo a verificação dos limites da imagem antes da expansão.

Nas figuras 5.2.7 e 5.2.8 é possível verificar o resultado final do *software* com a imagem da câmera, a imagem segmentada e os pontos identificados representados na imagem à direita. Também é possível ver o resultado impresso no console, abaixo do *Applet*, com a identificação das coordenadas centrais de cada ponto.

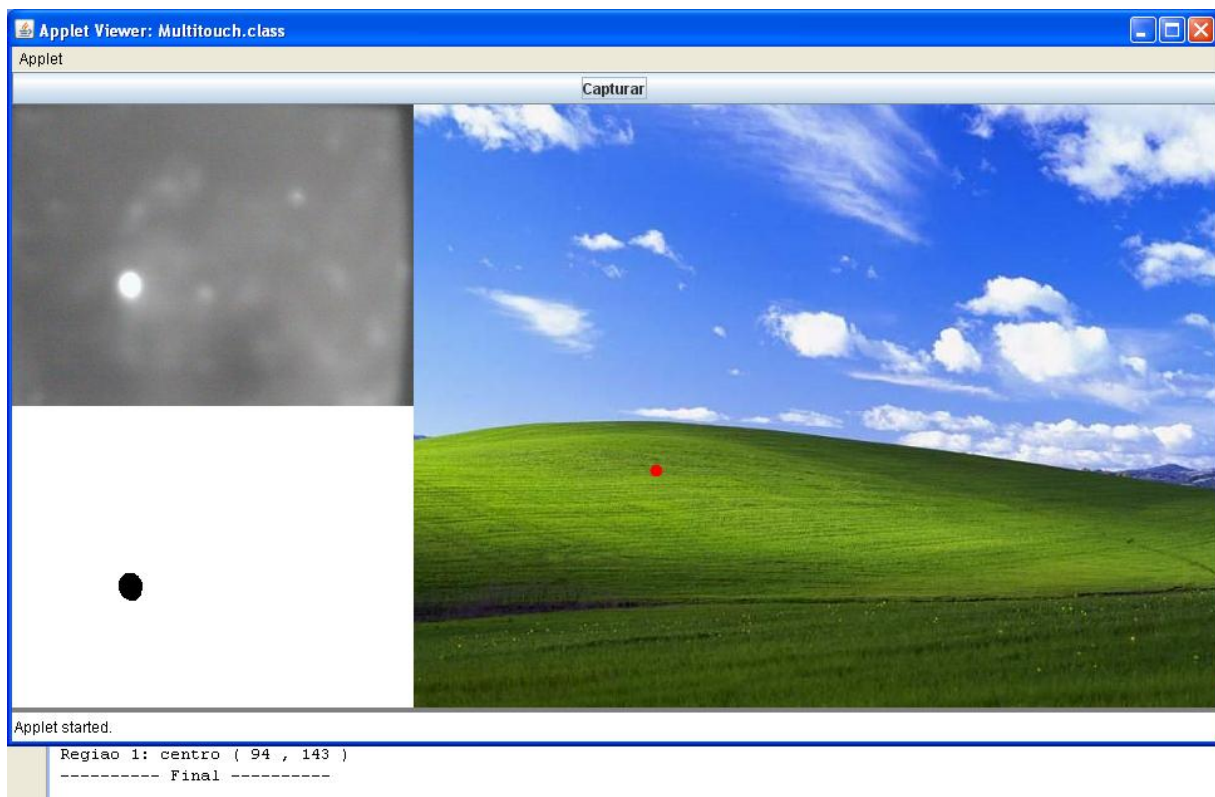


Figura 5.2.7 Software em funcionamento com um ponto.

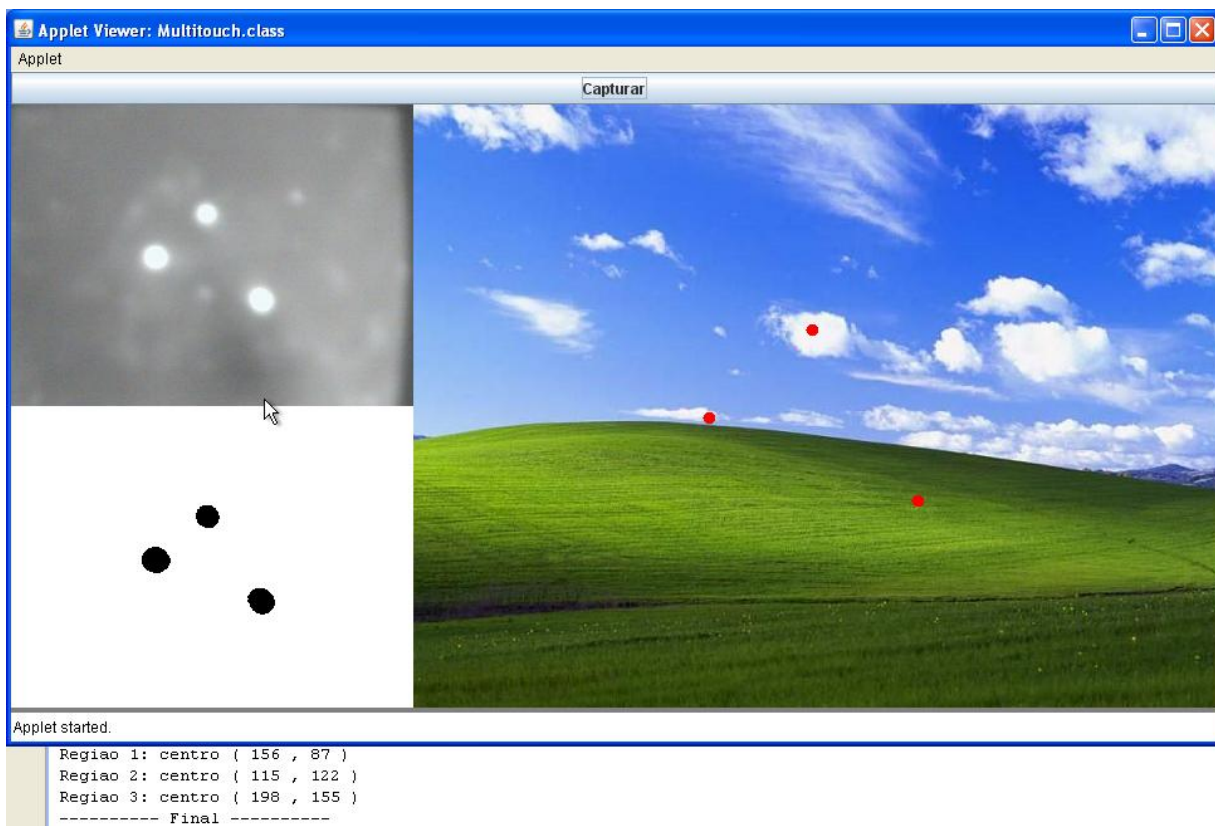


Figura 5.2.8 Software em funcionamento com três pontos.

5.3 Teste da Interface

Com o *software* e o *hardware* finalizados, fez-se o teste do funcionamento completo da interface produzida como segue na figura 5.3.1.

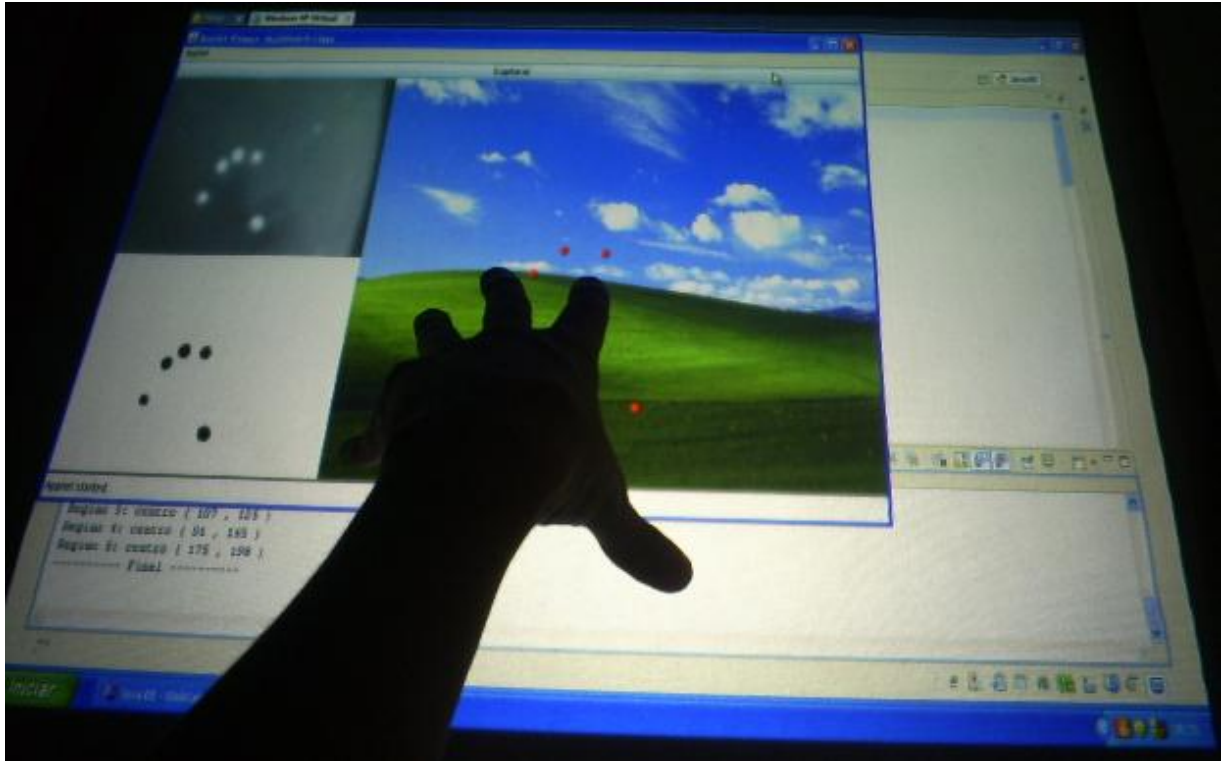


Figura 5.3.1 Interface produzida em funcionamento.

Capítulo 6

Conclusão

O objetivo deste trabalho é demonstrar o desenvolvimento de uma interface com a tecnologia de multitoques, que estão em evidencia hoje em dia. Como um trabalho inicial de pesquisa e desenvolvimento deste tipo de interface, foi proveitoso principalmente para a identificação e preenchimento de lacunas nas informações sobre a tecnologia.

Em produtos comerciais a divulgação das informações é restrita, por isso fez-se necessária uma pesquisa mais vasta, complementada com projetos experimentais. Grande parte das informações adquiridas foi através de sites públicos de discussão sobre tecnologia. Partindo de informações como dos projetos *Surface* e *Perceptive Pixel*, conseguiu-se confirmar e deduzir os passos principais para o desenvolvimento do *hardware* e do *software* para esse trabalho.

Sobre a tecnologia pode se afirmar, com base na pesquisa feita, que a expansão destas interfaces para a grande população acontecerá em breve. Interfaces de baixo custo e a produção em massa darão suporte para essa popularização.

Os pontos fortes identificados nestas interfaces são o fator intuitivo para o usuário, a redução de periféricos e a ampla área de aplicação.

Em trabalhos futuros prevê-se a utilização de técnicas para reduzir as distorções na imagem da câmera, propiciando a implementação relacionada ao tamanho do toque na tela.

Sabendo que neste trabalho o *software* foi finalizado na detecção do ponto, tem-se a intenção de implementar funções para controle do ponteiro do mouse, de navegação, identificação de objetos e conectividade com dispositivos móveis sobre a superfície.

Em nível de *hardware*, pretende-se reduzir o tamanho da interface, onde é preciso um reposicionamento ou substituição dos elementos internos como câmera e projetor. Um estudo para o uso de várias câmeras combinadas faz-se importante como opção.

Referências Bibliográficas

- [1] MOREIRA, D.; *Multi-touch: conheça as origens e aplicações do cobiçado recurso do iPhone*. 2007. Disponível em <http://idgnow.uol.com.br/computacao_pessoal/2007/06/28/idgnoticia.2007-06-28.4782885249>. Último acesso em novembro de 2009.
- [2] HAN, J.; New York University Courant Institute of Mathematical Sciences. Disponível em <<http://www.cs.nyu.edu/~jhan/>>. Último acesso em novembro de 2009.
- [3] Época Negócios – NOTÍCIAS. *O reinventor do nosso futuro*, 2007. Disponível em <<http://epocanegocios.globo.com/Revista/Epocanegocios/0,,EDG78392-8377-6,00.html>>. Último acesso em novembro de 2009.
- [4] CAMPOS, C. A. T.; *Comportamento das componentes de sistemas multi-toque baseados em reflexão interna total confinada*. Rio de Janeiro, 2009. 98p. Dissertação (Mestrado em Informática) – Departamento de Informática, Pontifícia Universidade Católica do Rio de Janeiro. Disponível em <http://www.maxwell.lambda.ele.puc-rio.br/Busca_etds.php?strSecao=resultado&nrSeq=14013@1>. Último acesso em novembro de 2009.
- [5] WILSON, T.; *Como funciona o iPhone*. Disponível em <<http://eletronicos.hsw.uol.com.br/iphone.htm>>. Último acesso em novembro de 2009.
- [6] Microsoft Corporation; Welcome to Microsoft Surface. Disponível em <<http://www.microsoft.com/surface/>>. Último acesso em novembro de 2009.
- [7] Perceptive Pixel, Inc.; Perceptive Pixel Founded by Jeff Han Developer of Advanced Multi-Touch Solution. Disponível em <<http://www.perceptivepixel.com/>>. Último acesso em novembro de 2009.
- [8] Mindsorm, Inc.; iBar – intelligent surface system – interactive bar. Disponível em <<http://www.i-bar.ch/en/info/>>. Último acesso em novembro de 2009.
- [9] NUI Group; Natural User Interface Group. Disponível em <<http://www.nuigroup.com/>>. Último acesso em novembro de 2009.

[10] Apple – Iphone . Disponível em <<http://www.apple.com/br/iphone>>. Último acesso em novembro de 2009.

[11] BUXTON, B.; *Multi-Touch Systems that I Have Known and Loved*, 2007. Disponível em <<http://www.billbuxton.com/multitouchOverview.html>>. Último acesso em novembro de 2009.

[12] DiamondSpace – Mitsubishi Electric Research Laboratories. Disponível em <<http://www.diamondspace.merl.com/>>. Último acesso em novembro de 2009.

[13] FACON, J.; *Processamento e Análise de Imagens* – Pontifícia Universidade Católica do Paraná Curso de Mestrado em Informática Aplicada, Fevereiro de 2002. Notas de Aula.

[14] WANGENHEIM, A.; *Introdução a Visão Computacional* – Disponível em <<http://www.inf.ufsc.br/~visao/>>. Último acesso em novembro de 2009.

[15] GONZALEZ, R.C.; Woods, R. E. *Digital Image Processing*. 3. ed. Upper Saddle River, NJ: Prentice Hall, 2002. p. 567-642.

[16] MARTINS, A. C. G.; *Segmentação Crescimento de Regiões* – Disponível em <<http://www.sorocaba.unesp.br/professor/amartins/aulas/pdi/aula11.pdf>>. Último acesso em novembro de 2009.

[17] MUNDIM, K. C.; *Aula 12 – O princípio de Fermat, A Reflexão, A Refração, Reflexão interna total, Aplicação* – Disponível em <<http://vsites.unb.br/iq/kleber/EaD/Fisica-4/Aulas/Aula-12/aula-12.html>>. Último acesso em novembro de 2009.

[18] FREUDENRICH, C.; *Como funcionam as fibras óticas* – Disponível em <<http://informatica.hsw.uol.com.br/fibras-opticas.htm>>. Último acesso em novembro de 2009.

[19] VIANA, R. L.; *Aula 30 – Reflexão e Refração da Luz* – Disponível em <<http://fisica.ufpr.br/viana/fisicab/aulas2/aula30.html>>. Último acesso em novembro de 2009.

[20] Sun Microsystems; Java SE Desktop Technologies – Java Media Framework API (JMF). Disponível em <<http://java.sun.com/javase/technologies/desktop/media/jmf/>>. Último acesso em novembro de 2009.

[21] Microsoft Corporation; Home Page do Windows 7. Disponível em <<http://windows.microsoft.com/pt-PT/windows7/products/home?os=winvista>>. Último acesso em novembro de 2009.

[22] PENENBERG, A. L.; Revista Galileu – *Toque de Gênio*. Disponível em <<http://revistagalileu.globo.com/Revista/Galileu/0,,EDR77829-8489,00.html>>. Último acesso em novembro de 2009.

[23] Museu das Comunicações – *Reflexao Interna Total*. Disponível em <http://macao.communications.museum/por/exhibition/secondfloor/MoreInfo/2_8_4_TotalInternalReflection.html>. Último acesso em novembro de 2009.

[24] AVELINO, D.; *Multi-touch*. Disponível em <<http://www.unifran.br/blog/cienciaDaComputacao/?action=d3d31446a793743680c65030e6a43434d3d366b30a819d87&post=d3d315e48715a6d4af9c51db31bcb7201bb3a7717e400663&area=a77644e4b8d189c7efceee6ccc3d8bbb6f7e5268>>. Último acesso em novembro de 2009.

[25] SCHÖNING, J. et al; *Multi-Touch Surfaces: A Technical Guide*. Disponível em <http://ifgiweb.uni-muenster.de/~j_scho09/pubs/bymultitouch.pdf>. Último acesso em novembro de 2009.