

6 Aplicações

Neste trabalho, é importante implementar aplicações de interação em tempo real para que seja possível avaliar a aplicabilidade das técnicas de Visão Computacional descritas ao longo dos capítulos anteriores na interação homem-máquina. Neste capítulo são apresentadas três aplicações de interação baseadas em gestos da mão. Na primeira aplicação, é desenvolvido um mecanismo de interação em que a posição do dedo índice e os gestos reconhecidos são utilizados para movimentar o cursor e simular funções básicas do mouse. Na segunda aplicação, a posição e a direção dos dedos são utilizadas para definir direções de movimento, simulando assim as teclas de direção do teclado. Juntamente com alguns gestos reconhecidos, essas direções são aplicadas ao controle de algumas das principais funções de um *vídeo game*. Na terceira aplicação o foco principal é a interação puramente gestual, ao contrário das duas anteriores, em que são requeridas posições e direções dos dedos. Nesta aplicação são reconhecidas duas mudanças específicas nos gestos, as quais são aplicadas na movimentação de slides de apresentação.

6.1. Detalhes da Implementação

Aplicação, neste capítulo, significa o processo de mapear os gestos reconhecidos para o controle e simulação de algumas funções básicas em programas já existentes no computador. Para poder controlar funções básicas nesses programas é necessário um mecanismo que permita simular os controladores (*drivers*) de entrada, tanto o do mouse como o do teclado. Neste capítulo, todas as aplicações foram implementadas no sistema Windows, porque a plataforma SDK possui funções que permitem gerar eventos do mouse e do teclado da mesma forma que são gerados nos seus controladores convencionais. Os eventos gerados são automaticamente enviados para as janelas ativas e processados por qualquer programa do sistema.

6.2. Controlando o Cursor do Mouse

Nesta aplicação, os gestos, seus parâmetros e algumas combinações destes são aplicados no controle do cursor de um mouse convencional.

Para simular o cursor do mouse é necessária a realização de duas tarefas importantes. A primeira é movimentar o cursor e a segunda é simular os eventos básicos utilizando os gestos reconhecidos. Portanto, nesta aplicação, numa primeira parte é definido um mecanismo que, utilizando a posição do dedo indicador, define a direção do movimento e a posição do cursor ao longo do tempo. Na segunda parte, é explicado que gestos e quais combinações destes foram utilizados para simular os eventos básicos do mouse. Os eventos a simular serão MOVIMENTAR, CLICK, CLICK_DUPLO, CLICK_DIREITO e ARRASTAR.

Existem diferentes formas de mapear as coordenadas da posição do dedo detectado na imagem para as coordenadas da tela onde o cursor é exibido. Como a resolução da imagem capturada pela câmera na maioria dos casos é menor do que a resolução da tela, mecanismos de mapeamento direto, da imagem para a tela, têm problemas de escala. Um pixel na imagem muitas vezes representa um número maior de pixels na tela. Para evitar esse problema, nesta aplicação optamos por um mecanismo de mapeamento relativo. Nesse mecanismo, é necessário definir um ponto de referência na imagem e, depois, levando em conta a posição atual do dedo em relação a esse ponto de referência, é calculada a direção do movimento e a posição final do cursor.

Na Figura 32 é esquematizado o mecanismo utilizado para movimentar o cursor. Os pontos **p1**, **p2** e **p3** representam três posições diferentes (em quadros diferentes) do dedo índice. Esses pontos, em relação o ponto de referência 'O', formam três vetores, os quais indicam a posição do dedo em relação ao ponto de referência. As coordenadas (x, y) dos vetores são calculadas levando em conta o ponto de referência 'O' como sendo a origem. A cada quadro, a direção do movimento no cursor é dada pelas coordenadas (x, y) do vetor e a velocidade (ou salto) do movimento é dado pela norma do vetor.

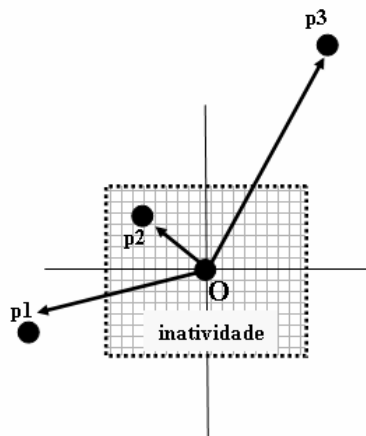


Figura 33 - Mecanismo de movimento no cursor. Para cada ponto **p1**, **p2** e **p3**, a direção do movimento é dada pelas coordenadas **(x,y)** do vetor e a velocidade do movimento é dada pela norma do vetor. Em torno do ponto de referência **O** existe uma área de inatividade na qual os pontos que caem nessa área (**p2**) não ocasionam nenhum movimento.

Dada essa configuração, a única forma de manter o cursor estático seria com vetores cujas normas fossem zero. Isto na prática é difícil de se conseguir, pois implicaria manter o dedo totalmente parado no ponto **O**. Além disso, na etapa de detecção dos dedos sempre existem, entre os quadros, pequenas variações na posição detectada. Considerando este fato, em volta do ponto **O** foi definida uma área de inatividade onde os pontos que caem nessa área (p. ex., o ponto **p2**) não ocasionam movimento no cursor. Os únicos gestos que geram o evento MOVIMENTAR são o INDICE_2 e o INDICE_1 (estes gestos estão ilustrados na Figura 30). No caso do gesto INDICE_2, ele somente movimenta o cursor e não executa nenhum evento. No caso do gesto INDICE_1, o movimento é feito combinado ao evento ARASTAR. A Figura 33 mostra o gesto INDICE_2 indicando uma direção específica de movimento para o cursor. Isto significa que, a partir da sua última posição, o cursor se movimenta na direção indicada pelo vetor e com saltos do tamanho da sua norma.

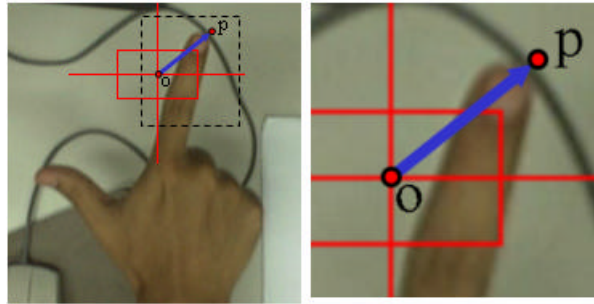


Figura 34 - Mecanismo de movimento do cursor. O vetor formado entre a posição do dedo índice e o ponto de referência indica a direção do movimento. A cada quadro este vetor é somado à última posição do cursor.

Para simular o evento CLICK_DIREITO foi utilizado o gesto ESTENDIDO (mostrado na Figura 30). Assim, cada vez que este gesto é apresentado, o cursor do mouse se mantém na sua última posição e invoca o evento CLICK_DIREITO do mouse.

Para simular o evento CLICK foi utilizada uma seqüência dos gestos INDICE_2 e INDICE_1. Cada vez que existe uma seqüência de mudança do gesto INDICE_2 para o gesto INDICE_1 e deste novamente para INDICE_2, essa seqüência é considerada como sendo um evento CLICK. Este depende do tempo e, para ser validado, a seqüência dos gestos deve ser feita dentro do intervalo de tempo T predefinido, como mostra a Figura 34. A partir do gesto corrente (atual) é analisada a seqüência de gestos anteriores. Se, dentro do intervalo de tempo definido existe o padrão de mudança procurado, então o evento CLICK é validado.

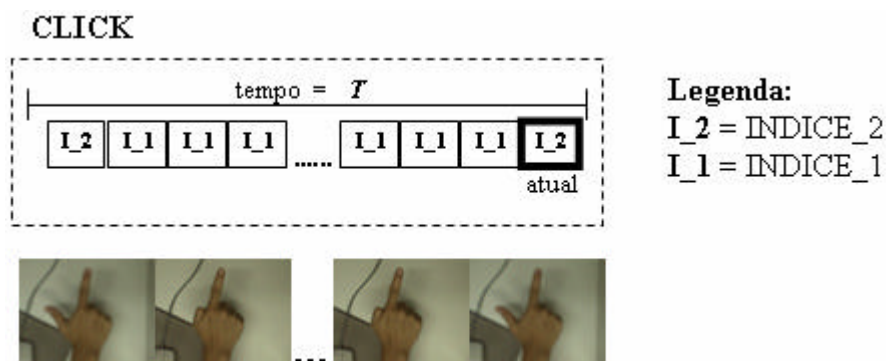


Figura 35 - Padrão de mudança do evento CLICK. Na nossa aplicação, o intervalo de tempo foi definido em número de quadros, $T=20$. Entre os gestos INDICE_2 pode haver vários gestos INDICE_1. A partir do gesto corrente (**atual**) é analisada a seqüência em busca do padrão de mudança requerido.

Para simular o evento **ARRASTAR** foi utilizado o gesto **INDICE_1**. O evento acontece enquanto o gesto **INDICE_1** é mantido. Para validar o evento **ARRASTAR**, o gesto **INDICE_1** deve ser mantido além de um determinado tempo **t** (Figura 35). Este tempo é importante para poder diferenciar se o gesto **INDICE_1** mostrado indica um evento **ARRASTAR** ou simplesmente faz parte da seqüência do evento **CLICK**.

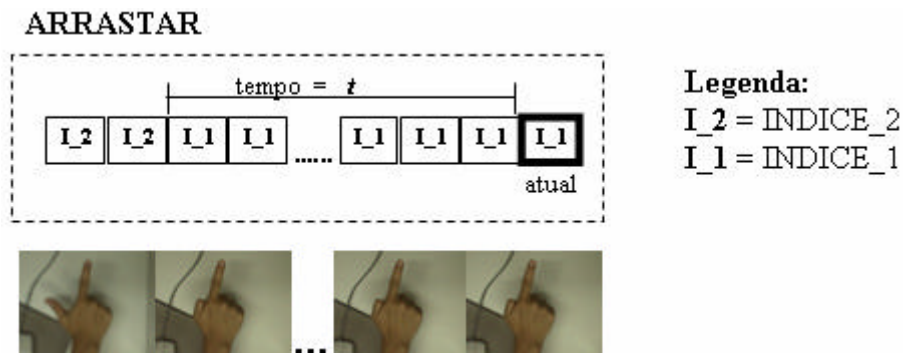


Figura 36 - Evento **ARRASTAR**. O evento acontece enquanto o gesto **INDICE_1** é mantido. Para validar o evento, o intervalo de tempo em que o gesto **INDICE_1** é mantido deve ser maior do que o **t** predefinido. Na nossa aplicação o tempo foi **t = T-5**. O tempo **t** está relacionado com o tempo **T**.

O evento **CLICK_DUPLO** é simplesmente uma seqüência de dois eventos **CLICK** consecutivos. Na Tabela 3 são mostrados os resultados de um teste de reconhecimento feito para os eventos **CLICK** e **CLICK_DUPLO**.

Gesto	Total de Tentativas	Total de Acertos	Total de Erros	Porcentagem de Acerto	Porcentagem de Erro
CLICK	300	295	5	98.4 %	1.6 %
CLICK_D	200	185	15	92.5 %	7.5 %

Tabela 3 - Resultado do teste de reconhecimento dos eventos **CLICK** e **CLICK_DUPLO**.

Na Figura 36 é mostrado o diagrama de fluxo do controle do cursor. A entrada é sempre um gesto reconhecido e os eventos são gerados dependendo do gesto que no momento está sendo processado. O evento **CLICK** é validado para diferenciar o evento **CLICK** do evento **MOVIMENTAR**. Também o evento **ARRASTAR** é validado para diferenciar se o gesto **INDICE_1** é apenas parte da seqüência do evento **CLICK**.

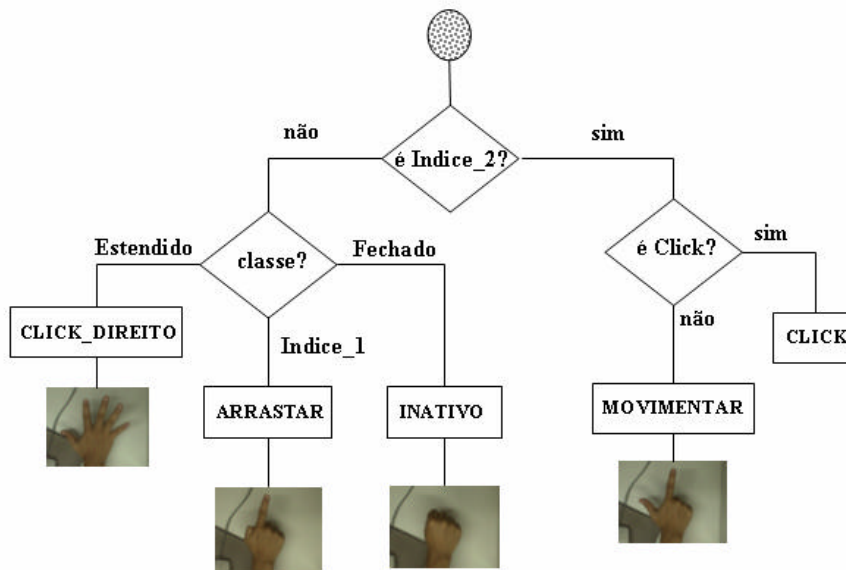


Figura 37 - Diagrama de fluxo do controle do cursor do mouse. Os eventos são validados dependendo do gesto que está sendo processado. O evento CLICK é validado para diferenciar o CLICK do evento MOVIMENTAR. O evento ARRASTAR é validado para diferenciar se o gesto INDICE_1 indica o evento ARRASTAR o simplesmente faz parte da seqüência do CLICK.

A seguir são apresentadas algumas imagens que mostram a implementação do controle do cursor feita para controlar o mouse do sistema Windows. Estas imagens mostram os gestos sendo mapeados em eventos do mouse, o que permite interagir com vários programas suscetíveis a esses eventos.

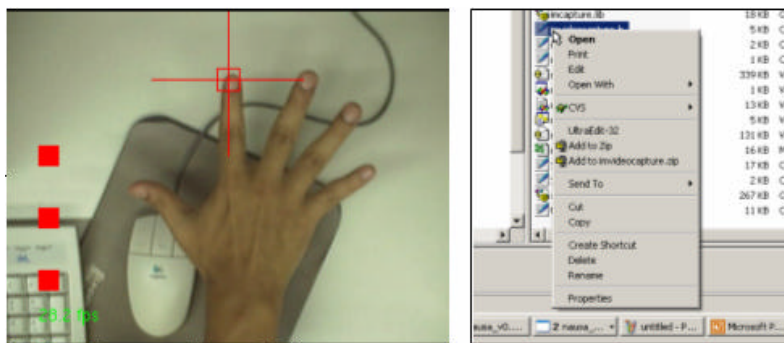


Figura 38 - Gesto ESTENDIDO sendo mapeado para o evento CLICK_DIREITO do mouse no sistema Windows. Ao gerar o evento no arquivo selecionado, é mostrado o menu disponível no arquivo.

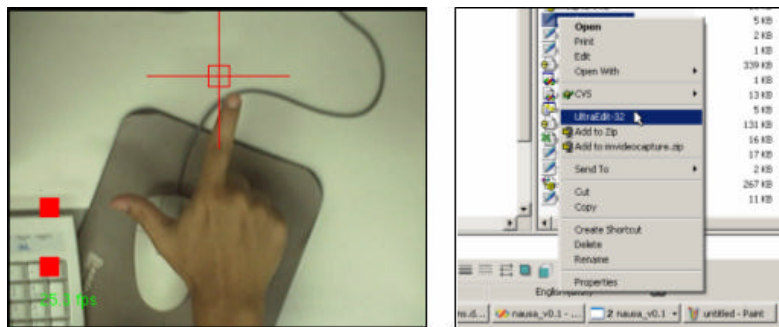


Figura 39 - Gestos INDICE_2 sendo utilizados para movimentar o ponteiro do mouse. Utilizando o gesto INDICE_2 podem-se selecionar itens do menu.

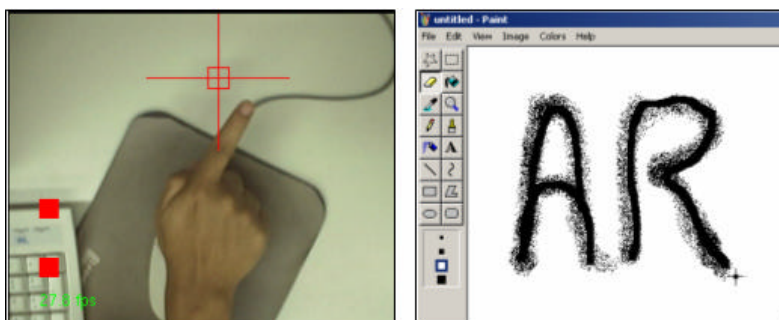


Figura 40 - Gestos INDICE_1 sendo mapeado para o evento ARRASTAR, o qual é utilizado para desenhar letras no PAINT.

6.3. Controlando um Jogo de Computador

Nesta aplicação o objetivo principal é simular algumas das principais funções de um *vídeo game* (AirStrike II). Nesta aplicação também há duas tarefas a cumprir no que se refere ao movimento e aos eventos que devem ser gerados. A posição e a direção do dedo índice serão utilizadas para definir quatro possíveis direções de movimento. Tais direções serão: ESQUERDA, DIREITA, PARA_FRENTE, PARA_TRAS. Com respeito aos eventos que devem ser gerados, os gestos reconhecidos serão mapeados para simular os eventos de ATIRAR, MUDAR_ARMA e SOLTAR_EXPLOSIVO.

O mecanismo utilizado para calcular a direção do movimento é similar ao empregado na aplicação anterior. As direções PARA_FRENTE e PARA_TRAS são definidas pela posição do dedo índice com respeito a um ponto de referência. As direções ESQUERDA e DIREITA são definidas pelo valor do ângulo da direção do dedo com relação à horizontal.

Na Figura 40(a) são apresentadas três possíveis posições **p1**, **p2** e **p3** do dedo índice, um ponto de referência **O** e uma área de inatividade que se estende

ao longo do eixo horizontal, em volta do ponto de referência. Se a posição (x, y) do dedo está fora da área de inatividade, quer dizer que existe movimento. Dependendo de se a coordenada y da posição do dedo é maior (ex. **p3**) ou menor (ex. **p1**) do que a coordenada y do ponto de referência, inferem-se as direções PARA_FRENTE e PARA_TRAS, respectivamente. No caso do ponto **p2** não se infere nenhum movimento.

O valor da direção do dedo índice é dado por um ângulo aproximado que o dedo faz com a horizontal (cf. Capítulo 5). Os vetores (**p1**, **p2** e **p3**) apresentados na Figura 40(b) representam diferentes direções do dedo índice. Se o valor do ângulo é menor (ex. **p3**) do que um valor a (ângulo limite da área de inatividade), a direção do movimento inferida é a DIREITA. Se o ângulo for maior (ex. **p1**) do que $-a$, a direção de movimento inferida é a ESQUERDA. Para as direções próximas ao eixo vertical (ex. **p2**) não existe movimento.

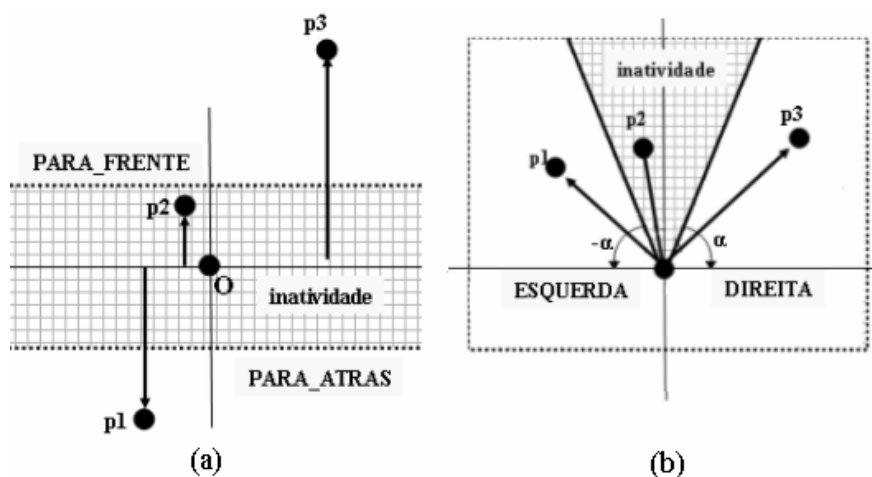


Figura 41 - Mecanismo utilizado para calcular a direção do movimento. (a) Mecanismo utilizado para calcular as direções PARA_FRENTE e PARA_TRAS. (b) Mecanismo utilizado para calcular as direções DIREITA e ESQUERDA. O valor utilizado na nossa implementação foi $a = 75^\circ$.

No que se refere aos eventos que devem ser gerados, foram empregados alguns gestos e uma combinação destes. Para simular o evento ATIRAR, foi utilizado o gesto INDICE_1. O evento ATIRAR é mantido enquanto o gesto INDICE_1 também é mantido. Este gesto é utilizado da mesma maneira como foi usado no evento ARASTAR da aplicação anterior (veja as Figuras 41(a) e 42(a)).

Existe a possibilidade de se selecionar a arma com a qual se quer atirar. Para simular o evento MUDAR_ARMA foi utilizada uma seqüência específica

dos gestos INDICE_2 e INDICE_1. Neste evento foram empregados os gestos da mesma maneira que no evento CLICK da aplicação anterior. Para validar o evento deve existir uma seqüência de gestos que começa em INDICE_2, muda para INDICE_1 e novamente muda para INDICE_2.

Além da arma convencional, existe a possibilidade de se soltar explosivos que são ganhos no transcurso do jogo. Para simular o evento SOLTAR_EXPLOSIVO foi utilizado o gesto ESTENDIDO (Figura 42(b)).

A seguir são apresentadas algumas imagens que mostram cenas do jogo controlado com os gestos reconhecidos. Nas imagens são mostrados os eventos gerados de acordo com a direção e a posição do dedo índice e os gestos reconhecidos.

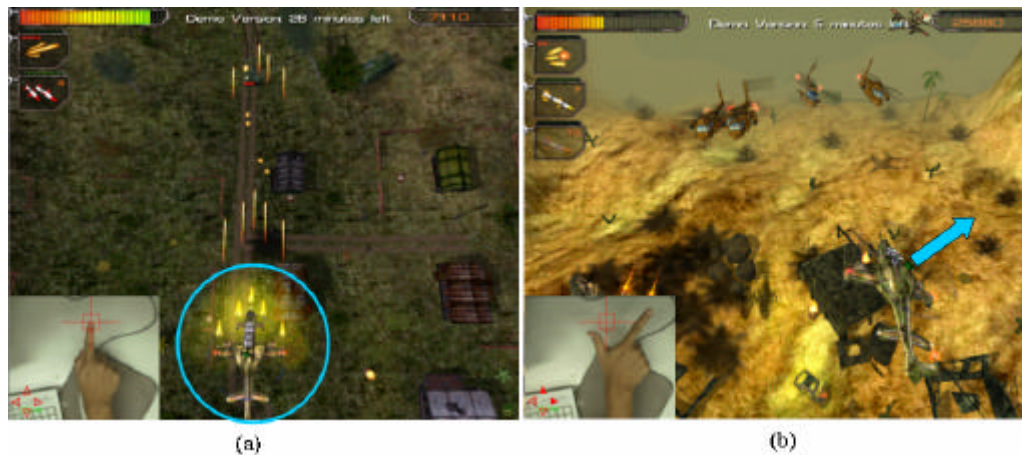


Figura 42 - (a) Geração do evento ATIRAR utilizando o gesto INDICE_1. A posição e a direção do dedo índice não indicam movimento. (b) Movimentação do helicóptero utilizando o gesto INDICE_2. A posição e a direção de dedo indicam os eventos PARA_FRENTE e PARA_DIREITA . Neste caso não é gerado o evento ATIRAR.



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.

6.4. Interação Gestual

Ao contrário das duas aplicações anteriores, nas quais foram consideradas as informações referentes ao dedo índice, nesta aplicação serão consideradas apenas informações gestuais. Nesta aplicação são reconhecidas duas mudanças específicas nos gestos para gerar dois eventos do teclado, que, neste caso, são aplicados ao movimento de slides de apresentação (PowerPoint). Os eventos que são reconhecidos, analisando-se a mudança nos gestos, são AVANÇAR e RECUAR. Cada vez que esses eventos são validados, são gerados eventos do teclado que, na configuração padrão, controlam o avanço e recuo nos slides.

O evento AVANÇAR é gerado cada vez que existe uma mudança do gesto FECHADO para o gesto ESTENDIDO. Este evento valida a mudança nos gestos, isto é, se num determinado momento estamos com a mão fechada e logo abrimos a mão mostrando os cinco dedos, como o indica o gesto ESTENDIDO, é gerado o evento AVANÇAR.

O evento RECUAR é gerado de maneira similar ao evento AVANÇAR. O ponto de partida é também o gesto FECHADO, com a diferença de que a mudança deve ser feita para o gesto INDICE_1 ou INDICE_2. Se num determinado momento estamos com a mão fechada e logo abrimos a mão mostrando o gesto INDICE_1 ou o gesto INDICE_2, o evento RECUAR é gerado.

Existem outras mudanças nos gestos que podem gerar outros eventos, mas foram somente duas as utilizadas. Estas seqüências de gestos têm a particularidade de terem o mesmo ponto de partida (o gesto FECHADO). Na Figura 43 são mostrados os diferentes estados da aplicação como sendo um autômato finito. O autômato é composto por quatro estados; dois estados não terminais **q0** e **q1** e dois estados terminais **q2** e **q3**, os quais geram os eventos AVANÇAR e RECUAR, respectivamente. As transições entre os estados representam os gestos que são mostrados ao longo do tempo. O estado inicial no autômato é o estado **q0**.

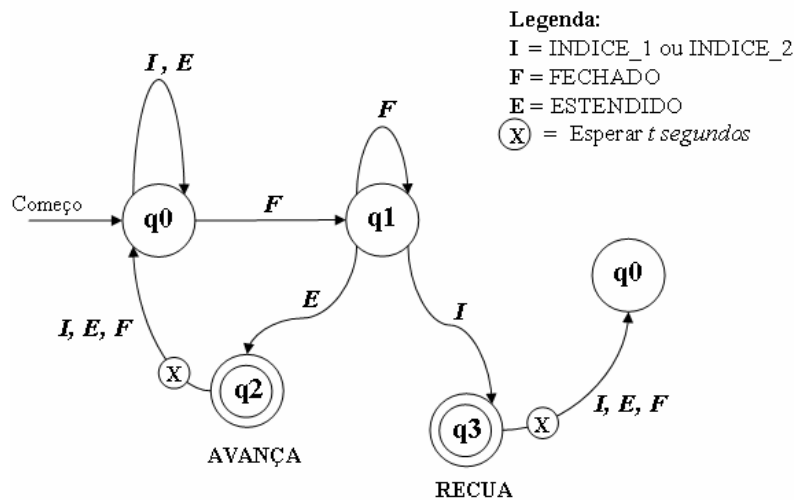


Figura 44 - Estados da aplicação representados como sendo um autômato finito de quatro estados. Os estados **q2** e **q3** geram os eventos AVANÇAR e RECUAR, respectivamente.

A seqüência que gera qualquer um dos dois eventos passa pelo estado **q1**. Dependendo do gesto da transição pode-se se chegar aos estados **q2** ou **q3**, gerando os respectivos eventos. Isto quer dizer que qualquer um dos dois caminhos que geram os eventos passa pela mão fechada. Depois que os eventos são gerados, volta-se ao estado inicial no qual novamente são analisados os gestos subseqüentes. Antes de se voltar ao estado inicial, existe a possibilidade de se definir um retardo na transição, esperando-se assim que o evento do teclado seja gerado pelo sistema antes de voltar novamente ao estado inicial.

A seguir são apresentadas duas imagens que mostram a geração dos eventos AVANÇAR e RECUAR na nossa implementação para mexer slides no PowerPoint.

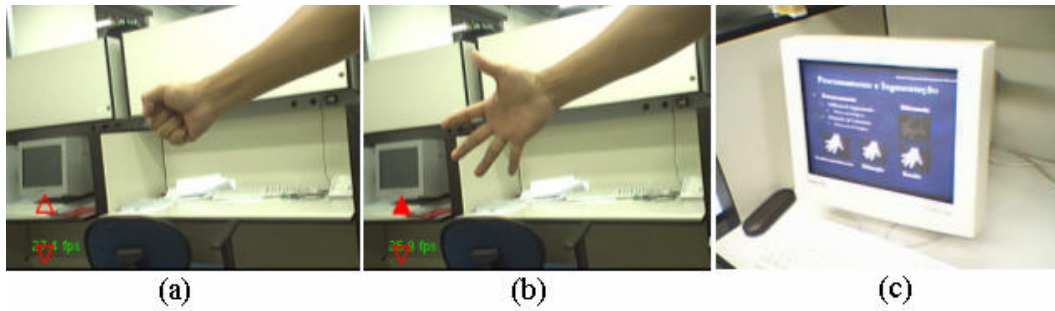


Figura 45 - Geração do evento AVANÇAR. A mudança do gesto (a) FECHADO para o gesto (b) ESTENDIDO gera o evento AVANÇAR, que é refletido na (c) tela do monitor.

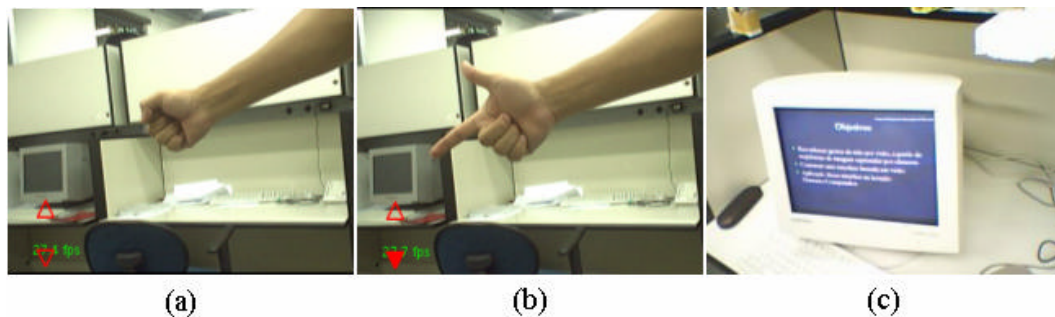


Figura 46 - Geração do evento RECUAR. A mudança do gesto (a) FECHADO para o gesto (b) INIDICE_2 gera o evento RECUAR, que é refletido na (c) tela do monitor.

6.5. Conclusões Parciais

Na implementação das três aplicações foram cumpridos todos os requisitos funcionais e não funcionais descritos no capítulo 2. Os dedos e os gestos foram adequadamente detectados, identificados e rastreados em tempo real, permitindo nos três casos interações maiores aos 20 quadros por segundo (20 Hz).¹

No que diz respeito ao ambiente e às condições de iluminação, para estas aplicações não foi feita nenhuma mudança que favoreça o seu desempenho. Em nenhuma das aplicações foram incluídas luzes ou ambientes especiais para a interação. Podemos dizer que houve uma adaptação ao ambiente e não uma adaptação do ambiente para o funcionamento das aplicações. Tal adaptação depende do ajuste dos parâmetros da subtração de fundo em cada ambiente.

¹ Alguns vídeos que mostram a interação nas três aplicações podem ser vistos em www.tecgraf.puc-rio.br/~alain/current/

A influência do ambiente e das condições de iluminação é maior quanto maior for a área do ambiente capturada pela câmera. No caso da aplicação feita para movimentar os slides, a influência do ambiente foi maior, gerando ruídos e deformando a morfologia da mão capturada. Apesar disso, a detecção dos dedos e o reconhecimento dos gestos foram cumpridos satisfatoriamente.

Para poder interagir adequadamente com as aplicações utilizando gestos, principalmente no caso de usuários sem experiência, é preciso um pequeno tempo de adaptação. Os principais problemas são referentes a algumas posições, ângulos ou a extensão dos dedos, os quais, às vezes, impedem que os gestos e os dedos sejam detectados. No geral, o conjunto de gestos é pequeno e fácil de ser lembrado. A interação mais simples se dá na aplicação de movimentar os slides.

Durante a interação e principalmente nas duas primeiras aplicações, alguns erros na detecção dos gestos ocasionam eventos não esperados. Esses erros acontecem devido a mudanças significativas no ambiente, as quais acontecem muito esporadicamente.

Se considerarmos a idéia de utilizar gestos da mão para realizar algumas tarefas comuns nos computadores pessoais, estas aplicações, mesmo com os bons resultados obtidos, ainda estão longe de poder substituir os atuais dispositivos de interação na medida que os usuários esperam ou com que estão acostumados.