

Aumentando a Interatividade de uma Apresentação com o Apontador-Mouse

Gabriel A. Rocha, Guilherme Morara, Vilson Müller Jr., Miguel A. Sovierzoski

Departamento de Eletrônica – Universidade Tecnológica Federal do Paraná (UTFPR)
Av. Sete de Setembro, 3165 – Curitiba – PR – Brazil

miguelaso@utfpr.edu.br

***Abstract.** The mouse-pointer was developed considering the environment of computer presentations using projector and software applications. The objective is to provide a mean for more interactivity with projections, permitting the presenter move and to keep the visual contact with the audience. The mouse-pointer emulates the function as a computer mouse. Capturing the projected image and through image processing techniques it is possible to estimate the position where the laser beam touches the projection screen, emulating the mouse cursor movement. The project resulted a slight resolution moving the mouse cursor due to the webcam used that had a 320x240 resolution, but it was totally functional, reaching the project objectives.*

***Resumo.** No contexto de apresentações com computador, projetor e aplicativos de software, foi desenvolvido o projeto apontador-mouse. O objetivo é proporcionar recursos para uma maior interação com a apresentação, resultando na liberdade de movimentação do apresentador e mantendo o contato visual com a platéia. O apontador-mouse emula o funcionamento do mouse do computador. Através da captura de imagem da tela de projeção e de técnicas de processamento de imagem é estimada a posição do feixe laser na tela de projeção, emulando o movimento do cursor do mouse. O projeto apresentou uma baixa resolução na movimentação do mouse, devido ao uso de uma webcam com resolução 320x240, mas demonstrou ser totalmente funcional, atendendo os objetivos do projeto.*

1. Introdução

O avanço tecnológico da eletrônica e da computação, e o barateamento destas tecnologias, devido a escala de consumo, possibilitou que a grande maioria das escolas públicas e privadas do Brasil fossem equipadas com computadores e projetores multimídia, segundo dados de pesquisa da Fundação Victor Civita (2009). A disponibilidade destes recursos em sala de aula gerou um novo paradoxo pedagógico, rompendo o limite secular da aula expositiva imposta pelo giz e quadro negro, tornando

a aula visualmente mais dinâmica e colorida. Não ocorreu uma simples substituição tecnológica trocando o retroprojetor e a transparência pelo computador com um aplicativo de apresentação e um projetor.

As apresentações e demonstrações interativas em sala de aula possuem valor didático-pedagógico elevando a qualidade da aula e melhorando o processo ensino-aprendizagem. Mas a interatividade destas ferramentas estava limitada as interfaces periféricas do computador, destacando-se o teclado e o *mouse*. Com o tempo, o computador recebeu outros periféricos auxiliares, como o controle remoto para o avanço e retrocesso de telas nos aplicativos de apresentação, destacando o *Power Point*® do *Office*® da Microsoft® e o *Impress* da *br office*, no ambiente Windows®.

A seção 2 descreve produtos disponíveis como soluções de mercado. A seção 3 apresenta o cenário de interatividade restrita com o computador, o projetor e o usuário. Na seção 4 é apresentado o cenário de interatividade com o uso do apontador-mouse, o computador e o projetor. A seção 5 detalha o projeto do apontador-mouse. A seção 6 apresenta resultados, e os comentários finais são apresentados na seção 7.

2. Soluções existentes

Diversas soluções para aumentar a interatividade de apresentações e demonstrações encontram-se no mercado na forma de produtos tecnológicos.

O *Black Screen Wireless Presenter* da empresa *Vesine* possui o apontador laser e permite o avanço e o retrocesso de telas em apresentações. O *Multimedia Presentation Remote AMP09US* [Targus] permite controlar o cursor do *mouse* através de um *joystick* e o *Nobo P3 Page Point & Present* [Infinitier] permite controlar o cursor do *mouse* através de um *track-ball*. O dispositivo mais comumente utilizado para interagir com apresentações é o *mouse* sem fio. Como exemplos cita-se o *Mouse óptico Microsoft 8000 Wireless Presenter* e o *Interlink VP6700 Wireless Presenter Mouse*. Outro tipo de dispositivo é a mesa digitalizadora que possui uma tela sensível ao toque, contudo, não apresenta a imagem. Esse tipo de dispositivo é mais comumente usado por profissionais de edição de imagens, porém não há restrição de seu uso em apresentações. Como exemplo cita-se a *Mesa Digitalizadora Bamboo Pen CTL460* da *Wacom*. O *SMART Board 800* [Smarttech] é um quadro branco interativo desenvolvido para ambiente educacional multi-usuário, visando melhorar a qualidade da didática expositiva dentro da sala de aula. Através de um projetor adaptado, de modo a eliminar ao máximo as sombras e reflexos, o equipamento possibilita a interação por toque (*touch screen*), através de uma caneta especial, ou até mesmo usando as mãos. Apresenta um painel de controle para o usuário e a tela possui aproximadamente 2,6 m².

3. Interatividade com Computador e Projetor

A sala de aula equipada com computador e projetor permite ao professor e aos alunos efetuarem apresentações e demonstrações. Neste cenário, apresentado pela figura 1, a tela recebe a projeção de uma apresentação. O usuário interage com o aplicativo do computador através dos periféricos do computador (teclado e *mouse*), e o usuário interage com a platéia através de um apontador laser sobre os objetos projetados na tela, ou com o próprio *mouse* que apresenta o seu cursor na tela de projeção.

As estruturas de software em execução no computador e os periféricos envolvidos neste cenário são apresentadas pela figura 2. A limitação desta estrutura é a dependência do usuário em ter que utilizar os periféricos do computador (teclado e *mouse*) para promover a interatividade da apresentação. O usuário tem que ficar na frente do computador para promover a interação com a apresentação. E nesse momento a platéia fica prejudicada pela perda de contato visual com o apresentador, pois este fica de costas ou de lado para a platéia, ou escondido atrás da tela do computador.

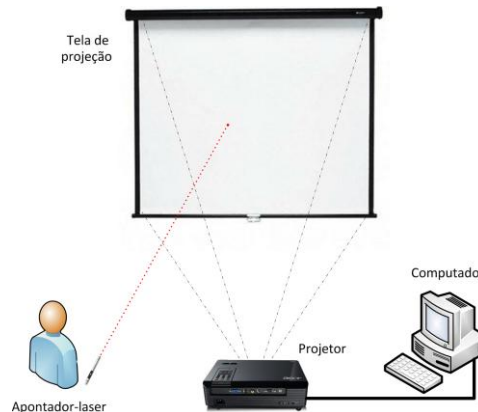


Figure 1. Cenário da interatividade entre o apresentador, o computador e o projetor.

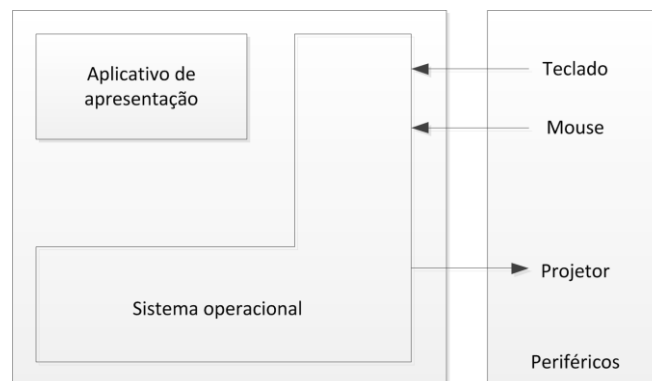


Figure 2. Aplicativos em execução no computador e periféricos envolvidos no cenário da figura 1.

A inserção neste cenário de um dispositivo que emule os comandos e os movimentos do *mouse* permite um aumento da interação entre o usuário e a apresentação, sem perda do contato visual do apresentador com a platéia.

4. Descrição funcional do Apontador-Mouse com o Computador e o Projetor

O aumento da interatividade entre o usuário e a apresentação faz-se dotando o apontador laser dos comandos de botão do *mouse*, e permitindo que o usuário execute os movimentos do *mouse* através do apontador laser. Esta é a ideia principal da pesquisa e do desenvolvimento do apontador-mouse.

Além dos elementos do cenário da figura 1, este novo cenário incorpora uma câmera de vídeo, para adquirir a imagem da tela de projeção, e o apontador-mouse, com

funções adicionais que aumentam as características de interatividade entre o usuário e o aplicativo de apresentação, não sendo necessário o usuário interagir com o *mouse* do computador. A figura 3 apresenta este cenário e os elementos envolvidos.

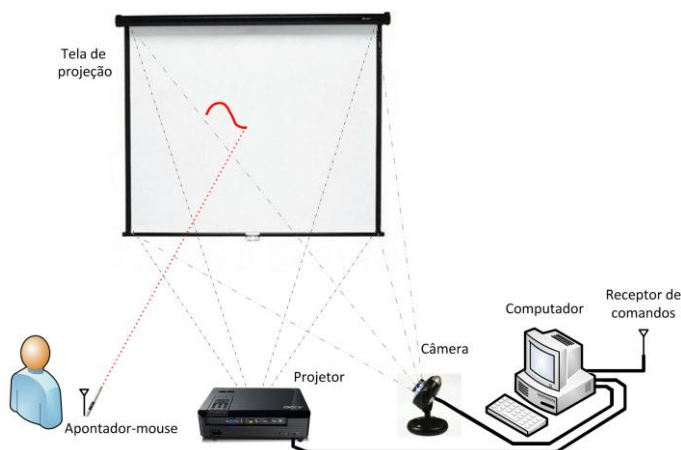


Figure 3. Cenário de interatividade entre o apresentador com o apontador-mouse, o computador e o projetor. Com o apontador-mouse o usuário possui o apontador laser, os botões que emulam os botões do mouse e o movimento do laser que emula o movimento do mouse.

O computador executa normalmente a apresentação, por exemplo, com algum aplicativo de apresentação. O aplicativo do apontador-mouse também está sendo executado. O usuário utiliza o apontador-mouse como sendo o apontador *laser* e os demais recursos emulam o funcionamento do *mouse*. A figura 4 apresenta a estrutura dos programas em execução no computador e os periféricos associados.

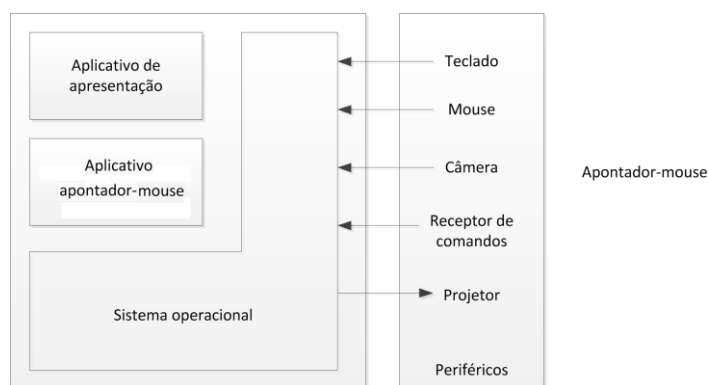


Figure 4. Aplicativos em execução no computador: aplicativo de apresentação e aplicativo do apontador-mouse. Periféricos envolvidos no cenário da figura 3, acrescido do apontador-mouse com o usuário e do receptor de comandos no computador.

4.1. Apontador-Mouse e o Receptor de Comandos

O apontador-mouse é um apontador laser comum com botões que emulam os controles do *mouse*. Ao acionar estes botões são emitidos comandos por rádio-frequência, que são

recebidos pelo receptor de comandos conectado na interface USB do computador. O aplicativo do apontador-mouse recebe estes comandos e transfere os eventos emulados para o sistema operacional, como se tivessem sido gerados pelo *mouse*, interagindo com os outros aplicativos em execução. Desta forma, através do apontador-mouse e do receptor de comandos são emulados os comandos dos botões do *mouse* do computador.

4.2. Câmera

Para aumentar a interação com o aplicativo no computador é necessário de outros recursos para emular o movimento do cursor do *mouse* pela tela. Uma câmera efetua a aquisição da imagem projetada na tela e o aplicativo do apontador-mouse analisa a imagem procurando identificar o laser. Caso o apontador-mouse esteja com o laser acionado na área da tela de projeção, o aplicativo do apontador-mouse determina a posição de contato do laser na tela, transformando esta informação em coordenadas, para as quais o cursor do *mouse* será deslocado. Se estas coordenadas estiverem sobre os controles (botões, barras de rolagem, etc...), da apresentação ou de um aplicativo em execução e o usuário gerar um comando no apontador-mouse emulando o *click* de um botão do *mouse*, o aplicativo do apontador-mouse irá gerar este evento para o sistema operacional, interagindo com a aplicação em exibição. A emulação do movimento do *mouse* pela tela é realizado pela análise da imagem projetada, capturada através da câmera, como apresenta o cenário da figura 3.

4.3. Aplicativo do apontador-mouse no computador

O aplicativo do apontador-mouse, desenvolvido em C#, executa diversas *threads* [Seixas Filho e Szuster 2003] no tratamento das informações recebidas e efetua o encaminhamento destes eventos para o sistema operacional. Desta forma, o aplicativo de apresentação recebe informações dos periféricos, que são gerenciados pelo sistema operacional, e que repassa informações do aplicativo do apontador-mouse.

O aplicativo do apontador-mouse analisa continuamente a imagem da tela recebida pela câmera e determina a posição do laser, transformando em coordenadas e atribuindo estas para a posição do cursor do *mouse*. Mudando a posição do laser na tela de projeção, o cursor do *mouse* irá seguir a trajetória descrita pelo laser.

Os comandos dos botões do apontador-mouse são tratados por *threads* no aplicativo do apontador-mouse, emulando os *clicks* do *mouse* e gerando estes eventos para o sistema operacional.

4.4. Comentários

O apontador-mouse, emulando todas as funcionalidades do *mouse* do computador, aumenta a interatividade da apresentação, permitindo uma maior liberdade de movimentação do usuário e um melhor contato com a platéia.

Neste cenário, apresentado pela figura 3, o usuário pode interagir com o aplicativo do computador através do apontador-mouse, sem o uso do *mouse* do computador. O apontador-mouse disponibiliza para o usuário todas as funcionalidades do *mouse* do computador, com a vantagem de estar interagindo com a apresentação através da tela de projeção e mantendo o contato visual com a platéia.

5. Detalhamento técnico do apontador-mouse

Este item aborda o detalhamento das soluções de *hardware* e de *software* implementadas no apontador-mouse.

5.1. Comandos do apontador-mouse

O apontador-mouse foi dotado de dois botões para emular os comandos do *mouse*. O controle destes botões é realizado por um microcontrolador MSP430F2274 e um transceiver de RF CC2500 do kit eZ430-RF2500 [Texas Instruments]. Foi implementado um protocolo para transmitir os eventos dos botões (botão esquerdo pressionado, botão esquerdo solto, botão direito pressionado, botão direito solto, apontador laser ligado, apontador laser desligado). O microcontrolador MSP430 [Texas Instruments] é um microcontrolador RISC de 16 bits, da Texas Instruments, de baixíssimo consumo de energia, adequado para aplicações portáteis alimentadas por bateria.

O protocolo de eventos transmitido pelo apontador-mouse é recebido pelo receptor de comandos, implementado também com um microcontrolador MSP430F2274 e um transceiver de RF CC2500. O *firmware* do receptor de comandos implementa o protocolo de recepção dos comandos do apontador-mouse. Após decodificado os comandos, estes são enviados para o aplicativo do apontador-mouse no computador através da interface USB. O *firmware* do apontador-mouse e do receptor de comandos foram implementados em linguagem C, no kit eZ430-RF2500. Foi utilizado o ambiente integrado de desenvolvimento, disponível pelos parceiros da Texas Instruments.

A figura 5 apresenta as rotinas da *thread* **RFThreadFunction** que interpretam os comandos enviados pelo apontador-mouse, recebidos pelo receptor de comandos e que emulam os *clicks* do *mouse*, representados no cenário da figura 3.

```
public void PressButton(string button)
{
    // rotina que emula o pressionar de botões do mouse
    currentButton = button;
    INPUT[] buffer = new INPUT[1];
    buffer[0].type = 0;
    buffer[0].mi.dx = buffer[0].mi.dy = 0;
    buffer[0].mi.mouseData = buffer[0].mi.time = 0;
    buffer[0].mi.dwExtraInfo = (IntPtr)0;
    if (currentButton == "left")
        buffer[0].mi.dwFlags = MOUSEEVENTF_LEFTDOWN;
    else if (currentButton == "right")
        buffer[0].mi.dwFlags = MOUSEEVENTF_RIGHTDOWN;
    SendInput(1, buffer, Marshal.SizeOf(buffer[0]));
}

public void ReleaseButton(string button)
{
    // rotina que emula o soltar de botões do mouse
    INPUT[] buffer = new INPUT[1];
    buffer[0].type = 0;
    buffer[0].mi.dx = buffer[0].mi.dy = 0;
    buffer[0].mi.mouseData = buffer[0].mi.time = 0;
    buffer[0].mi.dwExtraInfo = (IntPtr)0;
    if (currentButton == "left")
        buffer[0].mi.dwFlags = MOUSEEVENTF_LEFTUP;
    else if (currentButton == "right")
        buffer[0].mi.dwFlags = MOUSEEVENTF_RIGHTUP;
    SendInput(1, buffer, Marshal.SizeOf(buffer[0]));
}
```

Figure 5. Rotinas *PressButton()* e *ReleaseButton()* que emulam os eventos de pressionar e soltar os botões do mouse do computador.

5.2. Imagem da tela

O posicionamento ideal da câmera é perpendicular a tela de projeção no sentido horizontal e vertical, adquirindo uma imagem frontal da tela, e a certa distância para que a imagem adquirida tenha o tamanho da tela de projeção. Dificilmente será adquirida uma imagem nestas condições, sendo que a imagem da tela de projeção será uma sub-imagem com distorções de perspectiva, devido ao posicionamento da câmera. A figura 6(a) apresenta a captura de tela da imagem de teste e a figura 6(b) apresenta a imagem da tela de projeção capturada pela câmera.

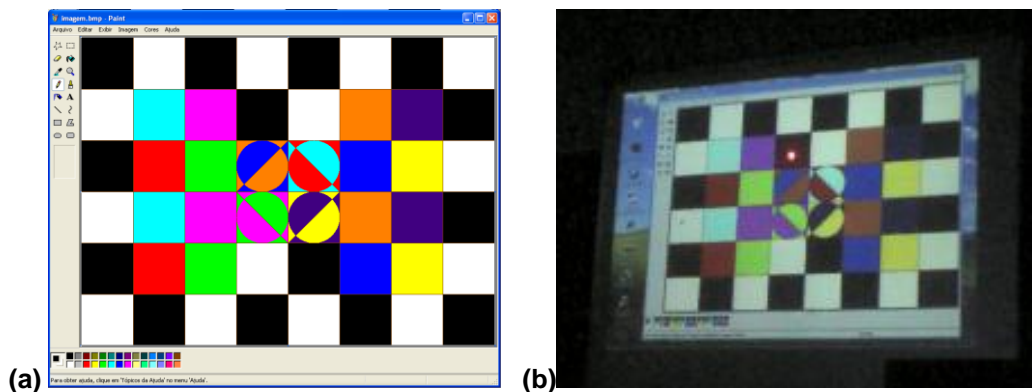


Figure 6. (a) Imagem de teste, (b) Imagem adquirida da tela de projeção.

A correção da imagem devido a distorção por perspectiva envolve a transformação linear de rotação em dois eixos para tornar a imagem perpendicular em relação a câmera [Vince, 2011]. Segue-se outra transformação necessária que é a aquisição da área de interesse da imagem (*region of interest* - ROI) com o redimensionamento da imagem para o tamanho desejado em pixels.

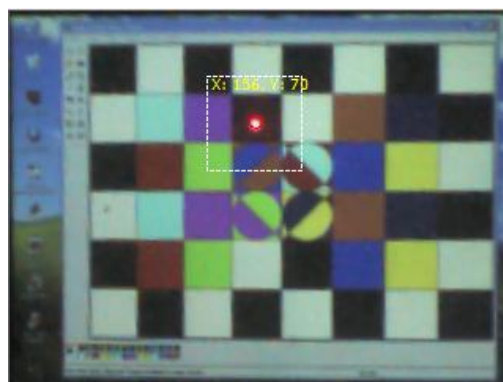


Figure 7. Imagem de teste da figura 6(b) com dupla rotação e reescalonamento, representando a imagem corrigida com o redimensionamento da ROI.

Devido as transformações de rotação, e o redimensionamento da imagem (mudança de resolução) existem diferenças entre a imagem original (figura 6(a)) e a imagem rotacionada e redimensionada (figura 7). A resolução da câmera e a posição da câmera, em relação a resolução do projetor, são os principais fatores que afetam a qualidade da imagem reconstruída.

Mas a qualidade da imagem reconstruída não é importante. O objetivo de todo este processo é estimar com a melhor resolução possível a posição do pixel central do feixe laser (coordenadas x e y) na tela de projeção em relação a imagem que está sendo projetada.

5.3. Detecção do laser na tela

As componentes da imagem são as cores: vermelha (R – *red*), verde (G – *green*) e azul (B – *blue*). Utilizando um apontador laser vermelho, foi aplicado um filtro com nível de matiz, saturação e luminância para identificar a região do laser na tela de projeção. Após aplicado o filtro, efetua-se um processo de busca na imagem pelo pixel de maior valor na componente vermelha.

A potência do apontador laser é superior a potência de iluminação do projetor, não ocorrendo interferências entre a componente vermelha da imagem da tela de projeção e o feixe do apontador laser. Isso é visível na figura 7 e na figura 8 onde o apontador laser encontra-se ligado. Devido a diferença de potência torna-se simples a identificação do laser em ambientes sem interferência da luz do sol e escurecidos com o desligamento da luz artificial do ambiente.

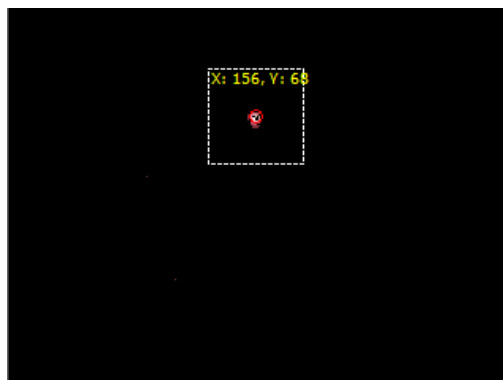


Figure 8. Imagem da figura 7 com filtro vermelho, binarizada com um valor elevado de *threshold* indicando a posição do apontador laser vermelho.

Na figura 7, que possui o apontador laser posicionado na tela, os pixels cuja componente vermelha possuem valor superior ao nível de decisão (*threshold*) são representados pelos seus valores, e os pixels com a componente vermelha com valor abaixo do valor de decisão são transformados para preto (0x00). Este processo gera a figura 8, com o nível de preto e os níveis de vermelho do apontador laser. Através de um algoritmo simples de busca na imagem reconstruída e binarizada, determina-se as coordenadas do pixel central da região vermelha, conforme apresenta a figura 8. Estas coordenadas que identificam a posição do apontador-laser ligado na tela são representadas também na figura 7. As figuras 7 e 8 são telas auxiliares utilizadas para o desenvolvimento do projeto, não apresentando utilidade para o usuário final.

Desta forma o aplicativo do apontador-mouse determina as coordenadas na tela do feixe de laser. Estas coordenadas são enviadas continuamente como um evento ao sistema operacional, fazendo com que dinamicamente o cursor do *mouse* siga o movimento do laser na tela. A figura 9 apresenta parte do código fonte destes controles.


```
//Tarefa que recebe coordenadas e movimenta o mouse
private void MoveCursorXY(void)
{
    while(true)
    {
        if(laserFound) //testa o flag se houve movimento do laser
        {
            // calcula a coordenada x da camera webcam para a resolução do monitor
            int x = Convert.ToInt32(((double)monitorXResolution/320)*xLaser);
            // calcula a coordenada y da camera webcam para a resolução do monitor
            int y = Convert.ToInt32(((double)monitorYResolution/240)*yLaser);
            if((x <= monitorXResolution && y <= monitorYResolution) &&
                (x >= 0 && y >= 0)) // verifica se as coordenadas são válidas
            {
                Cursor.Position = new Point(x, y);
                // move o cursor para a nova posição identificada na imagem
            }
            laserFound = false;
        }
    }
}
```

Figure 9. Código fonte que trata as coordenadas do posicionamento do laser na tela e emula o movimento do *mouse*.

O sistema foi testado com o apontador laser de cor vermelha. Foi utilizado também com apontador laser de cor verde, aplicando o filtro de verde e o sistema de *threshold* do feixe laser e foram obtidos os mesmos resultados, sendo que o apontador-mouse com laser vermelho e com laser verde funcionaram adequadamente. No aplicativo do apontador-mouse é configurável a cor do laser do apontador a ser utilizado (vermelho ou verde), para adequar o filtro a ser utilizado.

6. Resultados

São apresentados alguns resultados, como as figuras 10(a) e 10(b), para demonstrar o uso do apontador-mouse descrito no cenário da figura 3.

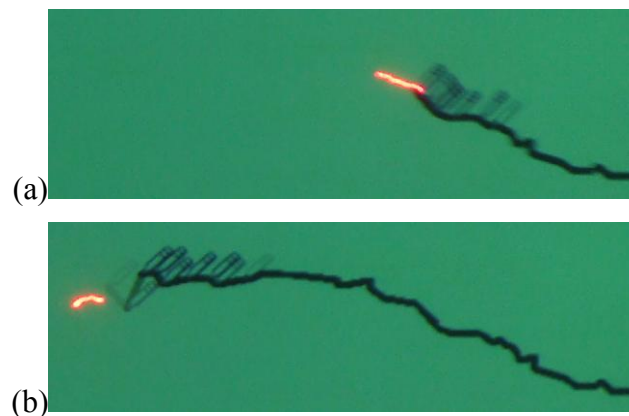


Figure 10. O apontador-mouse sendo utilizado no aplicativo Ms-paint para desenhar uma linha a mão-livre. Observa-se em (a) e em (b) o movimento do laser e do cursor do mouse na forma de lápis, seguindo o laser. O deslocamento do laser na tela foi registrado por uma máquina fotográfica digital configurada para fotos no escuro sem flash, aumentando o tempo de abertura do obturador (tempo de exposição do sensor CCD de imagem).

A figura 10 ilustra o uso do apontador-mouse interagindo com o aplicativo Ms-paint, na tarefa de desenhar uma linha a mão livre. Foi utilizado um laptop com a função

de rastro de *mouse* ativada para mostrar o percurso do cursor do *mouse* no acompanhamento do feixe de laser do apontador-mouse. O deslocamento do laser foi registrado nestas fotos pela máquina fotográfica configurada para um tempo elevado de obturador aberto.

7. Conclusões

O objetivo do projeto é substituir funcionalmente o *mouse* do computador pelo apontador-mouse, permitindo que o usuário interaja com a apresentação sem perder o contato visual da platéia para utilizar o *mouse* do computador. A emulação dos comandos do *mouse* por um dispositivo de acionamento remoto já é bem desenvolvido e utilizado a muito tempo, conforme apresentado.

O aumento da interatividade de apresentações foi realizado com a aquisição da imagem da tela de projeção e a determinação da posição do laser na tela, emulando o movimento do *mouse*. Desta forma, o usuário do apontador-mouse dispõe funcionalmente de um *mouse* remoto, podendo interagir com a apresentação e com outros aplicativos sem perder o contato visual com a platéia.

A limitação desta versão do projeto é o uso de uma câmera do tipo webcam com resolução de 320x240 pixels. A baixa resolução da câmera, em relação a resolução do projetor, e acrescido da ROI ser uma sub-imagem desta, gera uma imagem reconstruída com baixa resolução. Esta baixa resolução da imagem reconstruída determina que o passo de deslocamento do *mouse* são de alguns pixels na imagem real. O uso de uma câmera com resolução semelhante ou superior a resolução do projetor aumentaria a resolução do movimento do cursor do *mouse* gerado pelo apontador-mouse.

Referências

- Fundação Victor Civita (2009) “O uso do computador e da internet na escola pública”, disponível em <http://www.fvc.org.br/estudos-e-pesquisas/2009/uso-computador-internet-escola-publica-529446.shtml>.
- Infiniter, *Nobo P3 Page Point & Present* disponível em <http://www.infiniter.co.uk/shopdisplayproducts.asp?id=37&cat=Wireless+Remotes>.
- Smarttech, *SMART Board 800*, disponível em <http://smarttech.com/us/Solutions/Education+Solutions/Products+for+education/Interactive+whiteboards+and+displays/SMART+Board+interactive+whiteboards/885ix+for+education>.
- Seixas Filho, C., Szuster, M. (2003) “Programação Concorrente em Ambiente Windows – Uma visão de automação”, Editora UFMG.
- Targus, *Multimedia Presentation Remote AMP09US*, disponível em <http://www.targus.com/us/productdetail.aspx?sku=AMP09US>.
- Texas Instruments, Microcontrolador MSP430, disponível em: <http://focus.ti.com/mcu/docs/mcumspoverview.tsp?sectionId=95&tabId=140&familyId=342>.
- Texas Instruments, MSP430 Wireless Development Tool, disponível em: <http://focus.ti.com/docs/toolsw/folders/print/ez430-rf2500.html>.
- Vince, J., (2011), “Rotation Transforms for Computer Graphics”, Springer.