

**UNIVERSIDADE DO EXTREMO SUL CATARINENSE – UNESC
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

THAFFAREU ROBERTO BRIGIDO

**RASTREAMENTO DE OBJETOS APLICADO AO DESENVOLVIMENTO
DE UM PROTÓTIPO DE LOUSA DIGITAL INTERATIVA**

CRICIÚMA, DEZEMBRO DE 2011

THAFFAREU ROBERTO BRIGIDO

**RASTREAMENTO DE OBJETOS APLICADO AO DESENVOLVIMENTO
DE UM PROTÓTIPO DE LOUSA DIGITAL INTERATIVA**

Monografia apresentada para obtenção do grau de Bacharel em Administração de Empresas, no curso de Administração com Linha de Formação Específica em Comércio Exterior, da Universidade do Extremo Sul Catarinense, UNESC.

Orientador: Prof. MEng. Evânio Ramos Nicoleit

Co-Orientador: Prof. MSc. Paulo João Martins

CRICIÚMA, NOVEMBRO DE 2011

THAFFAREU ROBERTO BRIGIDO

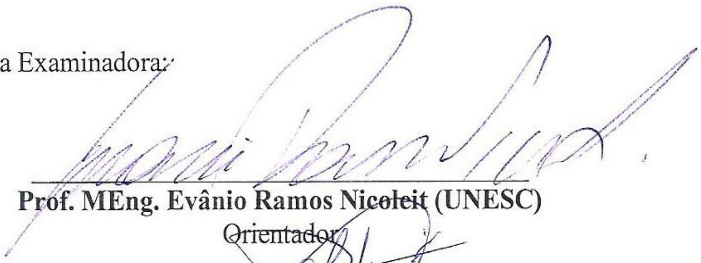
**RASTREAMENTO DE OBJETOS APLICADO AO
DESENVOLVIMENTO DE UM PROTÓTIPO DE LOUSA DIGITAL
INTERATIVA**

Submetido ao corpo docente do Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense como um dos requisitos para obtenção do grau de Bacharel em Ciência da Computação.



Profa. MSc. Ana Claudia Garcia Barbosa
Coordenadora do Curso de Ciência da Computação

Banca Examinadora:



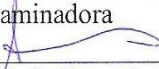
Prof. MEng. Evânio Ramos Nicoletti (UNESC)
Orientador



Prof. MSc. Paulo João Martins (UNESC)
Co-Orientador



Profa. MSc. Ana Claudia Garcia Barbosa (UNESC)
Examinadora



Profa. MSc. Leila Laís Gonçalves (UNESC)
Examinadora

DEDICATÓRIA

Aos meus pais, Paulo e Sandra, que
sempre me guiaram pelo caminho certo.

AGRADECIMENTOS

Gostaria de agradecer aos meus pais, que sempre estiveram motivando-me e guiando-me à forma correta.

A minha companheira, Diana, que sempre se fez presente, tendo paciência e compreensão, me ajudando e me mantendo firme.

Aos meus amigos de vivência em comum, Henrique, Jonas, Ítalo, Leandro, Hiago e todos os outros que acreditaram em mim.

Aos colegas de classe, em especial ao Lucas, Gélio, Guilherme, Anderson, Diego, Priscila, Cristina, Thiago, Jean e aos outros, que fizeram parte desta luta como uma vitória diária da busca pela conquista em vencer os obstáculos por nós enfrentados nas disciplinas, nas viagens, no trote e em muitos outros momentos que partilhamos juntos.

Às pessoas que entraram na minha vida por acaso e que mudaram minha forma de pensar sobre vários assuntos, tais como, Blissari, Vitor, Cris, Emily, Fran Pizetti, Samanta, Bianca, Charlana e Fran Abel.

RESUMO

O rastreamento de objetos é uma técnica que possibilita isolar e acompanhar um determinado objeto para vários fins, tais como detectores de colisões, radares e reconhecimentos faciais. Os modelos de *viewport*, por sua vez, possibilitam representar objetos de um domínio em outro suporte, tal como a tela de um computador ou uma projeção multimídia. Com base nessas considerações, este trabalho descreve o desenvolvimento de uma Lousa Digital Interativa – LDI, uma ferramenta de apoio ao processo educacional, baseado no rastreamento de objetos. O estudo discute a integração de modelos de *viewport* e rastreamento de objetos para o desenvolvimento da LDI. O trabalho também discute métodos de identificação e rastreamento de um objeto e seu transporte de um suporte para outro. O protótipo requereu o desenvolvimento de um *software* de conexão de um rastreador de objetos ao sistema. O rastreador de objetos utilizado foi um controlador de jogos do console Nintendo Wii que, por meio do uso de mapeamento *viewport*, simulou uma LDI. Os resultados são motivadores para o aprimoramento do sistema. O estudo indicou redução significativa de custos, já que o custo destes equipamentos é tipicamente elevado para as instituições de ensino.

Palavras chave: Lousa Digital Interativa – LDI. Rastreamento de Objetos. *Viewport*.

ABSTRACT

The object tracking is a technique that allows isolating and tracking a specific object for various purposes, such as collision detectors, radar and facial recognition. Models of viewport allow representing objects from a domain in another support, such as a computer screen or a multimedia projection. Based on these considerations, this work describes the development of a Digital Interactive Whiteboard - LDI, a support tool for the educational process, based on object tracking. The study discusses the integration of viewport and object tracking models for the development of LDI. The paper also discusses methods of identifying and tracking an object and its transport from one support to another. The prototype required the development of connection software between the object tracker and the system. The object tracking used was a Nintendo Wii game controller that through the use of viewport mapping simulated an LDI. The results are motivators for improving the system. The study found significant cost savings, since the cost of this equipment is typically high for educational institutions.

Key words: Digital Interactive Whiteboard – LDI. Object Tracking. Viewport.

LISTA DE ABREVIATURAS E SIGLAS

API - Application Programming Interface

GLCM - *Gray Level-Co-occurrence Matrices*

HID - Human Interface Device

HSV - *Hue, Saturation, Value*

IEEE - Institute of Electrical and Electronics Engineers

LED - Diodo Emissor de Luz

MEC - Ministério da Educação

JNI - Java Native Interface

JVM - Java Virtual Machine

RGB – Red Green Blue

USB - Universal Serial Bus

SRD - Sistema de Referência do Dispositivo

SRN - Sistema de Referência Normalizado

SRO - Sistema de Referência do Objeto

SRU - Sistema de Referência do Universo

LISTA DE ILUSTRAÇÕES

Figura 1 – Representação de objetos	22
Figura 2 – Câmera infravermelho do Wiimote	25
Figura 3 – Variações de movimento Wiimote	26
Figura 4 – Viewport com uma coordenada referencial	29
Figura 5 – Dois pontos referenciais o objeto acontecimento de clipping.....	29
Figura 6 – Transformação do objeto em referência SRU para SRD.....	30
Figura 7 – Exemplo e constatação do Viewport	31
Figura 8 – Representação de coordenadas do plano cartesiano.....	32
Figura 9 – Representação do sistema de coordenadas dos monitores	33
Figura 10 – Interface gráfica API WiiseJ.....	35
Figura 11 – Trecho de código.....	36
Figura 12 – Esquema de caneta infravermelho	40
Figura 13 – Ilustração do cenário	40
Figura 14 – Modelo da funcionalidade da API	42
Figura 15 – Modelo para sistema proposto.....	42
Figura 16 – Trechos de códigos da API WIIUSEJ.....	43
Figura 17 – Trecho do código IREvent	44
Figura 18 – Classe IREventchmandoIRSource	44
Figura 19 – Construtor e método ToString da classe IRSource	45
Figura 20 – Representação do objeto luz.....	46
Figura 21 – Trecho do código que mostra o evento de IR e movimentar o mouse	46
Figura 22 – Tela capturada do teste com coordenadas	47
Figura 23 – Modelo de Viewport sobr o protótipo	49
Figura 24 – Pontos calibrados	50
Figura 25 – Código fonte aplicado à fórmula de Viewport.....	50
Figura 26 – Aula de informática com a utilização do protótipo.....	52
Figura 27 – Secretário da Educação de Santa Catarina usando o protótipo	52

SUMÁRIO

1 INTRODUÇÃO	11
1.1 OBJETIVOS	12
1.1.1 Objetivo Geral.....	12
1.1.2 Objetivos Específicos	12
1.2 JUSTIFICATIVA	13
1.4 ESTRUTURA DO TRABALHO	14
2 TECNOLOGIAS NA EDUCAÇÃO	15
2.1 LOUSAS DIGITAIS INTERATIVAS - LDI.....	17
3 RASTREAMENTO DE OBJETOS	20
3.1 REPRESENTAÇÃO DO OBJETO.....	21
3.2 MÉTODOS USADOS PARA RASTREAMENTO	22
3.3 DISPOSITIVOS PARA RASTREAMENTO.....	24
3.3.1 Wiimote.....	25
4 VIEWPORT	27
4.1 SISTEMAS DE REFERÊNCIAS	27
4.2 TRANSFORMAÇÃO E VISUALIZAÇÕES 2D	28
4.2.1 Coordenadas cartesianas e coordenadas de monitores.....	32
5 JAVA	34
5.1 BIBLIOTECAS E APIs	34
5.1.1 Wiiusej.....	35
5.1.2 MoteJ.....	36
6 TRABALHOS CORRELATOS	37
6.1 HACKING THE NINTENDO WII REMOTE.....	37
6.2 RASTREAMENTO VISUAL DE OBJETOS UTILIZANDO MÉTODOS DE SIMILARIDADE DE REGIÕES E FILTRAGEM ESTOCÁSTICA.....	37
7 PROTOTIPAÇÃO DO SOFTWARE RECEPTOR E INTERPRETADOR DOS DADOS DE COORDENADAS	38
7.1 METODOLOGIA.....	38
7.2 O PROTÓTIPO.....	39
7.3 DISPOSIÇÃO DO HARDWARE	39
7.4 ENTENDENDO A API WIIUSEJ.....	41

7.4.1 Ir Tracking.....	44
7.5 EDIÇÕES NA API.....	46
7.5.1 Utilização de Viweport	48
7.6 TESTES E OBSERVAÇÕES.....	50
7.6.1 Instalação na Escola de Ensino Fundamental São Cristóvão	51
8 CONCLUSÃO.....	54
REFERÊNCIAS.....	55
APÊNDICE A.....	559

1 INTRODUÇÃO

Por rastreamento de objetos pode-se entender um método de seguir um determinado objeto através de sucessivos quadros de imagem, com o objetivo de se conhecer a forma como ele se move em relação a outros objetos (SHALOM; JAFFER, 1998).

Muitas técnicas vêm sendo desenvolvidas para esses sistemas de rastreamento em diversos segmentos, tais quais vigilância automatizada, monitoramento de tráfego, navegação automatizada de carros e interação humana com computador (SHAH; JAVED; YILMAZ, 2006), ressaltando-se que esta última técnica está sendo desenvolvida para fins de realidade virtual, jogos, quadro digitais, tecnologias *touchscreen*, entre outras.

Há vários modos de se analisar o objeto a ser rastreado. Contudo, há de se considerar que para rastrear um objeto deve-se isolar sua singularidade para que se possa distingui-los de outras formas ou objetos. Somente dessa forma é que o mesmo pode ser rastreado com precisão. As formas mais comuns de rastreamento são por cores, arestas, texturas e quantidade de brilho (SHAH; JAVED; YILMAZ, 2006).

Algumas formas se sobressaem em relação a outras dependendo da aplicação, como por exemplo, um sistema de rastreamento baseado em cores ou brilho, promove a percepção mais rápida dos objetos e cálculos de trajetórias. Entretanto, é menos seguro (DIETMAYER; SPARBERT; STRELLER, 2001).

Para uma detecção mais aprimorada usam-se algumas técnicas de isolamento de objetos, que podem ser por pontos, exclusão do fundo, segmentação e aprendizado supervisionado. Assim, dependendo do objeto a ser rastreado devem ser usadas técnicas diferentes. Alguns modelos usam algoritmos e técnicas de codificação de vídeo para melhorar o desempenho (BERNARDIN; GEHRIG; STIEFELHAGEN, 2009).

Conforme se verifica, existem muitos fatores que contribuem para a incerteza na localização exata de um objeto e sua configuração. Entre esses fatores incluem-se o ruído de medição, modelagem imprecisa, desorganização chamada falso-positivas, flutuações nas condições ambientais, entre outros (SMITH; PEREZ; ODOBEZ; BA, 2005).

Com base nessas considerações introdutórias, este trabalho focaliza o

estudo de técnicas envolvidas para a criação de um protótipo de *software* em plataforma Java, que seja capaz de processar os dados fornecidos por um *hardware* de rastreamento de objetos, para a criação de uma Lousa Digital Interativa – LDI. No trabalho, utiliza-se o método de rastreamento por câmera infravermelho e para a interação visual, faz-se uso de um projetor multimídia.

Contudo, convém ressaltar, conforme já mencionado, que, para cada domínio de aplicação existe um método mais eficiente. Dessa forma, para que a construção da lousa fosse possível e viável foram apenas estudados métodos que usam câmeras para rastreamento de imagens em 2D. Para tanto, fez-se necessário o levantamento de algumas questões sobre qual técnica poderia ser usada, como por exemplo, qual algoritmo adotado pelo *hardware* detector, qual a técnica de detecção, bem como a de isolamento, além de qual interface de comunicação poderia ser adotada, tempo de envio e resposta de dados, entre outros fatores que possibilitaram a viabilidade da construção da lousa digital.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Desenvolver um protótipo de Lousa Digital Interativa – LDI baseado no Rastreamento de Objetos.

1.1.2 Objetivos Específicos

- a) Identificar as técnicas de rastreamento de objetos;
- b) Analisar o modelo de viewport mapping;
- c) Criar um *software* capaz de receber dados de um rastreador;
- d) Utilizar técnica de *viewport mapping* para criação do protótipo.

1.2 JUSTIFICATIVA

O barateamento dos diversos tipos de equipamentos eletrônicos permitiu a ampliação da quantidade de recursos para apresentação de imagens, sons, vídeos e fotos. Isso fez enriquecer e aperfeiçoar os processos cognitivos e interativos, tornando o contato com a linguagem audiovisual mais presente em experiências vivenciadas pela sociedade atual. A exemplo disso, pode-se citar o surgimento das lousas digitais (NAKASHIMA; AMARAL, 2007).

Quadro ou lousa digital difere-se de outras tecnologias de recursos audiovisuais, como rádio, televisão ou o computador, porque ela aproxima e incorpora a interatividade entre esses recursos por processos comunicativos. Isso porque, essas lousas possibilitam integrar a linguagem audiovisual, tornando-as uma prática educativa para uso dos professores na prática didático-pedagógica (NAKASHIMA; AMARAL, 2007).

O uso dessas lousas contribui para atividades desenvolvidas pelo professor para novas práticas de ensino e de aprendizagem, proporcionando, assim, aos alunos uma compreensão e assimilação melhor das disciplinas ministradas, bem como fornece uma forma de motivação para a apreensão dos conteúdos (NAKASHIMA; AMARAL, 2007).

Dentro desse contexto, o rastreamento envolve a detecção e localização recursiva de um objeto ou objetos de interesse com base sequencial de dados, frame a frame. Esses métodos de detecção são usados amplamente em várias áreas, como por exemplo, a detecção de face em uma sequência de vídeo (GOKTURK; BOUGUET; GRZESZCZUK, 2001) e o acompanhamento de uma aeronave com radar de localização por robô móvel, usando medidas de laser gama (SHALOM; RONG; KIRUBARAJAN, 2001).

Assim, acredita-se que o estudo a respeito de sistema de rastreamento de objetos seja importante porque possui uma ampla quantidade de aplicações a serem exploradas, como na área de vigilância e segurança, na qual possibilita reconhecer pessoas, capta vídeo para a anotação automática dos mesmos, gerando resumos baseados em objeto. Também há uma grande gama na área de gestão de tráfego, tais como fazer análises sobre o fluxo de veículos na detecção de acidentes. Pode ser também usado no esporte, do mesmo modo que pode ser utilizado na área deste

estudo, ou seja, em jogos e aplicações práticas interativas, para as quais pode fornecer maneiras usuais de interação com sistemas, tais como lousas digitais.

Diante disso, justifica-se o uso de um método de rastreamento de objetos por câmera infravermelho acoplado a um controle do console Nintendo Wii, devido a sua capacidade de rastrear a posição e a orientação de um determinado objeto pela radiação infravermelha com alta velocidade. Além disso, há a grande precisão, em função da resolução da câmera do sensor, que consegue prover uma captura de 1024x768 pixels, mais de 4 bits para tamanho do feixe luminoso, possuindo uma taxa de atualização de 100Hz. Outro fator que justifica o método utilizado é pelo fato de que o mesmo é baseado em infravermelho, o que minimiza os dados transportados, sendo mais rápido na transferência, além de facilitar na implementação. Na verdade, acredita-se que, essa técnica, comparativamente a outros sistemas de rastreamento por câmeras que não sejam por meio de infravermelho (webcams), normalmente atingem apenas 640x480, com taxa 30Hz, requerendo um grande processamento da CPU para conseguir se tornar um rastreamento em tempo real, tornando assim inviável.

1.4 ESTRUTURA DO TRABALHO

Para cumprir os objetivos pretendidos, o presente trabalho foi estruturado da forma relatada na sequência.

Num primeiro momento, contempla aspectos referente à educação no contexto brasileiro e mundial considerando as novas tecnologias digitais e informacionais para melhoria no processo ensino-aprendizagem, tal como o quadro digital.

O segundo capítulo trata sobre técnicas e formas de rastreamento de objetos, bem como sua utilização na criação de um protótipo de lousa digital de baixo custo.

O terceiro capítulo relata ferramentas de hardware, como o caso de sensores disponíveis no mercado, e softwares que permitem a realização do protótipo de forma satisfatória.

2 TECNOLOGIAS NA EDUCAÇÃO

A capacidade das tecnologias informacionais, em especial o computador, de processar e exibir programas com sons, imagens, animação e textos, de modo interativo e integrado pode contribuir significativamente para a melhoria da qualidade do processo ensino-aprendizagem (MORAES, 2007)

A tecnologia de informação e comunicação articulada com outros recursos tecnológicos e diferentes mídias (TV, computador, Internet, vídeo) oferecem meios para o aluno diversificar a representação de conhecimento através de diferentes formas de linguagem, que são específicas de cada um desses recursos. No sentido de resolver um problema ou desenvolver um projeto, o aluno aprenda a selecionar, comparar, diferenciar, buscar informações que sejam significativas para aquilo que estiver produzindo (BARBOSA, 2002).

A utilização do computador no ensino, que permite a realização de atividades pedagógicas por seu intermédio encontra nos aplicativos que envolvem processamento de texto, planilhas, manipulação de banco de dados, construção e transformação de gráficos, sistemas de autoria, etc., mas que também possibilite ao aluno do campo criar ambientes de aprendizagem que façam surgir novas formas de pensar e aprender de acordo com sua realidade e aplicação no seu dia a dia.

Alguns aspectos podem ser destacados com relação à melhoria da qualidade no processo ensino-aprendizagem com a informática como recurso pedagógico: desenvolvimento da compreensão de conceitos matemáticos; melhoria da linguagem e da escrita; favorecimento da interdisciplinaridade; desenvolvimento da criatividade; desenvolvimento da autonomia, da cooperação e da criticidade (MORAES, 2007).

Ambientes computacionais que utilizam ferramentas apropriadas criam espaços para a interdisciplinaridade, através de projetos e atividades que integram várias disciplinas. O computador, analisa o autor, é visto como um objeto para a expressão da criatividade e uma ferramenta para a integração e a organização de conteúdos socialmente relevantes (MORAES, 2007)

Em função disso, a escola, principalmente, a pública precisa garantir aos alunos o acesso às mídias de forma ativa e produtiva, favorecendo a comunicação e, conseqüentemente, a possibilidade de fazer circular, em condições de igualdade, diferentes discursos e entendimentos. Novos objetos só podem ser valorizados,

analisados e utilizados de maneira crítica e inovadora se forem compreendidos. Portanto, para atuar e intervir no espaço eletrônico é necessário desenvolver a fluência tecnológica, explorar as telecomunicações no trabalho, entrar em rede para trocar idéias com os pares, aprender a se localizar, mover, estabelecer parcerias e cooperar em ambientes virtuais (BARBOSA, 2002).

A tecnologia da informática deverá ser instrumento para desenvolver habilidades nos alunos, de forma que, os conteúdos trabalhados na escola sejam significativos socialmente e provoquem mudanças individuais e coletivas. Deverá promover a democratização do ensino no Brasil e instituir a formação de cidadãos participativos para a construção de uma sociedade mais justa e melhor.

A grande disseminação da informática, em segmentos importantes da sociedade, revoluciona formas tradicionais de equilíbrio e institui um novo paradigma que alcança e modifica a comunicação, os modos de aprendizagem, as relações humanas e as organizacionais. Com as mudanças ocorridas na organização das sociedades através das inovações técnicas e científicas que configuram a chamada revolução tecnológica, a educação escolar não pode ficar à margem dessa evolução (BOELTER, 2006).

Na área das comunicações, ocorre a popularização dos recursos da informática, antes restrita a pequenos setores privilegiados; atualmente novos instrumentos permitem acesso mais ágil às fontes de conhecimento e recursos que cada vez mais, favorecem avanço cultural, e meios para adquirir conhecimentos. Neste contexto a escola é a principal condutora desse processo (CASTELLS, 1997).

Mas, pensar em inovações tecnológicas educacionais é antes de tudo repensar os ambientes de aprendizagem e a capacitação dos profissionais da educação e não apenas a aplicação da técnica pela técnica. Ambientes de aprendizagem abertos e motivadores através da multimídia devem contribuir para promover as competências nos alunos de aprender a aprender (BOELTER, 2006).

Por isso, para garantir o êxito da incorporação do recurso informático como instrumento útil para a atividade intelectual, criativa e profissional, é preciso que se garanta uma capacitação do docente não somente em nível técnico, como também, e fundamentalmente, pedagógico. O professor precisa apropriar-se da tecnologia em função de seus interesses profissionais, e assim situar-se, avaliar e planejar sua aplicação em aula (BOELTER, 2006).

Desse modo, em meio ao cenário tecnológico em que se encontra o

profissional docente, as atuais discussões e políticas públicas na área de informática (na educação) têm considerado o professor como um componente fundamental para o processo de introdução do computador no cotidiano do ensinar e aprender, mas na prática pouco se tem feito (CASTELLS, 1997).

Espera-se que o docente, na sala de aula, promova a interação entre a informática e a sua disciplina e, por meio dessa interação, proporcione aos alunos o acesso às novas informações e experiências, de modo que aprendam efetivamente, e que sejam críticos diante das informações e do conhecimento promovido por meio da tecnologia.

Assim sendo, é importante que em um processo de formação em informática na educação o professor seja concebido não apenas como um profissional, mas como uma pessoa que tem sentimentos e reações diversas diante do computador. Desse modo, não basta apenas o professor saber operacionalizar a tecnologia, ele precisa compreender as implicações pedagógicas envolvidas no seu uso para poder criar condições de aprendizagens que favoreçam o processo de construção de conhecimento do aluno e saiba aplicar esse conhecimento para resolver problemas cotidianos ou participar da busca de alternativas para solucioná-los (BARBOSA, 2002).

2.1 LOUSAS DIGITAIS INTERATIVAS - LDI

Conforme visto, a tecnologia vem atingindo pontos cruciais para a educação, sua crescente e constante inovações e atualizações sobre como definir e representar informações está cada vez mais visível no contexto atual, como foi o caso do computador, que ajudou a elaboração de trabalhos, apresentações áudios-visuais, entre tantas outras melhorias. Outra conquista para o mundo da educação foi o ensino a distância, graças à internet e seus meios de divulgação da informação ficou muito mais fácil aprender, com esses novos métodos de ensino, como aulas por vídeo conferência e fórum de atividades pela internet. (MORAN, 2000).

De acordo com a Lei de Diretrizes e Bases da Educação Nacional, de 1996 (LDB), em seu artigo 36, a introdução das novas tecnologias deve fazer parte dos currículos das escolas públicas do País, bem como discussão crítica dos meios de comunicação social. Esse processo perpassa, obrigatoriamente, pela necessidade de explorar os diversos meios como: a televisão, o rádio, o jornal, o vídeo, a revista, o

computador, o teatro, o fax, os *softwares* educacionais (FRANÇA; SANTOS, 2005).

Não resta dúvida de que, nos dias de hoje, a utilização de novas formas de interação on-line atende às novas necessidades dos alunos; o incentivo à aprendizagem ativa e significativa ao aluno já pode ser comprovada por meio de vários projetos já desenvolvidos em todos os países; é evidente o acesso rápido e eficiente na obtenção de informações relevantes e diversificadas e a melhoria da qualidade da comunicação entre professores e alunos são viabilizadas pelas ferramentas interativas (BUCKINGHAM, 2008).

A revolução tecnológica se reveste de uma grande promessa para a educação. A tecnologia da comunicação e de processamento de dados e imagens está avançando em uma velocidade enorme e sem precedentes, ao mesmo tempo em que se faz mais econômica e confiável, com enormes conseqüências para a educação, oferecendo, em escala crescente, um potencial preciso e bem definido como um recurso pedagógico, campo no qual, mesmo que ainda não esteja ocorrendo de forma plena, se reveste de uma possibilidade de uma revolução tecnológica (BUCKINGHAM, 2008).

A tecnologia de informação e comunicação articulada com outros recursos tecnológicos e diferentes mídias (TV, computador, Internet, vídeo) oferecem meios para o aluno diversificarem a representação de conhecimento através de diferentes formas de linguagem, que são específicas de cada um desses recursos. No sentido de resolver um problema ou desenvolver um projeto, o aluno necessita aprender a selecionar, comparar, diferenciar, buscar informações que sejam significativas para aquilo que estiver produzindo (BARBOSA, 2002).

Em função disso, a escola, principalmente, a pública, precisa garantir aos alunos o acesso às mídias de forma ativa e produtiva, favorecendo a comunicação e, conseqüentemente, a possibilidade de fazer circular, em condições de igualdade, diferentes discursos e entendimentos. Novos objetos só podem ser valorizados, analisados e utilizados de maneira crítica e inovadora se forem compreendidos. Portanto, para atuar e intervir no espaço eletrônico é necessário desenvolver a fluência tecnológica, explorar as telecomunicações no trabalho, entrar em rede para trocar idéias com os pares, aprender a se localizar, mover, estabelecer parcerias e cooperar em ambientes virtuais (BARBOSA, 2002).

Devido a isso, atualmente discute-se a aprendizagem por meio do uso de lousas digitais. O artefato, integrado à sala de aula, sugere a possibilidade de avanços

nos processos de aprendizagem.

A lousa digital é uma tecnologia moderna e inovadora com recursos que podem auxiliar na criação de novas metodologias de ensino. Existem vários modelos de lousas digitais, variando o tamanho, a marca e o custo, mas a maioria é composta por uma tela conectada a um computador e um projetor multimídia. A superfície dessa tela é sensível ao toque, isto é, quando alguém executa algum movimento sobre ela, o computador registra o que se fez em um *software* específico que acompanha a lousa digital (LÉVY, 1993, p. 40).

Neste contexto de educação com base em recursos tecnológicos, a lousa digital pode ser de grande valia no processo de aprendizagem. Com essa lousa, os professores poderão usar recursos como imagens, sons, e vídeos, navegar na internet e muitos outros recursos. Entretanto, devido ao alto custo de equipamentos como este, faz com que muitas escolas não consigam ter acesso a tal, e uma vez que escolas e universidades particulares já possuam essa tecnologia, nas escolas públicas essa é uma realidade muito distante.

Isso porque, geralmente, estas LDI em sua maioria são extremamente caras para implantar na escola. Muitas secretarias de Educação e seus gestores entendem que é importante para os alunos e para o ensino o uso de LDI, entretanto o alto custo das lousas dificulta a implantação das mesmas. Além disso, há o entendimento de que muitos professores não estão capacitados para o uso de tal tecnologia, fazendo com que as lousas fiquem em desuso ou sendo usadas de forma equivocada, não tendo assim o devido resultado esperado.

Contudo, novas pesquisas estão sendo desenvolvidas para auxiliar a educação e suas formas de aprendizado com uso de artefatos tecnológicos educacionais. Essas novas pesquisas levam a crer que será possível a prototipação de uma lousa digital usando técnicas de rastreamento de objetos, permitindo baratear os custos de uma lousa digital.

3 RASTREAMENTO DE OBJETOS

O rastreamento de objetos pode ser considerado como uma tarefa de extrema importância dentro do campo de visão computacional. Pode ser definido como um processo de localização de uma ou mais características desejadas de um objeto em vários frames de uma cena, seguindo sua semelhanças em cada cena para criar uma possível rota para rastrear tal objeto (VIDAL; ALCALDE, 2005).

O rastreamento visual de um objeto em movimento por um observador humano consiste no acompanhamento do movimento do objeto pelo sistema de visão do observador. A visão vai capturando as imagens do objeto e o cérebro as processa de modo que seja possível seguir a trajetória descrita. As diversas aplicações deste processo, bem como a necessidade da sua automação, levaram à tentativa de implementação do rastreamento por uma máquina dotada de um sistema de visão artificial para captura de imagens e de um sistema de processamento digital destas imagens para acompanhar a trajetória sem perder de vista o objeto (VIDAL, 2009, p.18)

A proliferação de computadores de alto desempenho, a disponibilidade de alta qualidade e de baixo custo das câmeras de vídeo, bem como a necessidade crescente de automatização nas análises de vídeo geraram um grande interesse em algoritmos de monitoramento de objetos. Dentro desse contexto, há três passos importantes em vídeo análise: detecção de objetos em movimento, rastreamento de objetos e análise do objeto de modo a reconhecer seu comportamento.

Atualmente, novas aplicações e metodologias surgiram para o rastreamento visual de imagens digitais, tanto para múltiplos como para únicos objetos, fazendo com que várias pesquisas estejam sendo elaboradas em diversas áreas, tais como, biomedicina, automação de processos, indústria bélica, sistemas de monitoramento, vigilância entre outras.

Há duas etapas fundamentais a serem cumpridas para a realização completa de um rastreamento visual de objetos. A primeira é a representação e a localização do objeto, que é essencialmente um processo de *bottom-up*, que lida com as mudanças no aparecimento do alvo, e a filtragem associada aos dados, que é essencialmente um processo de *top-down*, que lida com a dinâmica do objeto rastreado, e avaliação de diferentes hipóteses.

A forma como os dois componentes são combinados e ponderados é dependente da aplicação e desempenha um papel decisivo para a eficiência do rastreador, como por exemplo, o monitoramento de um rosto em uma cena precisa

confiar mais no alvo do que na sua representação sobre a dinâmica de destino (DECARLO; METAXAS, 2000), enquanto que em aéreas de vigilância de vídeo, por exemplo, o movimento do destino e o movimento da câmera são os componentes mais importantes.

Para aplicações em tempo real, apenas uma pequena porcentagem dos recursos do sistema pode ser alocada para o rastreamento, sendo o resto necessário para as fases de pré-processamento ou de alto nível, tarefas como reconhecimento, interpretação trajetória, e raciocínio. Portanto, é desejável manter a complexidade computacional de um sistema de rastreamento reduzida (WILDES *et al*, 2001).

3.1 REPRESENTAÇÃO DO OBJETO

Em um cenário de rastreamento, um objeto pode ser definido como algo que é de interesse para outras análises. Por exemplo, os barcos no mar, os peixes dentro de um aquário, os veículos em uma estrada, aviões no ar, as pessoas andando em uma estrada ou bolhas na água são um conjunto de objetos que podem ser importantes para controlar em um domínio específico. Os objetos podem ser representados por suas formas e aparências. Estes podem ser representados nesse conjunto de formas:

- **Pontos:** o objeto é representado por um ponto central (Figura 1 (a)), ou por um conjunto de pontos (Figura 1 (b)). Em geral, a representação de ponto é adequado para rastreamento de objetos que ocupam pequenas regiões em uma imagem (SERBY, 2004).

- **Formas geométricas primitivas:** o objeto é representado por um retângulo, uma elipse (Figura 1 (c), (d)), etc. O uso de formas geométricas primitivas são mais adequados para representar objetos simples e rígidos (COMANICIU, 2003).

- **Objeto pontilhado e contorno:** essa representação define o limite de um objeto (Figura 1 (g), (h)). A região no interior do contorno é chamada de silhueta do objeto (Figura 1 (i)). Silhueta é a representação do objeto, esse tipo de rastreamento é adequado para o rastreamento de formas não rígidas e complexas (YILMAZ, 2004).

- **Formas articuladas:** objetos articulados são compostos de partes do corpo que são realizada em conjunto com as articulações. Por exemplo, o corpo humano é um objeto articulado com tronco, pernas, mãos, cabeça e pés ligados por

articulações. A relação entre as partes é governada por modelos cinemáticos de movimento, como o ângulo articular. A fim de representar um objeto articulado, pode-se juntar modelo das partes constituintes de cilindros ou elipses, como na Figura 1 (e) (AGGARWAL, 2001, tradução nossa).

- **Modelos esqueléticos:** estes modelos podem ser extraídos através da aplicação do eixo do meio, de forma a transformar o objeto em silhueta. Este modelo é muito usado como uma maneira de representação para reconhecer objetos. A representação do esqueleto pode ser usada para a modelagem dos objetos articulados e rígidos (Figura 1 (i)) (BALLARD, 2001).

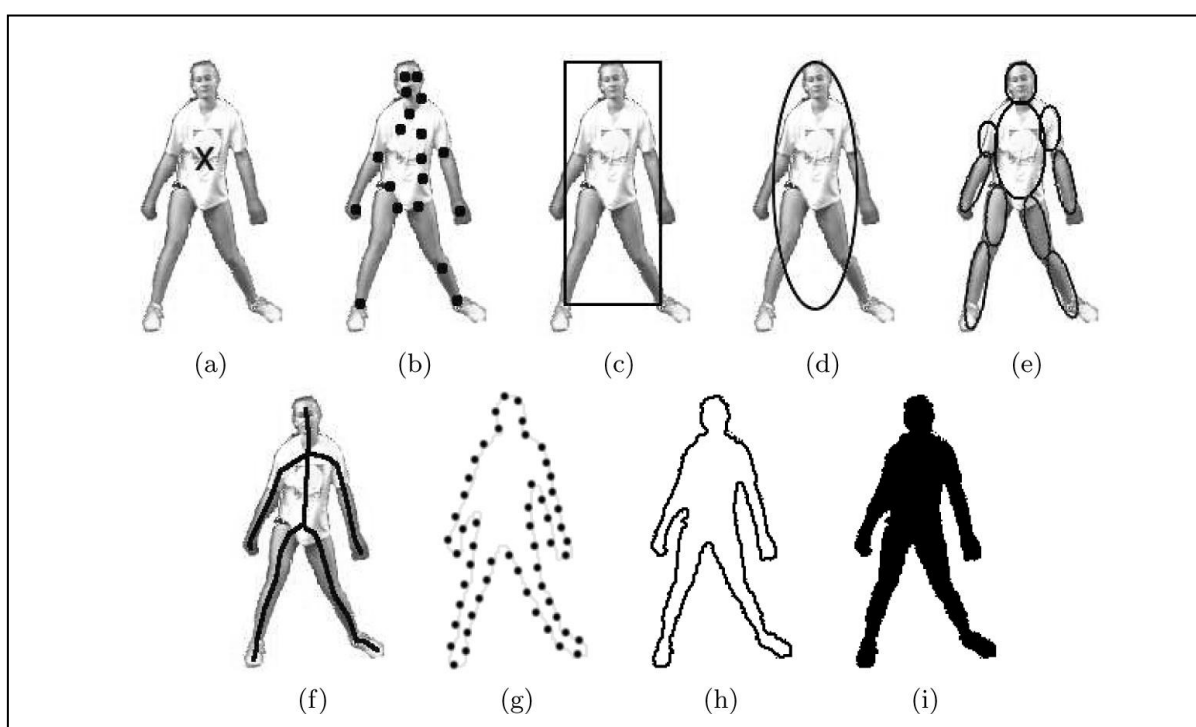


Figura 1 – Representação de objetos
Fonte: Ballard (2001)

Desse modo, conforme pode-se evidenciar, os objetos podem ser representados por suas formas e aparências, representadas pelo conjunto de formas descritas.

3.2 MÉTODOS USADOS PARA RASTREAMENTO

Selecionar as características certas desempenha um papel crítico no rastreamento. Em geral, o mais desejável é isolar a singularidade de um objeto dentro

do recurso visual para que os objetos possam ser facilmente distinguidos em função do espaço. A seleção de recursos está intimamente relacionada com a representação do objeto. Por exemplo, a cor é usada como um recurso para o aparecimento baseado em representações de histograma, enquanto que para a representação da base de contorno, as bordas do objeto são normalmente utilizadas como recursos.

Em geral, os algoritmos de rastreamento podem recorrer a uma combinação desses recursos, obtendo um melhor desempenho no rastreamento e usando menor complexidade e recursos computacionais. Os detalhes das características visuais comuns são os descritos na sequência:

- **Cor:** a cor aparente de um objeto é influenciada principalmente por dois fatores físicos: a distribuição de potência de luminosidade e o reflexo de luz sobre superfície das propriedades do objeto. No processamento de imagem, o RGB é geralmente usado para representar cores.

No entanto, o espaço RGB não é perceptivelmente uniforme dentre a gama de cores, ou seja, as diferenças entre as cores no espaço RGB não correspondem às diferenças de cor perceptível por seres humanos (PASCHOS, 2001).

Além disso, as dimensões RGB são altamente correlacionadas. Enquanto o *Hue, Saturation, Value* (HSV) é uma gama de cor uniforme. No entanto, esses espaços de cor são sensíveis a ruídos (SONG; KITTLERS, PETROU. 1996). Em resumo, não há última palavra sobre qual espaço de cor é mais eficiente, portanto, qualquer um dos dois sistemas podem ser implementados para o rastreamento de um objeto pela sua cor.

- **Bordas:** os limites dos objetos costumam gerar fortes mudanças na intensidade da imagem. Detecção de bordas é usada para identificar essas mudanças. Uma propriedade importante das bordas é que elas são menos sensíveis às mudanças de iluminação em relação às características de cor.

Algoritmos que seguem o contorno dos objetos normalmente usam o recurso de bordas como representante. Devido a sua simplicidade e precisão, a abordagem de detecção de borda mais populares é o Bordas de Canny (CANNY 1986).

- **Fluxo óptico:** é um denso campo de vetores de deslocamento que define a tradução de cada pixel em uma região. Ele é calculado a partir da restrição de luminosidade, que assume a constância de luminosidade dos pixels correspondentes em quadros consecutivos (HORN, SCHUNK 1981). Fluxo óptico é normalmente usado como um recurso em movimento com base em aplicações de segmentação e

rastreamento (HORN, SCHUNK 1981).

- **Textura:** é uma medida da variação da intensidade de uma superfície que quantifica propriedades, tais como a lisura, densidade, irregularidades e etc. Comparado-se à cor, a textura exige uma etapa de processamento para gerar descritores.

Existem vários descritores de textura: *Gray Level-Matrices Cooccurrence* (GLCM), que é um histograma 2D que mostra a concorrência de intensidades em uma direção específica e a distância. Similar ao método de bordas os recursos, as características de textura são menos sensíveis às mudanças de iluminação relação à cor.

Geralmente, as características são escolhidas manualmente pelo usuário, dependendo do domínio de aplicação. Para tal, há uma combinação entre método de rastreamento e reconhecimento de objeto, de forma que para cada aplicação se consiga chegar a um resultado melhor, como por exemplo, para um quadro digital, o grupo de rastreamento que deve ser estudado. Neste caso, seria o de reconhecimento por pontos e de formas geométricas primitivas, e para o rastreamento, usar-se-ia fluxo óptico para se escolher uma câmera por infravermelho, e usar o rastreamento por cor, caso opte-se por uma câmera digital de vídeo comum (WILDES; *et al*, 2001).

3.3 DISPOSITIVOS PARA RASTREAMENTO

Entre os tipos de dispositivos rastreadores que podem ser utilizados para a confecção de um lousa digital, podem-se citar as câmeras digitais.

As câmeras são naturalmente a principal classe de sensores. Além disso, o processamento de visão e os ganhos provenientes do aumento do poder de computação das últimas décadas permitem uma análise em tempo real para o monitoramento de um objeto com o mesmo exercendo certo esforço de *hardware* razoavelmente aceitável (BEHRINGER,1997).

No entanto, a diferença entre os dados da imagem bruta e as representações geométricas dos objetos interessantes ainda continuam grandes, especialmente os parâmetros geométricos, como a distância e a velocidade de outros objetos, mas exige ainda mais poder de processamento e esforço de *hardware*.

Outro sensor disponível e de fácil acesso seria o do tipo sensores infravermelhos, que conseguem uma análise dos rastreamento de objetos em tempo

real, com pouquíssimo esforço computacional, como é o caso do Wiimote, sobre o qual discorre-se a seguir.

3.3.1 Wiimote

O Wiimote é um controle de um console de vídeo game. Possui uma câmera de infravermelho com definição de 1024x768 100Hz e é acompanhado de duas fontes de infravermelho, possibilitando-se descobrir a exata localização do controle. Esta possibilidade pode ser usada para rastrear um objeto, e verificar o caminho que está seguindo (VICARIA, 2007). A Figura 2 representa esse dispositivo de rastreamento.

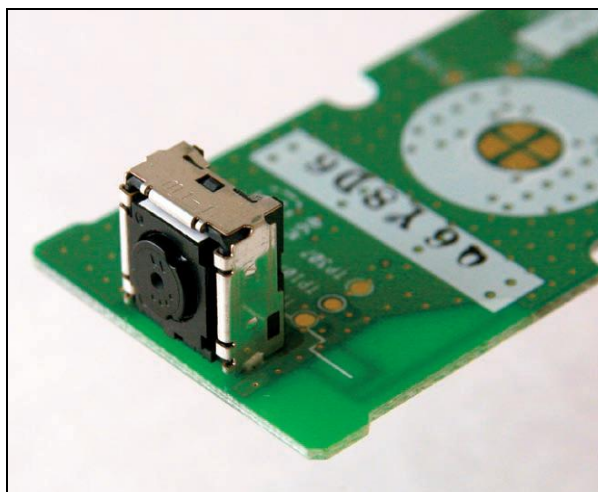


Figura 2 – Câmera infravermelho do Wiimote
Fonte: Lee (2007)

Também é possível usar o Wiimote como uma câmera fixa, que rastreia o movimento de dispositivos que emitem luz infravermelha (LEE, 2007).

Em outra definição, tem-se que o Wiimote é um controle de um console que tem uma câmera de infravermelho capaz de rastrear emissões de luz providas de *leds* infravermelhos, de modo que possa ser usado para apontar para a tela. O dispositivo se enquadra na categoria de fluxo óptico, ou seja, aquele que capta e interpreta as diferentes formas e intensidades de pontos luminosos.

O sensor inclui um sistema PixArtona-Chip, que permite um processamento das imagens dentro do controle, fazendo com que apenas sejam enviados os dados essenciais para fazer o rastreamento, pelo fato de o Wiimote ser um dispositivo sem

fio, que usa Bluetooth como protocolo de comunicação. Devido a isso, é possível enviar e receber dados para o controle remoto de qualquer computador equipado com Bluetooth, apenas um simples adaptador.

O Wiimote vai se identificar com o perfil *Human Interface Device* (HID), que é comumente empregado por muitos periféricos USB Bluetooth (WILLI, 2007). Seu sistema transfere uma quantidade de dados diversos, tal como as coordenadas de posição de todos os tamanhos do ponto e valor de pixels envolvidos.

Na Figura 3, encontram-se representadas as variações de movimento Wiimote.



Figura 3 – Variações de movimento Wiimote
Fonte: Lee (2007)

Além disso, estes dados podem ser recuperados por meio de diferentes modos, ampliado ou integral, por exemplo.

4 VIEWPORT

Uma vez entendido o funcionamento sobre rastreamento de objetos e qual equipamento será usado cabe-se então definir como se dá a representação gráfica da lousa digital e como será feito o enquadramento das informações obtidas pelos recursos de hardware para o *software* da LDI. Em função disso, neste capítulo, definem-se os conceitos sobre o método de *Viewport* dentro área de computação gráfica.

Viewport Mapping é a limitação de uma região na qual o desenho ou um objeto em questão é exposto em uma janela de exibição. Essa janela representa um determinado objeto do mundo real ou de um desenho para sua representação digital, bem como o seu posicionamento e escalonamento fiel do objeto que se busca. Para fazer essa representação digital, usam-se cálculos de geometria e matemática por um sistema de coordenadas (BLINN, 1992).

Essas equações também podem ser derivadas de um conjunto de transformações que converte a área da janela para a área do visor (BLINN, 1992)

4.1 SISTEMAS DE REFERÊNCIAS

Um sistema de coordenadas é a referência de objetos de um determinado meio para outro. É medido em tamanho e posicionamento de sua representação de um plano para outro.

Quando esse sistema de coordenadas é usado para alguma finalidade específica, ele ganha o nome de Sistema de Referência, e deve respeitar características fundamentais para que isso ocorra e apresentar uma unidade de referência básica e seus limites extremos de valores. No mercado atual, há alguns sistemas bem utilizados, que são:

- **SRU (Sistema de Referência do Universo):** Também é denominado como coordenadas do mundo. Esse sistema descreve os objetos em termos de aplicação do usuário, exemplos: centímetros, metros.

- **SRO (Sistema de Referência do Objeto):** Este sistema descreve um objeto com o próprio sistema de coordenadas como se ele fosse um pequeno universo dentro dele. O centro desse sistema coincide muitas vezes com seu centro de gravidade.

- **SRN (Sistema de Referência Normalizado)**: Usa os valores entre 0 e 1. Ele é como uma referência para intermediar os valores do SRU e SRD, fazendo com que as imagens geradas sejam independentes do dispositivo, ou seja, pode gerar imagens não importando a resolução, sempre gera imagens fiéis e mantendo as devidas proporções.

- **SRD (Sistema de Referência do Dispositivo)**: Utilizam-se coordenadas de um dispositivo de saída específico. Em um visor de celular, pode indicar o número máximo de pixels ou uma de suas resoluções específicas de uma aplicação, por exemplo: (640x480), (320x480).

4.2 TRANSFORMAÇÃO E VISUALIZAÇÕES 2D

O mapeamento na janela de visualização acontece no sistema de referência do universo SRU e o SRD. Obedece a uma área na matriz de pixels da saída do dispositivo, podendo até corresponder a toda essa matriz, já que se permite ter várias viewports no dispositivo.

No plano cartesiano, como as suas coordenadas reais são infinitas, para que seja possível representar qualquer modelo deste espaço dentro da viewport, é necessário definir então um domínio no qual os objetos possam ser representados. Esse domínio é chamado de janela.

Essa janela tem praticamente o mesmo sentido de uma janela comum, porque é através dela que se vê a cena do mundo real ser representada digitalmente. Depois que esse espaço esteja disponível para o desenho gráfico, na saída do dispositivo deve-se especificar seus valores máximos e mínimos e, então, traduzir a área da janela escalada para a posição do visor.

As proporções relativas dos objetos são mantidas se os fatores de escala são os mesmos ($F_{sx} = F_{sy}$). Caso contrário, os objetos do mundo vão ser esticados ou comprimidos como mostra na Figura 4:

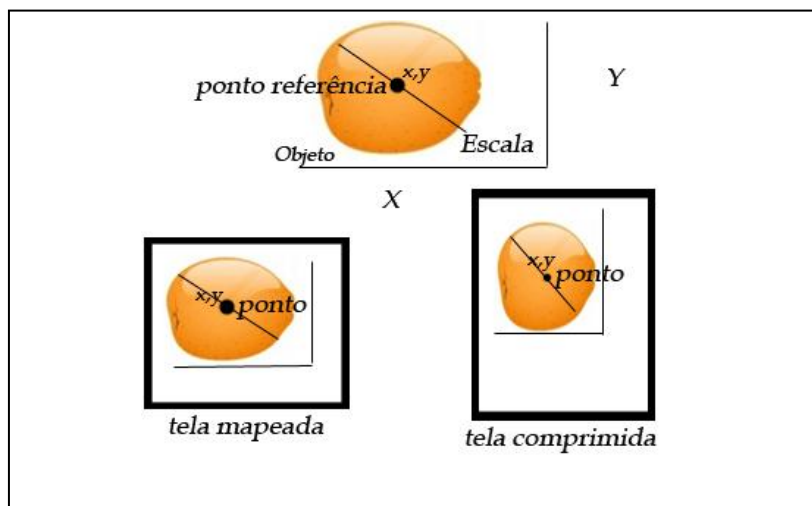


Figura 4 – Viewport com uma coordenada referencial
Fonte: Dados do pesquisador

Para manter as proporções de escala fiéis, deve-se usar um escalonamento com dois pontos. Em cada ponto dessa coordenada, X e Y recebem seus valores mínimos e máximos de sua representação na tela, sendo assim, Xmin, Xmax, Ymin e Ymax. Mesmo que o objeto não seja mostrado em toda tela de visualização, ocorre um processo de corte, chamado de *clipping*. Porém as suas proporções sempre serão mantidas, conforme verificado na Figura 5.

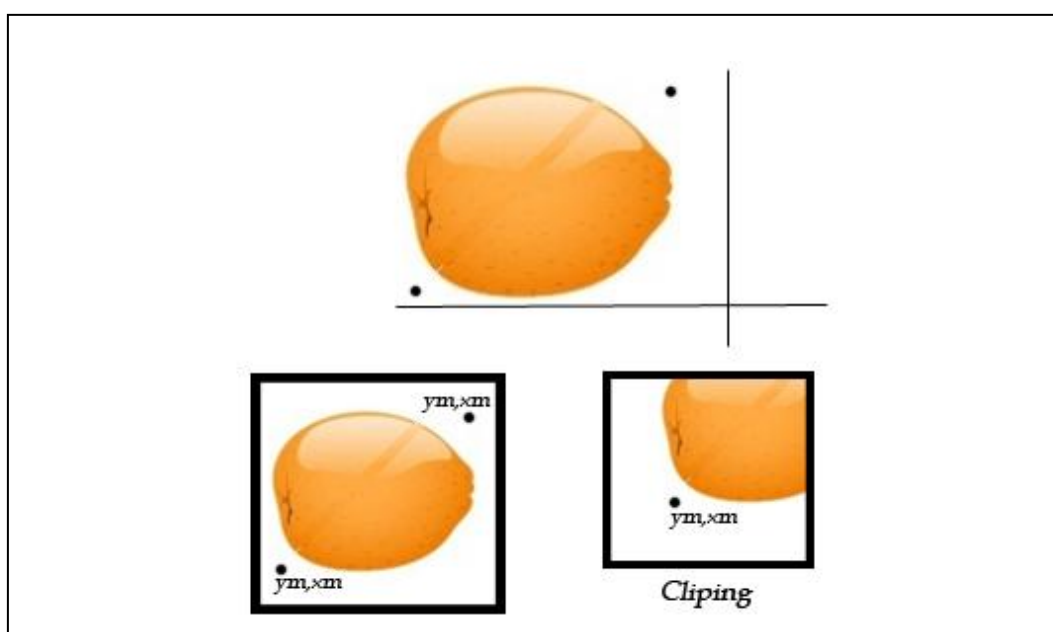


Figura 5 – Dois pontos referenciais o objeto acontecimento de clipping
Fonte: Dados do pesquisador

Sendo então Xmin e Xmax os valores mínimos e máximos do eixo

horizontal da janela de visualização representando o objeto real, e XV_{min} e XV_{max} os valores para o objeto na Viewport representando o objeto virtual, o mesmo acontece para o eixo vertical, o Y , tendo, então, Y_{min} e Y_{max} , YV_{min} e YV_{max} .

A transformação do objeto SRU para SRD se tem por cálculo de proporcionalidade. Então, a coordenada se estabelece da forma representada na Figura 6:

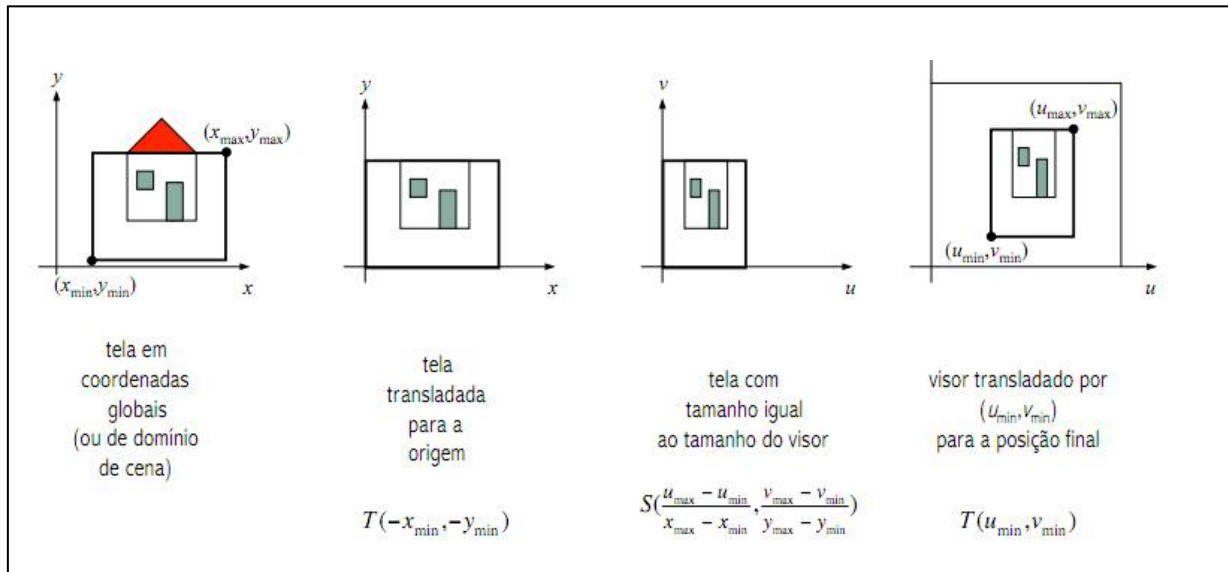


Figura 6 – Transformação do objeto em referência SRU para SRD

Fonte: Lee (2007)

Com os cálculos de proporcionalidade, é possível chegar à seguinte matriz para realização da visualização de pixel do dispositivo:

$$\frac{XV - XV_{MIN}}{X - X_{MIN}} = \frac{XV_{MAX} - XV_{MIN}}{X_{MAX} - X_{MIN}}$$

Resolvendo-se, então, o X em função do XV e o XV em função de X , se tem as relações:

$$XV = \underbrace{\frac{XV_{MAX} - XV_{MIN}}{X_{MAX} - X_{MIN}}}_{SX \text{ (fator de Escala)}} (X - X_{MIN}) + XV_{MIN}$$

$$\begin{cases} XV = SX(X - X_{MIN}) + XV_{MIN} \\ X = (XV - XV_{MIN}) / SX + X_{MIN} \end{cases}$$

Na equação, SX é o fator de escala na dimensão horizontal. Então, o cálculo para a coordenada do objeto em função de X é :

$$\text{Coordenada } X = \frac{(X - XVirtMin) * (XMostraMax - XMostraMin)}{XVirtMax - XVirtMin} + XMostraMin$$

Consequentemente, a analogia da coordenada para o eixo vertical em função de Y é a mesma, ficando da seguinte maneira:

$$\text{Coordenada } Y = \frac{(Y - YVirtMin) * (YMostraMax - YMostraMin)}{YVirtMax - YVirtMin} + YMostraMin$$

Com essa fórmula, calculam-se os valores de coordenadas de certo ponto e os transporta do mundo real para a janela de visualização do Viewport.

Para fazer o teste da fórmula, observa-se o seguinte exemplo da Figura 7.

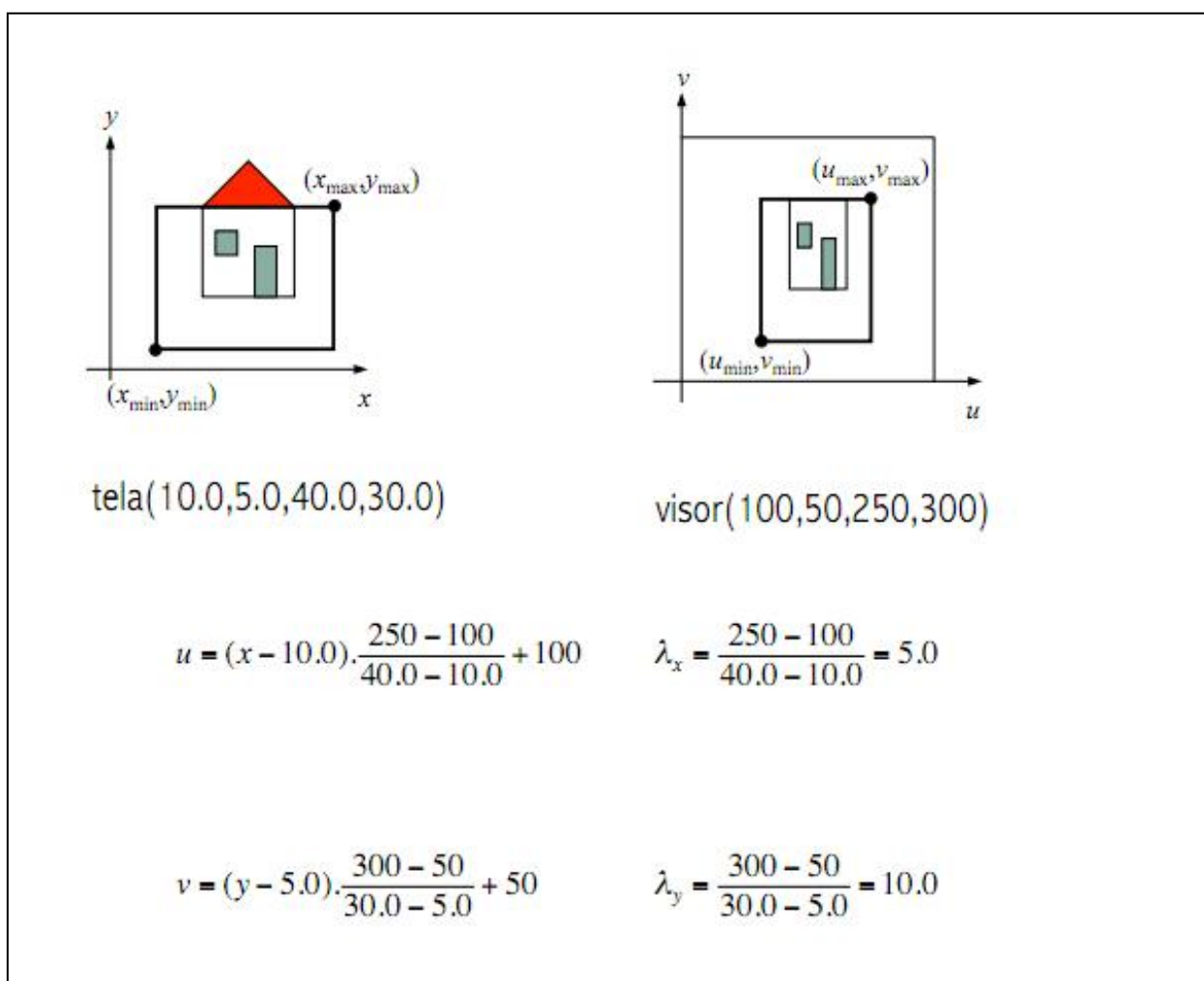


Figura 7 – Exemplo e constatação do Viewport
Fonte: Lee (2007)

4.2.1 Coordenadas cartesianas e coordenadas de monitores

O sistema cartesiano de duas dimensões consiste em uma origem e dois eixos perpendiculares entre si. A intersecção de ambos é chamada de ponto de origem. Cada ponto no plano contém uma coordenada única, que é identificada por dois números. O primeiro representa a distância horizontal a partir do eixo vertical, e o segundo representa a distância vertical a partir do eixo horizontal. A sua dimensão horizontal é acrescida da esquerda para a direita, já sobre o eixo vertical o sistema é ascendente de sua origem (BLINN; NEWEL, 1978), conforme verifica-se na Figura 8.

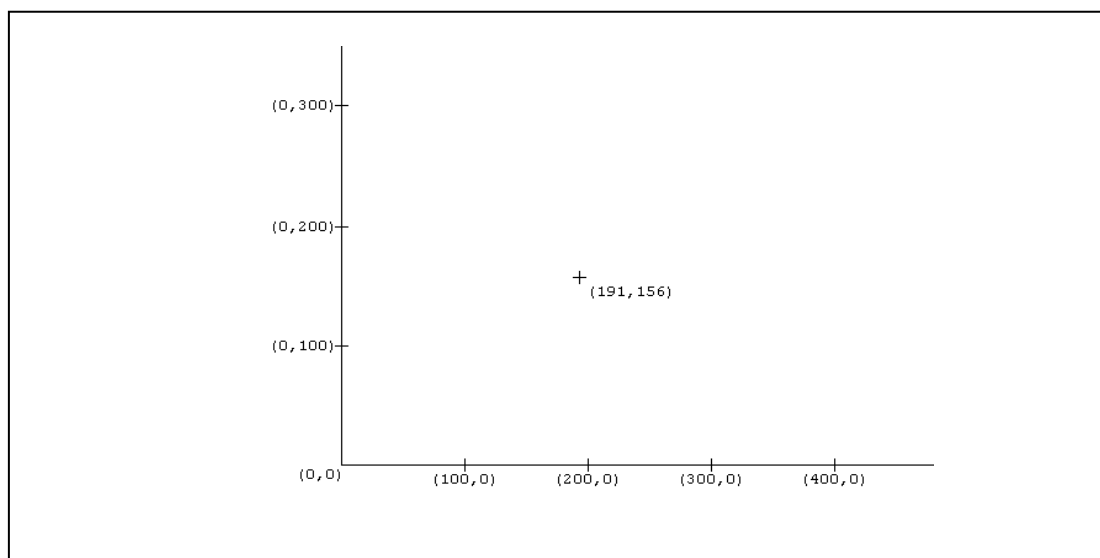


Figura 8 – Representação de coordenadas do plano cartesiano

Fonte: Blin; Newel, 1978

Os monitores usam um sistema de coordenadas que também utiliza uma origem e dois eixos perpendiculares, igual ao plano cartesiano. Entretanto, a diferença entre os dois é que, ao invés do eixo vertical ser ascendente, ele é tipicamente descendente. Ou seja, sua escala cresce de cima para baixo. Então, para a representação de imagens nos monitores deve haver a conversão entre os sistemas.

Neste trabalho, para a elaboração de uma imagem, um LED infravermelho representará o ponteiro de um mouse em um editor de imagem, cuja conversão de coordenadas será essencial (BLINN;NEWEL, 1978), de acordo com o representado na Figura 9.

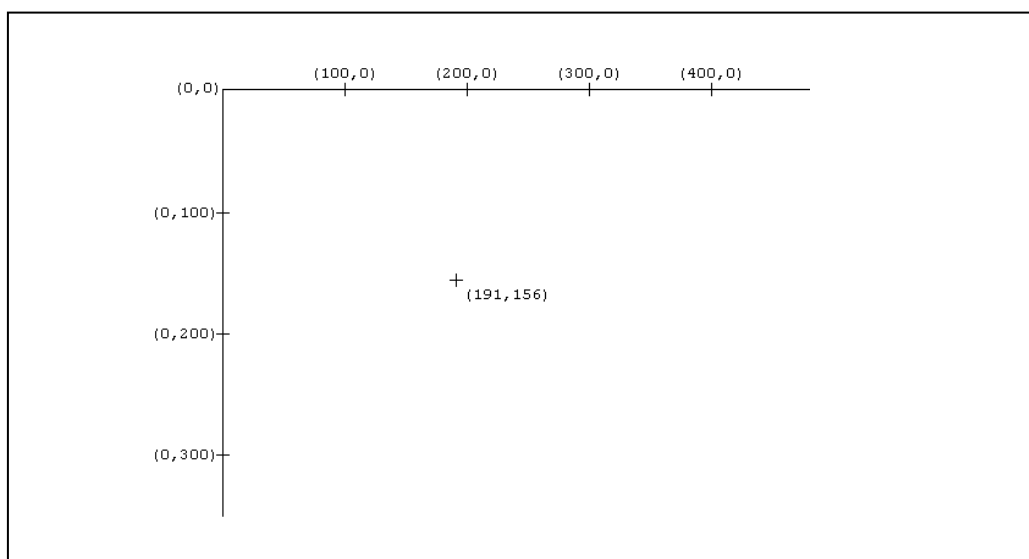


Figura 9 – Representação do sistema de coordenadas dos monitores
Fonte: Blin; Newel, 1978

5 JAVA

Para a realização do protótipo do *software*, é necessário utilizar uma linguagem de programação. Neste capítulo é apresentado um breve resumo da linguagem JAVA e suas vantagens para o projeto, bem como discorre-se sobre as bibliotecas e as API - *Application Programming Interface*, que serão necessárias para que seja possível a criação do protótipo.

Java é uma linguagem de programação de alto nível, com abrangência para todas as plataformas, desde que haja a Java Virtual Machine (JVM) instalada, com sintaxe de código semelhante à linguagem C++. Possui diversas características peculiares legadas de outras linguagens, tais como Smalltalk e Modula-3. É portada de simplicidade, fortemente tipada, segura, forte, bem estruturada e distribuída (DEITEL, 2003).

Nesse sentido, sabe-se que desempenho é algo que sempre deve ser levado em consideração, ainda mais em se tratando de processamento em tempo real.

O interpretador da linguagem tem certas otimizações que admitem a execução mais eficiente de código Java. Para situações nas quais aplicações precisam de um poder computacional ainda maior com mais desempenho, algumas sessões do código podem ser escritas em código nativo como é o caso da linguagem C.

5.1 BIBLIOTECAS E APIS

Java fornece uma vasta coleção de bibliotecas e de classes testadas e que dão suporte a inúmeras funcionalidades. Estas bibliotecas são facilmente estendidas ao programa, tornando ele ajustável conforme as necessidades da aplicação em particular, por intermédio de herança, assim como muitas *Application Programming Interface* (API).

APIs são pacotes pré-compilados ou compilados que são distribuídas pelas comunidades ou organizações, para facilitar o uso de certas funcionalidades Java (LEMAY; CADENHEAD, 1999).

5.1.1 Wiiusej

WiiuseJ é uma API Java para usar Wiimotes em computadores. Esta API foi construída com base na biblioteca Wiiuse, usando *Java Native Interface* (JNI).

Com essa biblioteca, é possível obter todas as funcionalidades de um controle wiiremote. Possui facilidades de uso, é multiplataforma, concilia todos os eventos do acelerômetro, da câmera rastreadora infravermelha, botões, vibrações, enfim, todos os recursos disponíveis da tecnologia.

Por ser de natureza livre, muitos desenvolvedores constroem novas melhorias e atualizações, tornando o desempenho de leitura e rastreamento do infravermelho cada vez melhores. Todo o código fonte e documentação, junto ao download da API, pode ser encontrada no site do desenvolvedor: <<http://code.google.com/p/wiiusej/>>.

Na Figura 10, encontra-se representada a interface gráfica API WiiuseJ.

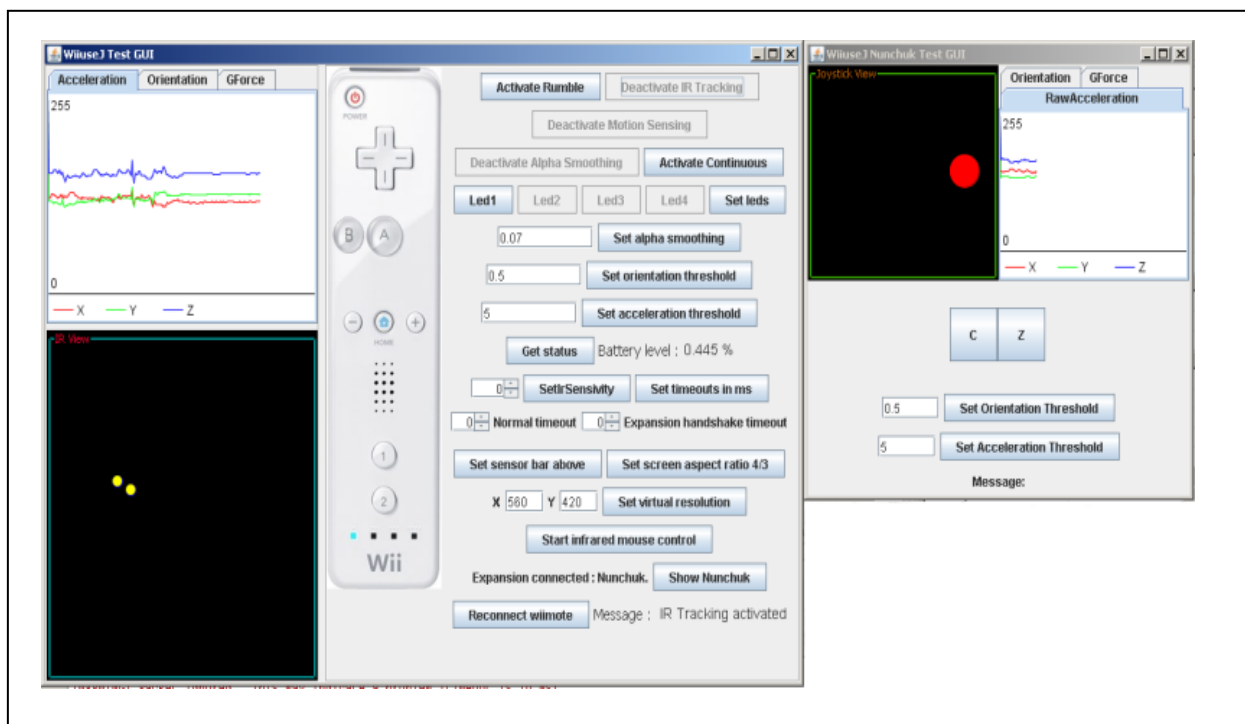


Figura 10 – Interface gráfica API WiiuseJ

Fonte: Dados do pesquisador

5.1.2 MoteJ

MoteJ é uma API desenvolvida em Java, que possibilita usar os recursos do controle wiimote, controle do console Nintendo Wii.

Consiste-se em dois pacotes principais: a biblioteca motej e a motej extras. Enquanto as extras ainda que dependam da biblioteca para funcionar, ela traz melhorias no desempenho com transmissão de dados via Bluetooth, que apenas a biblioteca não possui. Suas principais funcionalidades são:

- Reconhecer e conectar mais de um dispositivo;
- Wiimote Câmera de fácil edição;
- Consegue abranger os recursos proporcionados pelo acelerômetro;
- Vibração do controle;
- Botões de Wiimote;
- Ler a memória EEPROM;
- Escrever registros Wiimote;
- Informação sobre a angulação;
- Obter os dados da calibração.

Uma das vantagens da motej é sua facilidade de lidar com a câmera infravermelha, uma vez que todo o cálculo de rastreamento é feito pelo próprio controle e armazenado pela biblioteca. A única tarefa que o programador se reserva a fazer, é mapear o cenário de aplicação, como mostra a Figura 11.

```
IrCameraListener listener = new IrCameraListener() {  
    public void irImageChanged(IrCameraEvent evt) {  
        System.out.println(evt.getX() + " : " +  
            evt.getY());  
    }  
};  
  
Mote mote =  
MoteFinder.getMoteFinder().findMote();  
mote.addIrCameraListener(listener);  
mote.enableIrCamera();  
  
mote.setReportMode(ReportModeRequest.DATA_REPORT_0x36);
```

Figura 11 – Trecho de código

Fonte: Dados do pesquisador

6 TRABALHOS CORRELATOS

Muitos dos trabalhos que foram obtidos para servir como base teórica e científica são de origem de publicações em workshops, criados pela IEEE (organização voltada à padronização e pesquisas do mundo todo), e outros trabalhos, como teses de doutorados e outras pesquisas comerciais, sobre os quais discorre-se a seguir.

6.1 HACKING THE NINTENDO WII REMOTE

Este trabalho revela possíveis aplicações para o controle de Nintendo Wii, sem ser seu uso para o console (LEE, 2000).

Algumas das aplicações construídas por ele são:

- a) Reconhecimento de gestos;
- b) Rastreamento de objetos;
- c) Monitoramento de movimentos 3D;
- d) Realidade aumentada.

O trabalho em questão discute o WIIREMOTE com detalhes técnicos.

6.2 RASTREAMENTO VISUAL DE OBJETOS UTILIZANDO MÉTODOS DE SIMILARIDADE DE REGIÕES E FILTRAGEM ESTOCÁSTICA

É uma tese de doutorado produzida por Flávio de Barros Vidal, na Universidade Brasília.

A tese apresenta diversas formas de rastreamento de objetos, definindo-as e demonstrando suas aplicabilidades e fórmulas de reconhecimento.

Mostra também variadas formas de filtragem de dados e ainda possíveis otimizações para algoritmos que usam essas filtragens.

7 PROTOTIPAÇÃO DO SOFTWARE RECEPTOR E INTERPRETADOR DOS DADOS DE COORDENADAS

É proposto o desenvolvimento de um *software* linguagem Java, capaz de integrar um dispositivo sensorial de rastreamento infravermelho, que seja apto a interpretar valores de calibração e rastreamento providos pelo sensor, por intermédio de comunicação via Bluetooth, com o foco de prover um protótipo de uma LDI de baixo custo, utilizando-se destes recursos.

7.1 METODOLOGIA

O levantamento bibliográfico referente à pesquisa proporcionou um vasto conhecimento dos conceitos científicos teóricos e práticos, uma vez que grande parte das referências bibliográficas possuem suas respectivas publicações na IEEE. Esse estudo, referente ao rastreamento de objetos, objetivou a escolha de um método dos mencionados anteriormente, que melhor se adaptaria às condições envolvendo *hardwares* e *softwares* ao alcance de todos.

Como visto, também foi feita uma breve análise dos *hardwares* capazes de servirem de sensor para o rastreamento de objetos. Foi verificado que há dois sensores mais comuns e que podem ser encontrados com facilidade: uma câmera digital de vídeo normal e um rastreador infravermelho, disponível em um controle de um game console, Nintendo Wii.

Depois de analisado o *hardware*, fez-se necessário optar por uma linguagem de programação para criação de um *software* capaz de interpretar os dados recebidos pelo *hardware* e, assim, conciliar ambos para criação da lousa digital. A linguagem de programação foi o Java, uma linguagem cuja licença de uso é livre e que vem sendo vastamente utilizada, sem mencionar a gama de APIs livres disponíveis na internet, que se comunicam com um dos sensores mencionados, ou seja, a câmera infravermelho.

7.2 O PROTÓTIPO

Para o desenvolvimento da LDI, como proposto por esta pesquisa, são necessários alguns recursos de *hardware* e *software*, sendo assim, é necessário o uso de um *hardware* capaz de visualizar o objeto a ser rastreado e repassar os dados ao computador.

A escolha deu-se para o uso do *hardware* mencionado, o Wiimote, o controle de um console de videogame, capaz de rastrear leds infravermelhos em tempo hábil para o processamento dos dados obtidos. Como recurso de *hardware* também é imprescindível o uso de um dispositivo para projeção da tela da lousa digital, podendo ser qualquer projetor de quaisquer marca disponível.

O uso de um *software* intermediador da informação provida pelo *hardware* sensor é essencial para o bom funcionamento do protótipo. Como já citado, optou-se pelo Java, uma linguagem totalmente gratuita, com várias APIs e recursos necessários para implementação.

Desse modo, a API escolhida para criação do protótipo foi a WIUSEJ por ter diversas facilidades e diversos recursos já disponíveis que fará com que o programador preocupe-se apenas com a obtenção dos dados de coordenadas para fazer a calibração dos dados.

7.3 DISPOSIÇÃO DO HARDWARE

Para que não haja problemas com interferências ou o mau uso dos recursos fazendo ineficaz o rastreamento do LED infravermelho, deve-se tomar algumas medidas para o bom funcionamento da lousa digital. A começar com a disposição dos equipamentos.

Como mencionado, o uso do projetor é necessário para criação da lousa virtual, sendo assim, para as projeções ocorrerem de forma satisfatória, o equipamento deve ser posto cerca de 3 a 5m de distância do quadro digital, como recomendado pelas fábricas dos equipamentos. Isso faz com que se obtenha uma projeção média de 180 polegadas. Deve-se ter em mãos um LED infravermelho para a emissão de luz. Aconselha-se a usar um recipiente com forma de caneta bastão comum em que a ponta seja o LED, para facilitar o manuseio e a interação por parte dos professores

com a LDI.

A Figura 12 representa o esquema de caneta infravermelho.

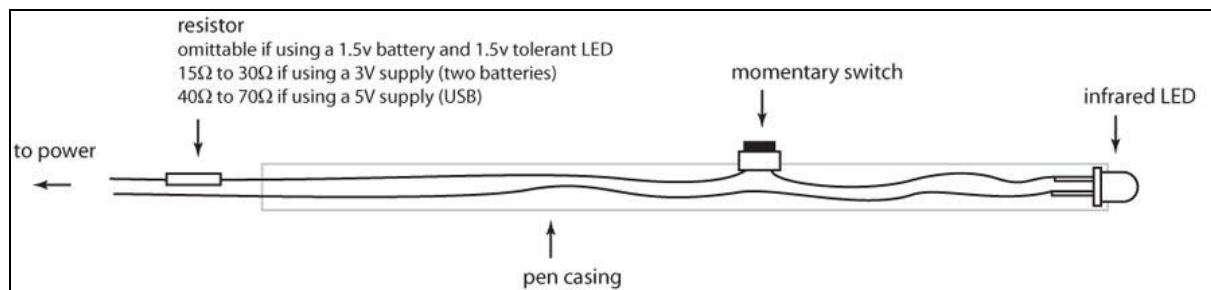


Figura 12 – Esquema de caneta infravermelho

Fonte: Lee (2007)

O controle de Nintendo Wii deve estar a uma distância equivalente ao projetor, com uma angulação entre a ponta inferior de qualquer um dos lados da projeção entre 30° e 50° para que o uso do quadro não interfira no contato visual entre a caneta infravermelha e a câmera rastreadora de objetos.

Seguindo todas essas recomendações, o cenário deve ficar parecido com a Figura 13, tendo o projetor multimídia (a), a lousa projetada por ele (b), a caneta emitindo luz infravermelho (c) para a câmera rastreadora (d) que emite por Bluetooth para o computador (e).

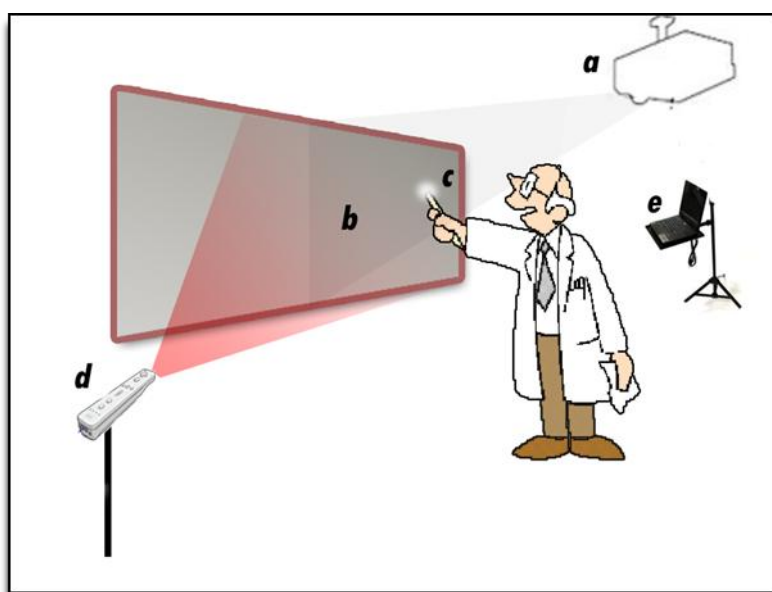


Figura 13 – Ilustração do cenário

Fonte: Dados do pesquisador

7.4 ENTENDENDO A API WIIUSEJ

Como mencionado anteriormente, as API que poderiam ser utilizadas são a WiiuseJ e a WiimoteJ, optou-se então pela WiiuseJ versão 0.12b. Essa é uma API robusta com vários recursos disponíveis. Sua interface é amigável e bem completa. Entretanto, seu idioma nativo é o inglês, mas pode ser reescrito em outro idioma, uma vez que o código fonte possui licença de uso livre no site do desenvolvedor.

Como visto na Figura 10, na pagina 35, existem botões capazes de ativar e desativar todos os recursos que o controle de Nintendo Wii oferece aos usuários, desde os mais simples, tais como os LED, a ativação da vibração, a visualização do nível de bateria do aparelho, algo muito relevante, visto que o dispositivo em meio aos testes apresentou baixo desempenho quando o nível de energia se apresentou inferior a 30%.

A aplicação permite também usufruir das partes mais robustas do controle que são: medidor de força gravitacional, a orientação do controle, a medição de aceleração e o mais importante para o projeto: o rastreador infravermelho, que pode ser ativado e desativado facilmente, apertando no botão "Active IR Tracking".

A API traz um recurso muito útil e que será usado para criação do *software*, que é controlar o mouse do computador usando o controle de Nintendo Wii, apenas apertando no botão "*Start Infrared Mouse Control*".

Entretanto, o emissor infravermelho deve permanecer fixo para que o controle se mova, e o trabalho propõem que o emissor infravermelho deva se mover para emular uma espécie de caneta para o utilizador usar no quadro digital.

A Figura 13 representa o Modelo da funcionalidade da API.

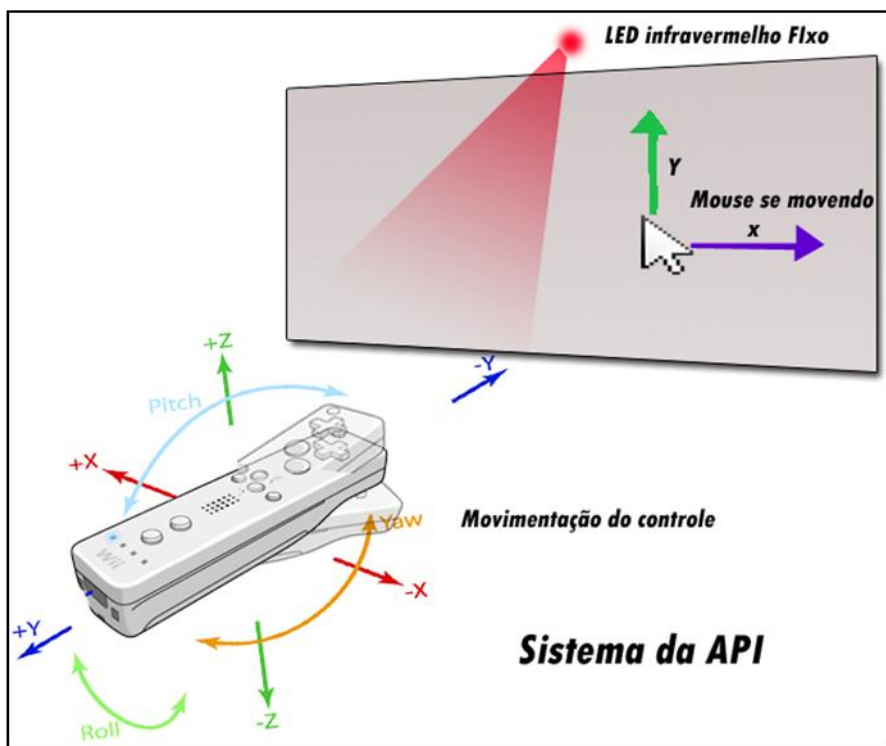


Figura 14 – Modelo da funcionalidade da API
Fonte: Dados do pesquisador

Como pode ser visto na Figura 14, o modelo de como a API se comporta em relação à funcionalidade. Não há como usar essa funcionalidade como lousa digital, pois deve-se observar a diferença do sistema proposto na Figura 15, a seguir:

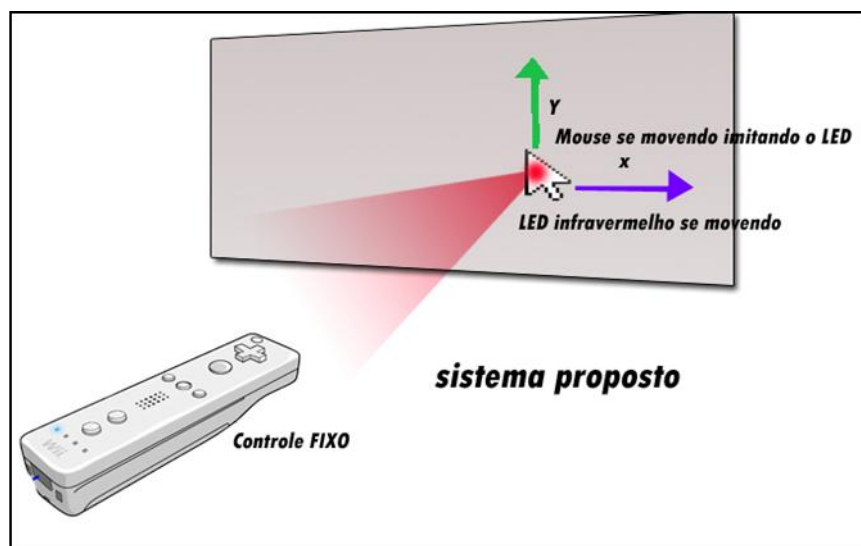


Figura 15 – Modelo para sistema proposto
Fonte: Dados do pesquisador

Note-se que o que nesse modelo, quem se move é o LED perto da lousa, fazendo com que se pareça uma funcionalidade TouchScreen para o quadro, o que o

torna um quadro digital iterativo.

Para que se obtenha o resultado proposto é necessário fazer edições no programa ou a criação de um novo, usando a API como biblioteca. Porém, para que não haja dependências de bibliotecas e a criação de novas classes e outros métodos que possam, mesmo que pouco, tirar a facilidade e desempenho do projeto, preferiu-se por fazer edições na própria API.

O código fonte da WiiuseJ é composto da seguinte forma: há 8 pacotes separados por praticamente 3 grupos: recursos físicos do controle, recursos da API e utilitários. Com a API, pode-se conectar o controle ao computador. Basta instanciar um novo objeto Wiimote. Assim, se consegue acessar todos os métodos para ativar e desativar os recursos.

Na Figura 16, a seguir, verifica-se as partes do código da API, na qual demonstra-se pelo método main, o Wiimote sendo criado e iniciando a interface gráfica da API. Depois, é criado um construtor que inicia assim todos os *listeners* que criam um wiimote padrão.

```

1. //Classe Main, instanciando objeto Wiimote e iniciando a API.
2. //-----//
3. Wiimote[] wiimotes = WiiUseApiManager.getWiimotes(1, true);
4. WiiuseJGuiTest gui = null;
5. if (wiimotes.length > 0) {
6.     gui = new WiiuseJGuiTest(wiimotes[0]);
7. }
8. //-----//
9. //Criando a parte gráfica da API com os componentes e iniciando os listeners e as opções padrões do controle
10. //-----//
11. public WiiuseJGuiTest(Wiimote wiimote) {
12.     initComponents();
13.     this.addWindowListener(new CloseGuiTestCleanly());
14.     if (wiimote != null) {
15.         this.wiimote = wiimote;
16.         registerListeners();
17.         initWiimote();
18.         isFirstStatusGot = false;
19.         getStatusButtonMousePressed(null);
20.     }
21. }
22. //-----//
23. //Registrando os Listeners
24. //-----//
25. private void registerListeners() {
26.     wiimote.addWiiMoteEventListeners((IRPanel) irViewPanel);
27.     wiimote.addWiiMoteEventListeners((ButtonsEventPanel) buttonsPanel);
28.     wiimote.addWiiMoteEventListeners((OrientationPanel) motionSensingPanel);
29.     wiimote.addWiiMoteEventListeners((GForcePanel) gForcePanel);
30.     wiimote.addWiiMoteEventListeners((AccelerationPanel) accelerationPanel);
31.     wiimote.addWiiMoteEventListeners(this);
32. }
33. }

```

Figura 16 – Trechos de códigos da API WIIUSEJ

Fonte: Dados do pesquisador

7.4.1 Ir Tracking

Como apresentado na Figura 17, na lista dos *listeners* implementados, há o chamado o IRPanel. Nele, acontece a demonstração de como a câmera interpreta o infravermelho e o representando neste painel. Ela é a classe na qual é invocado o método OnIReventn, nele é acionado o IREvent.

```

1. public void onIrEvent(IRevent arg0) {
2.     // transfer points
3.     wiiusej.values.IRSource[] points = arg0.getIRPoints();
4.     nbPoints = points.length;
5.     for (int i = 0; i < points.length; i++) {
6.         xCoordinates[i] = (int) points[i].getRx();
7.         yCoordinates[i] = (int) points[i].getRy();
8.     }
9.     for (int i = points.length; i < MAX_NB_POINTS; i++) {
10.        xCoordinates[i] = -1;
11.        yCoordinates[i] = -1;
12.    }
13.
14.    // redraw panel
15.    repaint();
16. }

```

Figura 17 – Trecho do código IREvent

Fonte: Dados do pesquisador

Toda vez que ocorre um evento infravermelho, ou seja, quando há alguma mudança no último ponto rastreado, a classe IREvent é acionada. Ao acioná-la, ela automaticamente comunica à classe IRSource e se abastece com os valores das coordenadas enviadas do controle como mostra o código representado na Figura 18.

```

1. public IRSource[] getIRPoints() {
2.     IRSource[] ir = new IRSource[indexPoints];
3.     System.arraycopy(IRPoints, 0, ir, 0, indexPoints);
4.     return ir;
5. }
6.

```

Figura 18 – Classe IREventchmandoIRSource

Fonte: Dados do pesquisador

O sistema de rastreamento infravermelho na API se dá por basicamente 3 métodos. A fonte do rastreamento, que se chama IRSource, que é onde se trata da recepção dos dados pré-processados pelo controle usando métodos Sets e Gets, deixando disponível para aplicação os dados de coordenadas, de acordo como

demonstrado na Figura 19, que representa o construtor da classe e o método *toString*:

```

1. public IRSource(int xx, int yy, short rxx, short ryy, short si) {
2.     x = xx;
3.     y = yy;
4.     rx = rxx;
5.     ry = ryy;
6.     size = si;
7. }
8.
9. @Override
10. public String toString() {
11.     return "Interpolated coordinates (" + x + "," + y
12.         + "), Raw coordinates(" + rx + "," + ry + "), source size : "
13.         + size + " ";

```

Figura 19 – Construtor e método ToString da classe IRSource

Fonte: Dados do pesquisador

Os parâmetros *X* e *Y* são as coordenadas interpoladas; *rx* e *ry* são as coordenadas da câmera infravermelha, assumindo valores entre 0-1023, ou seja, 1024px e o parâmetro *size* é a medida do ponto infravermelho, que varia entre 0-15 totalizando 16px.

Como visto em fundamentação teórica, o reconhecimento do objeto rastreado se dá por pontos e o seu rastreamento por fluxo óptico, ou seja, por quantidade de luz. Nessa parte do código, ele considera o objeto com um ponto medindo seu tamanho em pixels. Esse tamanho refere-se ao centro principal de emissão de luz, pois o rastreamento se dá por quantidade de luminosidade entre os sucessivos quadros feitos pela câmera rastreadora.

A seguir, a Figura 20 demonstra um feixe luminoso produzido por LED infravermelho, no qual o círculo vermelho mostra a área máxima capaz da luz, e o ponto no meio de 4px. Assim, apenas o movimento do ponto seria rastreado e seriam mostradas as coordenadas.



Figura 20 – Representação do objeto luz

Fonte: Ballard (2001)

7.5 EDIÇÕES NA API

Uma vez que foi discutida a parte de rastreamento infravermelho, deve-se então começar a implementação do protótipo. Para isso, deve-se explorar a classe *WiiuseGuiTest*, que é a classe onde é criado o *Form* com os componentes da tela inicial da interface. É nela que serão feitas as alterações.

Toda vez que é acionado o botão "*Start IR Mouse Control*", a variável do tipo *robot*, que é a biblioteca responsável por utilizar os recursos do mouse do computador, é instanciada. Com isso, todo evento disparado como um *IREvent* faz um teste para ver se a variável *robot* foi iniciada. Em caso positivo, ele chama o método da classe *MouseMove(x,y)*, onde *x* e *y* são as coordenadas de para onde o mouse deve-se mover, como mostrado no código representado na Figura 21.

```

1.     public void onIrEvent(IREvent arg0) {
2.         if (robot != null) { // if mouse control activated
3.             robot.mouseMove(arg0.getX(), arg0.getY());
4.         }
5.         if (statusIRRequested) {
6.             xResolutionTextField.setText("" + arg0.getXRes());
7.             yResolutionTextField.setText("" + arg0.getYRes());
8.             statusIRRequested = false;
9.         }
10.    }

```

Figura 21 – Trecho do código que mostra o evento de IR e movimentar o mouse

Fonte: Dados do pesquisador

Fez-se uso da utilização deste recurso, mas observou-se qual conjunto de coordenadas está chegando nessa classe. Com base nessas coordenadas, é que se pode fundamentar o código para que o mouse possa se movimentar de acordo com a lousa digital. Para tal, foram introduzidos dois comandos dentro da condicional da variável *robot*, que seriam: `System.out.println("X"+arg0.getX());` e a `System.out.println("Y"+arg0.getY());`. Fazendo isso obtém o resultado demonstrado na Figura 22:

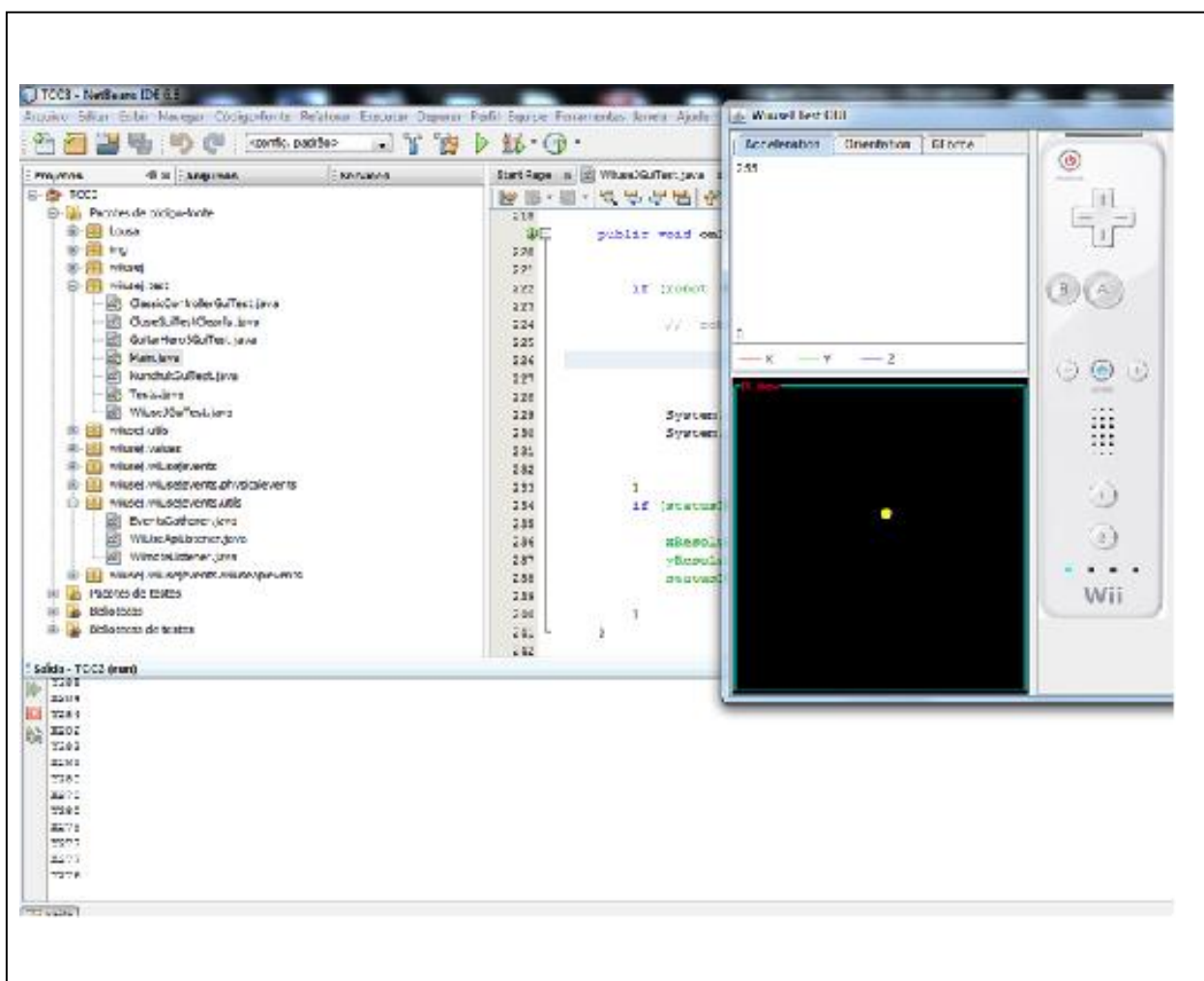


Figura 22 – Tela capturada do teste com coordenadas

Fonte: Dados do pesquisador

Pode-se perceber que na saída de informações da IDE, neste caso o Netbeans 6.8, as coordenadas X e Y estão envolvidas no argumento *arg0*. Essas coordenadas representam o ponto no painel virtual, sendo assim não seria possível a utilização delas para manusear o computador de forma precisa.

Deve-se, então, mapear a área virtual criada pelo controle para que cada coordenada do plano virtual torne-se uma coordenada física da lousa digital. Para

tanto, deve-se usar uma técnica da área de computação gráfica chamada ViewPort.

7.5.1 Utilização de Viweport

Para utilizar as técnicas de computação gráfica, na área de mapeamento, usou-se a técnica de *viewport*, que através de cálculos matemáticos, funções sobre planos cartesianos, mapeamento de pixels e etc., permite representar objetos de um determinado universo para o plano do computador, ou seja, sua representação na tela.

Para criação do protótipo, deve-se mapear a coordenada do LED do espaço físico e representá-los na tela do computador. Sabe-se que o computador tem sua resolução em diferentes monitores, diferentes sistemas, diferentes padrões de resolução e etc. Neste protótipo, utilizou-se da resolução 1024x768, que é a mais comum utilizada pelos projetores e que é a resolução da câmera rastreadora infravermelho.

A API demonstra o ponto do LED através de um painel chamado IRPANEL. O IRPANEL cria uma imagem virtual representativa do ponto rastreado. Nesse painel, tem-se duas coordenadas, a coordenada calculada que cria um plano cartesiano em meio à representação, e o valor da coordenada absoluta, ou seja, 1024x768px, a coordenada absoluta nativa da câmera rastreadora, que é a que será usada para fazer o mapeamento.

Precisa-se, então, abstrair os valores, as nomenclaturas, os tamanhos entre as janelas para que possa ser colocado na fórmula, conforme pode-se verificar na Figura 23.

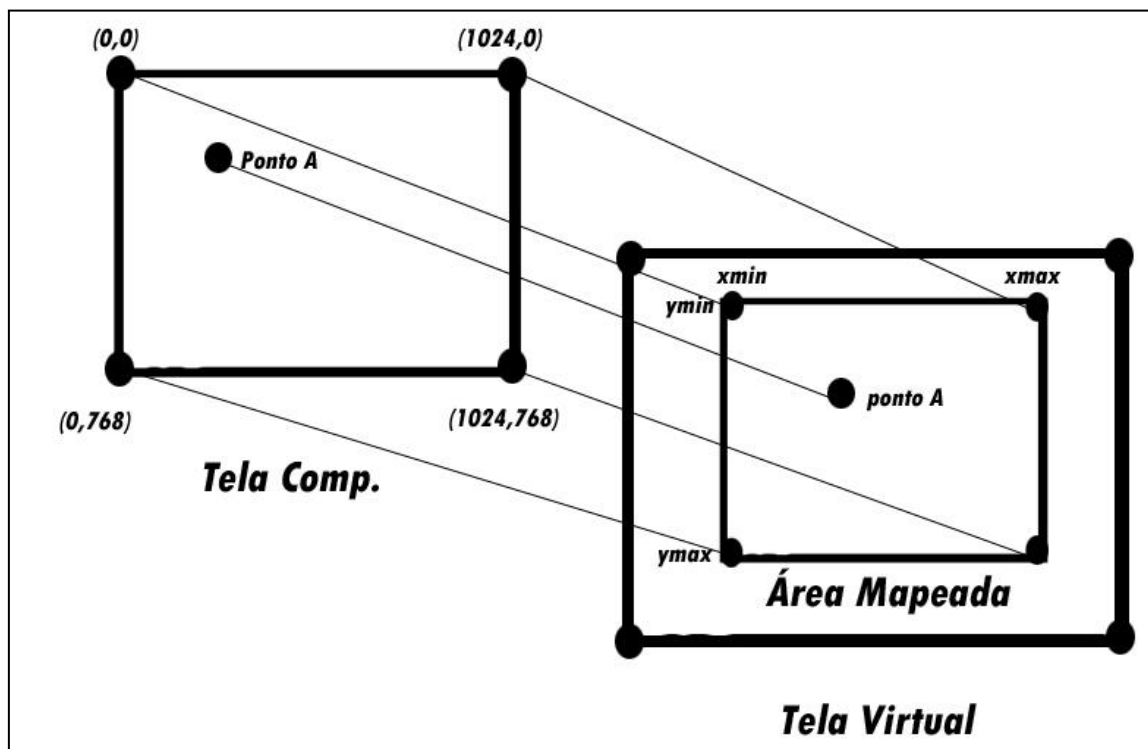


Figura 23 – Modelo de Viewport sobre o protótipo
Fonte: Dados do pesquisador

Deve-se calcular as coordenadas de X e de Y para a tela a fim de fazer com que cada ponto rastreado da caneta seja um ponto de pixel da tela no computador responsável por movimentar o mouse. A fórmula, como visto no capítulo sobre viewport, é a seguinte:

$$\text{Coordenada } X = \frac{(X - XVirtMin) * (XMostraMax - XMostraMin)}{XVirtMax - XVirtMin} + XMostraMin$$

$$\text{Coordenada } Y = \frac{(Y - YVirtMin) * (YMostraMax - YMostraMin)}{YVirtMax - YVirtMin} + YMostraMin$$

Onde:

X é a coordenada real a ser calculada para coordenada virtual;

XVirtMin é a ponta do IRPainel de menor valor;

XVirtMax é a ponta do IRPainel de maior valor;

XMostraMax sendo o maior valor da tela, caso seja a tela inteira tem-se 1024;

XMostraMin menor valor da tela, caso seja tela inteira tem-se 0;

Y coordenada real a ser calculada para coordenada virtual;

YVirtMin é a ponta do IRPainel de menor valor;

YVirtMax é a ponta do IRPainel de maior valor;

YMostraMax sendo o maior valor da tela, caso seja a tela inteira tem-se 768;

YMostraMin menor valor da tela, caso seja tela inteira tem-se 0.

Entretanto, para se obter esses valores, deve-se calibrar as coordenadas que devem ser obtidas rastreando as pontas extremas da tela de projeção e editando no código fonte, a representação demonstrada na Figura 24:

```
1. private int xvfinal = 153, xvinicial = 825, yvinicial = 760, yvfinal = 310, cordy=0, cordx=0;
2.
```

Figura 24 – Pontos calibrados

Fonte: Dados do pesquisador

Logo depois de calibradas as coordenadas, deve-se alterar o código com a expressão vista acima.

```
3. if (robot != null) { // if mouse control activated
4.
5.     cordx = (int) (((arg0.getAx() - xvfinal) * 1024) / (xvinicial-xvfinal));
6.     cordy = (int) (((arg0.getAy() - yvfinal) * 768) / (yvinicial-yvfinal));
7.
8.     robot.mouseMove(((cordx-1024)*-1), ((cordy-768)*-1));
```

Figura 25 – Código fonte aplicado à fórmula de Viewport

Fonte: Dados do pesquisador

Note-se que quando posto na função robot, que é a biblioteca que possibilita mover o mouse, é subtraído o *MostraMax* é multiplicado por 1 negativo, pois as coordenadas entre a imagem virtual e a real são opostas. Em IRPanel é (0,0) na representação real tem-se (1024,768), onde IRPanel é (0,768) em representação real é (1024,0) e assim sucessivamente.

7.6 TESTES E OBSERVAÇÕES

O protótipo desenvolvido possibilitou alcançar todos os objetivos propostos. O *software* conecta a câmera rastreadora, recebe os dados via Bluetooth, interpreta os

dados de coordenadas e, usando de técnicas de mapeamento gráfico, transforma o sistema em uma LDI.

Constatou-se uma pequena distorção com margem de erro entre 2 e 4 pixels por conta da posição do rastreador e do jeito que é manuseada a caneta. Outro pequeno erro foi percebido quanto á conexão com o controle de wii. Isso porque, toda vez que precisa executar o programa, é necessário remover o controle da lista dos dispositivos de Bluetooth, para então adicioná-lo novamente e logo em seguida executar o programa.

Para melhor funcionamento, constatou-se a necessidade de refinar as expressões e métodos de viewport. Este refinamento encontra-se em processo.

O *software* resultante pode ser utilizado tanto em SO Linux quanto em Windows. Por ser feito em Java, ele possui característica multiplataforma. Entretanto há 4 arquivos que sempre devem acompanhar o *software* em sua pasta raiz, sendo que o *libwiiuse.so* e *libWiiuseJ.so* responsáveis para o funcionamento no Linux e *wiiuse.dll*, *WiiUseJ.dll* para funcionamento no Windows.

7.6.1 Instalação na Escola de Ensino Fundamental São Cristóvão

Em 27 de Abril de 2011, deu-se o primeiro teste do protótipo em uma Escola Estadual do município de Criciúma: a Escola de Ensino Fundamental São Cristóvão , que teve suas aulas de informática com o uso do recurso da LDI.

A professora da escola, em regime de contratação temporária, Fabiana Milanez, teve ao seu dispor o recurso. A docente fez uso da LDI em suas aulas de ensino em informática e tecnologia para os alunos da 3ª série do Ensino Fundamental. Fabiana comentou que os alunos se comportaram de forma diferenciada, menos dispersos e demonstraram um maior nível de interesse e atenção na aula, por se tratar de um atrativo maior pela quantidade de informação que se pode trazer para dentro da sala. A professora também afirmou que os alunos tiveram maior aproveitamento em sala de aula e que usará mais vezes o recurso.

A diretora da Unidade Escolar informou que não somente os alunos, mas também o corpo docente se mostrou entusiasmado com a novidade e espera equipar todas as salas com a LDI.



Figura 26 – Aula de informática com a utilização do protótipo
Fonte: Dados do pesquisador

Nesse mesmo dia, o colégio foi surpreendido com a visita do Secretário da Educação do Estado de Santa Catarina, Professor Marco Tebaldi. Em sua visita, o Secretário da Educação aproveitou para testar o protótipo, assinando o protótipo como ilustra a figura a seguir:

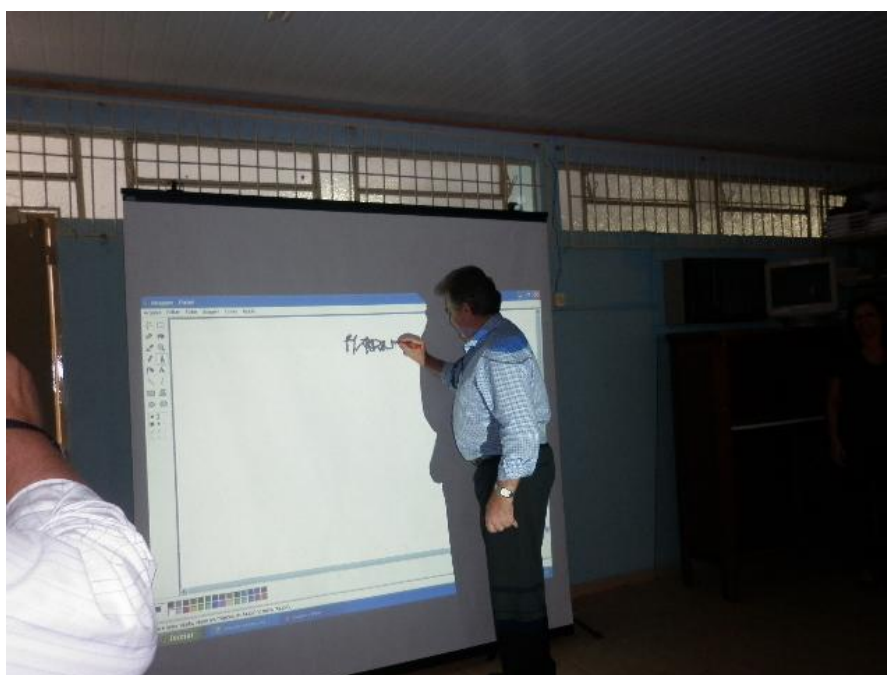


Figura 27 – Secretário da Educação de Santa Catarina usando o protótipo
Fonte: Dados do pesquisador

O secretário e sua equipe de assessoria imprensa reportaram no portal do MEC – Ministério da Educação e Cultura o acontecido e mencionaram o quadro digital com a postagem da seguinte manchete:

[...] Tebaldi visitou ainda as escolas João Frassetto e São Cristóvão, em Criciúma.[...] Na São Cristóvão, de ensino fundamental, que também atende a alunos surdos foi feita a apresentação de um coral de surdos e a demonstração da lousa eletrônica desenvolvida por um professor.

A sua assessoria de imprensa, posteriormente contatou com diretora, Deonilde Texeira, para manter contato com o pesquisador do protótipo para futuros trabalhos.

8 CONCLUSÃO

O estudo realizado ao longo da pesquisa promoveu o entendimento dos conceitos de rastreamento de objetos, bem como o mapeamento destes objetos de um suporte para as representações na LDI e na tela do computador por intermédio da técnica de *viewport*. O trabalho abordou a representação de objetos, formas de identificá-los, rastreá-los e com o uso de mapeamento de coordenadas representar da SRU para SRD.

Desse modo, a partir deste trabalho é possível aprimorar e conceber novas aplicações que envolvam *hardware* e *softwares* não só da área da educação, como também segurança, entretenimento, etc. Para a educação, que é a área aplicada dessa pesquisa, o estudo possibilitou a criação de uma LDI.

A utilização de *software* livre no desenvolvimento deste projeto facilitou alcançar os objetivos, devido ao grande compartilhamento e socialização de conhecimento da comunidade ao criar e disponibilizar, a API WiiuseJ, que serviu como cerne do projeto de LDI.

Dentre as dificuldades encontradas, constatou-se o acesso limitado a materiais em língua portuguesa para este tema. Contudo, esta limitação foi compensada com excelentes trabalhos produzidos para eventos e convenções da IEEE. Espera-se que este trabalho seja motivador para futuros trabalhos na área.

O protótipo possibilitou a concepção de uma LDI de baixo custo, o que torna possível o acesso a este tipo de tecnologias para escolas das redes públicas e privadas. O protótipo que foi desenvolvido pode trazer novas possibilidades para área da educação na escola e na região, especialmente com a incorporação das tecnologias educacionais.

Para trabalhos futuros na área de rastreamento, espera-se que este trabalho seja motivador como objeto de estudo e material de referência para o desenvolvimento de novos protótipos para outras áreas. E em se tratando de educação, pode-se propor um aprimoramento do software, automatizando algumas funções que conduzam à facilidade de uso e à praticidade, e aperfeiçoar a técnica de mapeamento, usando outras funções de *viewport*.

REFERÊNCIAS

ALLEN, B.D.; BISHOP, G.; WELCH, G..**Tracking: Beyond 15 Minutes of Thought**, *Proc. 28th Ann. Conf. ComputerGraphics and Interactive Techniques (Siggraph 01)*, ACM Press, 2001,

ARNAUD, E.; MEMIN, E.; FRIAS, B. Cernuschi.**Conditional filters for image sequence based tracking application to point tracking**, *IEEE-T-IP*, 14(1):63-79, 2005

BERNARDIN, Keni; EKENEL, Hazim Kemal; STIEFELHAGEN, Rainer; **Multimodal identity tracking in a smart room**. p. 387 (2009)

BARBOSA, Maria Lúcia Marangon. **Utilizando o computador como ferramenta pedagógica para vencer a resistência do professor**: o caso da 38ª Superintendência Regional de Ensino de Ubá-MG. Florianópolis : UFSC, 2002.

BOELTER, Eguemar Luiz. Tecnologia no cotidiano. In: **Gestão em Rede**, n. 74, nov. 2006. p. 19-20.

BLACK, J., **A Novel Method for Video Tracking Performance Evaluation**, in *Proc. IEEE PETS Workshop*, Oct. 2003.

BLINN, James; **Pixel coordinates**; *IEEE Computer Graphics and Applications*; 1991.

BLINN, James: **A trip down the graphics pipeline**: Grandpa, what does viewport mean? *IEEE Computer Graphics*; Jan 1992.

BLINN, James, NEWELL, Martin. **Clipping using homogeneous coordinates**. *Computer Graphics*; Agosto 1978.

BLINN, James. **A trip down the graphics pipeline**: The homogeneous perspective transform. *IEEE Computer Graphics and Applications*; Maio 1993.

BROWN, L.M..**Performance Evaluation of Surveillance Systems Under Varying Conditions**, in *Proc. IEEE PETS Workshop*, Jan. 2005.

BUCKINGHAM, América L. **Informática e educação**: Rio de janeiro: Vozes, 2008.

CASTELLS, Manuel. **A Era da Informação: Economia, Sociedade e Cultura - Volume I:A Sociedade em Rede**. Tradução de Roneide Venâncio Majer. São Paulo: Paz e Terra, 2005.

COLL, César; MONEREO, Charles. **Psicologia da Educação Virtual**: aprender e ensinar com as tecnologias da informação e da comunicação; Porto Alegre, 2010

DECARLO, D. ; METAXAS,D. **Optical flow constraints on deformable models with**

applications to face tracking, Intl. J. of Computer Vision, vol. 38, no. 2, pp. 99–127, 2000.

DEITEL, H. M.; DEITEL, P. J. **Java** : como programar; tradução e revisão técnica Carlos Arthur Lang Lisboa. 4a. edição, 2003.

GOKTURK, S.; BOUGUET, J.; GRZESZCZUK, R. **A datadriven model for monocular face tracking**. In Proc. Int. Conf. Computer Vision, pages 701–708, 2001.

JAVED, Omar; SHAH, Mubak; YILMAZ, Alper. **ObjectTracking**: A Survey. Disponível em: <www.cs.ucf.edu/vision/public_html/.../Object%20Tracking.pdf>. Acesso em 9 de abril. 2010.

LEE, Chung Lee. **HackingtheNintendo Wii Remote**. Disponível em: <www.cs.cmu.edu/~15-821/CDROM/PAPERS/lee08.pdf>. Acesso em 13 de Maio de 2010.

LEMAY, Laura; cadenhead, Rogers. **Aprenda em 21 dias**: Java. Rio de Janeiro: Campus, 1999.

LÉVY, Pierre. **As tecnologias da inteligência**: o futuro do pensamento na era da informática. Tradução: Carlos Irineu da Costa. São Paulo: 34, 1993.

MORAN, José Manuel. **Informática na Educação**: Teoria & Prática. Porto Alegre, vol. 3, 2000

MORAES, Cristiane Campos de Oliveira. **Integração da Informática na Educação**: a experiência da rede municipal de ensino de Juiz de Fora - Minas Gerais. Florianópolis: UFSC, 2002.

NAKASHIMA, Rosária Helena Ruiz. **Sistematização de indicadores didático-pedagógicos da linguagem interativa da lousa digital**. Disponibilizado em: <http://www.pucpr.br/eventos/educere/educere2008/anais/pdf/521_204.pdf> (2008) . Acesso em 27 Maio de 2010.

NAKASHIMA, Rosária Helena Ruiz; BARROS, Daniela Melaré Vieira; AMARAL, Sergio Ferreira do. **O uso pedagógico da lousa digital associado à teoria Dos estilos de aprendizagem**. Disponível em: <http://www.uned.es/revistaestilosdeaprendizaje/numero_4/Artigos/lsr_4_articulo_12.pdf> (2009). Acesso em: 22 de maio. 2010.

ODOBEZ, J.; PEREZ, D.; BA S. O.. **Embedding motion in model-based stochastic tracking**, IEEE T-IP, 2006.

SHALOM, Y. Bar; RONG, L. Xiao; KIRUBARAJAN, T.. **Estimation, Tracking and Navigation: Theory, Algorithms and Software**. John Wiley and Sons, 2001.

SHALOM, Y Bar; JAFFER, A. G. **Tracking and Data Association**. Academic Press, San Diego, CA, 1998.

SHIN, Jaewon; LEE, Nelso; THRUN, Sebastian; GUBIA, S Leonidas. **Lazy inference on object identities in wireless sensor networks**. Stanford, 2006.

SONG, K. Y., KITTLER, J., AND PETROU, M.. **Defect detection in random color textures**. Israel Verj, 1996

SMITH, Kevin; PEREZ, Daniel Gatica; ODOBEZ, Jean-Marc; BA, Sileye. **Evaluating Multi-Object Tracking**, Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops, p.36, June 20-26, 2005

STAUFFER, C. ; GRIMSON, E.. **Adaptive background mixture models for real-time tracking**. CVPR, vol2, p246--252, 1999.

VEIGA, Ilma Passos Alencastro. **Projeto político-pedagógico da escola: uma construção possível**. 11. ed. Campinas: Papirus, 1995.

VERAS, Dauro **Conteúdo digital torna as aulas mais interativas**, ed. Investimentos, SP, 2009. Disponível em: http://www.linearclipping.com.br/cnte/detalhe_noticia.asp?cd_sistema=93&codnot=956551>. Acesso em: 10 de maio. 2010

VIDAL, F. B.; ALCALDE V.H.C. , **Motion segmentation in sequential images based on the differential optical flow**, 2nd International Conference on Informatics in Control, Automation and Robotics -ICINCO,, 2005.

WANG, Yiwei; DOHERTY, John F; **Moving Object Tracking in Video**. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.97.7856&rep=rep1&type=pdf>>. Acesso em 22 de Abril de 2010.

WEISS, Maria Lúcia Lemme. **Psicopedagogia Clínica: uma visão diagnóstica dos problemas de aprendizagem escolar**. Rio de Janeiro: DP e A 8ª. Ed. 2001.

WILDES, R.; KUMAR, R.; SAWHNEY, H.; SAMASEKERA, S. ; HSU, S. ; TAO, H.; GUO, Y.; HANNA, K. ; POPE, A.; HIRVONEN, D.; HANSEN, M.; BURT, P. **Aerial video surveillance and exploitation**, Proceedings of the IEEE, vol. 89, no. 10, p. 1539, 2001.

WREN, C.; AZARBAYEJANI A A.; DARRELL, T.; PENTLAND, A.. **Pfinder: Real-time**

tracking of the human body, IEEE Trans. Pattern Anal. Machine Intell., vol. 19, pp. 780–785, 1997.

RUI, Y.; CHEN, Y..**Object tracking using unscented particle filter**.IEEE Conf. Computer Vision and Pattern Recognition, Kauai, Hawaii, volume II, 2001.

YU, T; WU, Y. **Collaborative tracking of multiple targets**.In*Proc. of CVPR'04*, Washington DC, USA.

Rastreamento de objetos aplicado ao desenvolvimento de um protótipo de lousa digital interativa

Thaffareu Roberto Brigido¹, Evânio Ramos Nicoleit², Paulo João Martins²

¹ Acadêmico do Curso de Ciência da Computação – Departamento de Ciência da Computação – Universidade do Extremo Sul Catarinense (UNESC) – Criciúma, SC - Brazil

² Professor do Curso de Ciência da Computação – Departamento de Ciência da Computação – Universidade do Extremo Sul Catarinense (UNESC) – Criciúma, SC - Brazil

{thaffa_rb, evnic}@hotmail.com, pjm@unesc.net

Abstract. *The object tracking is a technique that allows isolating and tracking a specific object for various purposes, such as collision detectors, radar and facial recognition. Models of viewport allow representing objects from a domain in another support, such as a computer screen or a multimedia projection. Based on these considerations, this work describes the development of a Digital Interactive Whiteboard - LDI, a support tool for the educational process, based on object tracking. The study discusses the integration of viewport and object tracking models for the development of LDI. The results are motivators for improving the system. The study found significant cost savings, since the cost of this equipment is typically high for educational institutions.*

Resumo. *O rastreamento de objetos é uma técnica que possibilita isolar e acompanhar um determinado objeto para vários fins, tais como detectores de colisões, radares e reconhecimentos faciais. Os modelos de viewport, por sua vez, possibilitam representar objetos de um domínio em outro suporte, tal como a tela de um computador ou uma projeção multimídia. Com base nessas considerações, este trabalho descreve o desenvolvimento de uma Lousa Digital Interativa – LDI, uma ferramenta de apoio ao processo educacional, baseado no rastreamento de objetos. O estudo discute a integração de modelos de viewport e rastreamento de objetos para o desenvolvimento da LDI. Os resultados são motivadores para o aprimoramento do sistema. O estudo indicou redução significativa de custos, já que o custo destes equipamentos é tipicamente elevado para as instituições de ensino*

1. Introdução

Por rastreamento de objetos pode-se entender um método de seguir um determinado objeto através de sucessivos quadros de imagem, com o objetivo de se conhecer a forma como ele se move em relação a outros objetos (SHALOM; JAFFER, 1998).

Muitas técnicas vêm sendo desenvolvidas para esses sistemas de rastreamento em diversos segmentos, tais quais vigilância automatizada, monitoramento de tráfego, navegação automatizada de carros e interação humana com computador (SHAH; JAVED; YILMAZ, 2006), ressaltando-se que esta última técnica está sendo desenvolvida para fins de realidade virtual, jogos, quadro digitais, tecnologias *touchscreen*, entre outras.

Há vários modos de se analisar o objeto a ser rastreado. Contudo, há de se considerar que para rastrear um objeto deve-se isolar sua singularidade para que se possa distingui-los de outras formas ou objetos. Somente dessa forma é que o mesmo pode ser rastreado com precisão. As

formas mais comuns de rastreamento são por cores, arestas, texturas e quantidade de brilho (SHAH; JAVED; YILMAZ, 2006).

Algumas formas se sobressaem em relação a outras dependendo da aplicação, como por exemplo, um sistema de rastreamento baseado em cores ou brilho, promove a percepção mais rápida dos objetos e cálculos de trajetórias. Entretanto, é menos seguro (DIETMAYER; SPARBERT; STRELLER, 2001).

Com base nessas considerações introdutórias, este trabalho focaliza o estudo de técnicas envolvidas para a criação de um protótipo de software em plataforma Java, que seja capaz de processar os dados fornecidos por um hardware de rastreamento de objetos, para a criação de uma Lousa Digital Interativa – LDI.

2. Tecnologias na Educação

A capacidade das tecnologias informacionais, em especial o computador, de processar e exibir programas com sons, imagens, animação e textos, de modo interativo e integrado pode contribuir significativamente para a melhoria da qualidade do processo ensino-aprendizagem (MORAES, 2007)

Alguns aspectos podem ser destacados com relação à melhoria da qualidade no processo ensino-aprendizagem com a informática como recurso pedagógico: desenvolvimento da compreensão de conceitos matemáticos; melhoria da linguagem e da escrita; favorecimento da interdisciplinaridade; desenvolvimento da criatividade; desenvolvimento da autonomia, da cooperação e da criticidade (MORAES, 2007).

Para garantir o êxito da incorporação do recurso informático como instrumento útil para a atividade intelectual, criativa e profissional, é preciso que se garanta uma capacitação do docente não somente em nível técnico, como também, e fundamentalmente, pedagógico. O professor precisa apropriar-se da tecnologia em função de seus interesses profissionais, e assim situar-se, avaliar e planejar sua aplicação em aula (BOELTER, 2006).

2.1 Lousas Digitais Interativas - LDI

Conforme visto, a tecnologia vem atingindo pontos cruciais para a educação, sua crescente e constante inovações e atualizações sobre como definir e representar informações está cada vez mais visível no contexto atual, como foi o caso do computador, que ajudou a elaboração de trabalhos, apresentações áudios-visuais, entre tantas outras melhorias. Outra conquista para o mundo da educação foi o ensino a distância, graças à internet e seus meios de divulgação da informação ficou muito mais fácil aprender, com esses novos métodos de ensino, como aulas por vídeo conferência e fórum de atividades pela internet. (MORAN, 2000).

De acordo com a Lei de Diretrizes e Bases da Educação Nacional, de 1996 (LDB), em seu artigo 36, a introdução das novas tecnologias deve fazer parte dos currículos das escolas públicas do País, bem como discussão crítica dos meios de comunicação social. Esse processo perpassa, obrigatoriamente, pela necessidade de explorar os diversos meios como: a televisão, o rádio, o jornal, o vídeo, a revista, o computador, o teatro, o fax, os softwares educacionais (FRANÇA; SANTOS, 2005).

Neste contexto de educação com base em recursos tecnológicos, a lousa digital pode ser de grande valia no processo de aprendizagem. Com essa lousa, os professores poderão usar recursos como imagens, sons, e vídeos, navegar na internet e muitos outros recursos. Entretanto, devido ao alto custo de equipamentos como este, faz com que muitas escolas não consigam ter acesso a tal, e uma vez que escolas e universidades particulares já possuam essa tecnologia, nas escolas publicas essa é uma realidade muito distante.

3. Rastreamento de Objetos

O rastreamento de objetos pode ser definido como um processo de localização de uma ou mais características desejadas de um objeto em vários frames de uma cena, seguindo sua semelhanças em cada cena para criar uma possível rota para rastrear tal objeto (VIDAL; ALCALDE, 2005).

A proliferação de computadores de alto desempenho, a disponibilidade de alta qualidade e de baixo custo das câmeras de vídeo, bem como a necessidade crescente de automatização nas análises de vídeo geraram um grande interesse em algoritmos de monitoramento de objetos. Dentro desse contexto, há três passos importantes em vídeo análise: detecção de objetos em movimento, rastreamento de objetos e análise do objeto de modo a reconhecer seu comportamento.

A forma como os componentes são combinados e ponderados é dependente da aplicação e desempenha um papel decisivo para a eficiência do rastreador, como por exemplo, o monitoramento de um rosto em uma cena precisa confiar mais no alvo do que na sua representação sobre a dinâmica de destino (DECARLO; METAXAS, 2000).

Há duas etapas fundamentais a serem cumpridas para a realização completa de um rastreamento visual de objetos. A primeira é a representação e a localização do objeto, que é essencialmente um processo de *bottom-up*, que lida com as mudanças no aparecimento do alvo, e a filtragem associada aos dados, que é essencialmente um processo de *top-down*, que lida com a dinâmica do objeto rastreado, e avaliação de diferentes hipóteses.

3.1. Representação do Objeto

Em um cenário de rastreamento, um objeto pode ser definido como algo que é de interesse para outras análises. Por exemplo, os barcos no mar, os peixes dentro de um aquário, os veículos em uma estrada, aviões no ar, as pessoas andando em uma estrada ou bolhas na água são um conjunto de objetos que podem ser importantes para controlar em um domínio específico. Os objetos podem ser representados por suas formas e aparências. Estes podem ser representados nesse conjunto de formas:

- a) Pontos: o objeto é representado por um ponto central (Figura 1 (a)), ou por um conjunto de pontos (Figura 1 (b)). Em geral, a representação de ponto é adequado para rastreamento de objetos que ocupam pequenas regiões em uma imagem (SERBY, 2004).
- b) Formas geométricas primitivas: o objeto é representado por um retângulo, uma elipse (Figura 1 (c), (d)), etc. O uso de formas geométricas primitivas são mais adequados para representar objetos simples e rígidos (COMANICIU, 2003).
- c) Objeto pontilhado e contorno: essa representação define o limite de um objeto (Figura 1 (g), (h)). A região no interior do contorno é chamada de silhueta do objeto (Figura 1 (i)). Silhueta é a representação do objeto, esse tipo de rastreamento é adequado para o rastreamento de formas não rígidas e complexas (YILMAZ, 2004).
- d) Formas articuladas: objetos articulados são compostos de partes do corpo que são realizada em conjunto com as articulações. Por exemplo, o corpo humano é um objeto articulado com tronco, pernas, mãos, cabeça e pés ligados por articulações. A relação entre a partes é governada por modelos cinemáticos de movimento, como o ângulo articular. A fim de representar um objeto articulado, pode-se juntar modelo das partes constituintes de cilindros ou elipses, como na Figura 1 (e) (AGGARWAL, 2001, tradução nossa).
- e) Modelos esqueléticos: estes modelos podem ser extraídos através da aplicação do eixo do meio, de forma a transformar o objeto em silhueta. Este modelo é muito usado como uma

maneira de representação para reconhecer objetos. A representação do esqueleto pode ser usada para a modelagem dos objetos articulados e rígidos (Figura 1 (i)) (BALLARD, 2001).

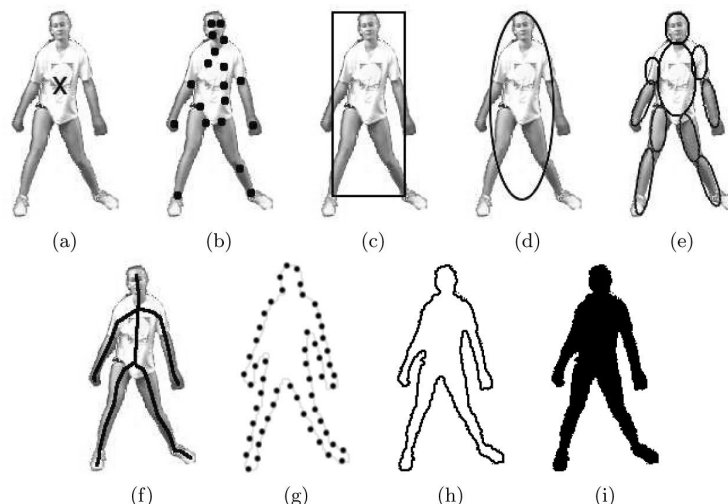


Figura 1 Representação de Objetos

3.2. Métodos Usados para Rastreamento

Selecionar as características certas desempenha um papel crítico no rastreamento. Em geral, o mais desejável é isolar a singularidade de um objeto dentro do recurso visual para que os objetos possam ser facilmente distinguidos em função do espaço. Em geral, os algoritmos de rastreamento podem recorrer a uma combinação desses recursos, obtendo um melhor desempenho no rastreamento e usando menor complexidade e recursos computacionais. Os detalhes das características visuais comuns são os descritos na sequência:

- a) Cor: a cor aparente de um objeto é influenciada principalmente por dois fatores físicos: a distribuição de potência de luminosidade e o reflexo de luz sobre superfície das propriedades do objeto
- b) Bordas: os limites dos objetos costumam gerar fortes mudanças na intensidade da imagem. Detecção de bordas é usada para identificar essas mudanças. Uma propriedade importante das bordas é que elas são menos sensíveis às mudanças de iluminação em relação às características de cor.
- c) Fluxo óptico: é um denso campo de vetores de deslocamento que define a tradução de cada pixel em uma região. Ele é calculado a partir da restrição de luminosidade, que assume a constância de luminosidade dos pixels correspondentes em quadros consecutivos (HORN, SCHUNK 1981). Fluxo óptico é normalmente usado como um recurso em movimento com base em aplicações de segmentação e rastreamento (HORN, SCHUNK 1981).
- d) Textura: é uma medida da variação da intensidade de uma superfície que quantifica propriedades, tais como a lisura, densidade, irregularidades e etc. Comparado-se à cor, a textura exige uma etapa de processamento para gerar descritores.

Geralmente, as características são escolhidas manualmente pelo usuário, dependendo do domínio de aplicação. Para tal, há uma combinação entre método de rastreamento e reconhecimento de objeto, de forma que para cada aplicação se consiga chegar a um resultado

melhor, como por exemplo, para um quadro digital, o grupo de rastreamento que deve ser estudado.

3.3. Dispositivos Rastreadores

Entre os tipos de dispositivos rastreadores que podem ser utilizados para a confecção de um lousa digital, podem-se citar as câmeras digitais.

As câmeras são naturalmente a principal classe de sensores. Além disso, o processamento de visão e os ganhos provenientes do aumento do poder de computação das últimas décadas permitem uma análise em tempo real para o monitoramento de um objeto com o mesmo exercendo certo esforço de hardware razoavelmente aceitável (BEHRINGER, 1997).

No entanto, a diferença entre os dados da imagem bruta e as representações geométricas dos objetos interessantes ainda continuam grandes, especialmente os parâmetros geométricos, como a distância e a velocidade de outros objetos, mas exige ainda mais poder de processamento e esforço de hardware.

Outro sensor disponível e de fácil acesso seria o do tipo sensores infravermelhos, que conseguem uma análise dos rastreamento de objetos em tempo real, com pouquíssimo esforço computacional, como é o caso do Wiimote, sobre o qual discorre-se a seguir.

4. ViewPort

Viewport Mapping é a limitação de uma região na qual o desenho ou um objeto em questão é exposto em uma janela de exibição. Essa janela representa um determinado objeto do mundo real ou de um desenho para sua representação digital, bem como o seu posicionamento e escalonamento fiel do objeto que se busca. Para fazer essa representação digital, usam-se cálculos de geometria e matemática por um sistema de coordenadas (BLINN, 1992).

Essas equações também podem ser derivadas de um conjunto de transformações que converte a área da janela para a área do visor (BLINN, 1992).

4.1. Sistemas de Referências

Um sistema de coordenadas é a referência de objetos de um determinado meio para outro. É medido em tamanho e posicionamento de sua representação de um plano para outro. Quando esse sistema de coordenadas é usado para alguma finalidade específica, ele ganha o nome de Sistema de Referência, e deve respeitar características fundamentais para que isso ocorra e apresentar uma unidade de referência básica e seus limites extremos de valores. No mercado atual, há alguns sistemas bem utilizados, que são:

- a) SRO (Sistema de Referência do Objeto): Este sistema descreve um objeto com o próprio sistema de coordenadas como se ele fosse um pequeno universo dentro dele.
- b) SRN (Sistema de Referência Normalizado): Usa os valores entre 0 e 1. Ele é como uma referência para intermediar os valores do SRU e SRD.
- c) SRD (Sistema de Referência do Dispositivo): Utilizam-se coordenadas de um dispositivo de saída específico. Em um visor de celular, pode indicar o número máximo de pixels ou uma de suas resoluções específicas de uma aplicação, por exemplo: (640x480), (320x480).
- d) SRU (Sistema de Referência do Universo): Também é denominado como coordenadas do mundo.

4.2. Transformação e Visualização 2D

O mapeamento na janela de visualização acontece no sistema de referência do universo SRU e o SRD. Obedece a uma área na matriz de pixels da saída do dispositivo, podendo até corresponder a toda essa matriz, já que se permite ter várias *viewports* no dispositivo.

No plano cartesiano, como as suas coordenadas reais são infinitas, para que seja possível representar qualquer modelo deste espaço dentro da *viewport*, é necessário definir então um domínio no qual os objetos possam ser representados. Esse domínio é chamado de janela.

As proporções relativas dos objetos são mantidas se os fatores de escala são os mesmos ($F_{sx} = F_{sy}$). Caso contrário, os objetos do mundo vão ser esticados ou comprimidos. Para manter as proporções de escala fiéis, deve-se usar um escalonamento com dois pontos. Em cada ponto dessa coordenada, X e Y recebem seus valores mínimos e máximos de sua representação na tela, sendo assim, X_{min} , X_{max} , Y_{min} e Y_{max} . Mesmo que o objeto não seja mostrado em toda tela de visualização, ocorre um processo de corte, chamado de *clipping*. Porém as suas proporções sempre serão mantidas, conforme verificado na Figura 2.

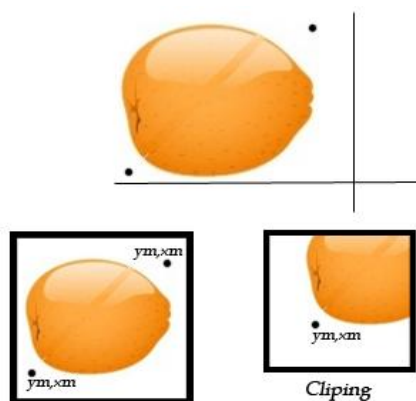


Figura 2 Dois pontos referênciais

A transformação do objeto SRU para SRD se tem por cálculo de proporcionalidade. Então, a coordenada se estabelece da forma representada na Figura 3:

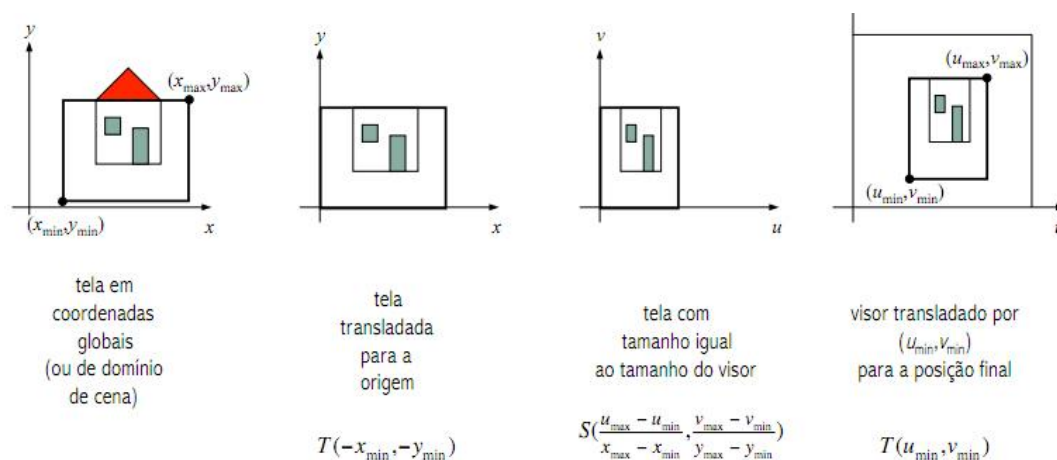


Figura 3 Transformação do objeto em referência SRU para SRD

Com os cálculos de proporcionalidade, é possível chegar à seguinte matriz para realização da visualização de pixel do dispositivo:

$$\frac{XV - XVMIN}{X - XMIN} = \frac{XVMAX - XVMIN}{XMAX - XMIN}$$

Resolvendo-se, então, o X em função do XV e o XV em função de X, se tem as relações:

$$XV = \underbrace{\frac{XVMAX - XVMIN}{XMAX - XMIN}}_{SX \text{ (fator de Escala)}} (X - XMIN) + XVMIN$$

$$\begin{cases} XV = SX(X - XMIN) + XVMIN \\ X = (XV - XVMIN) / SX + XMIN \end{cases}$$

Na equação, SX é o fator de escala na dimensão horizontal. Então, o cálculo para a coordenada do objeto em função de X é :

$$\text{Coordenada } X = \frac{(X - XVirtMin) * (XMostraMax - XMostraMin)}{XVirtMax - XVirtMin} + XMostraMin$$

Consequentemente, a analogia da coordenada para o eixo vertical em função de Y é a mesma, ficando da seguinte maneira:

$$\text{Coordenada } Y = \frac{(Y - YVirtMin) * (YMostraMax - YMostraMin)}{YVirtMax - YVirtMin} + YMostraMin$$

Com essa fórmula, calculam-se os valores de coordenadas de certo ponto e os transporta do mundo real para a janela de visualização do *Viewport*.

5.1. Metodologia

O levantamento bibliográfico referente à pesquisa proporcionou um vasto conhecimento dos conceitos científicos teóricos e práticos, uma vez que grande parte das referências bibliográficas possuem suas respectivas publicações na IEEE. Como visto, também foi feita uma breve análise dos hardwares capazes de servirem de sensor para o rastreamento de objetos.

Depois de analisado o hardware, fez-se necessário optar por uma linguagem de programação a linguagem de programação foi o Java, uma linguagem cuja licença de uso é livre e que vem sendo vastamente utilizada, sem mencionar a gama de APIs livres disponíveis na internet. Usou-se a API WiiuseJ, por ter diferentes e completos recursos para controlar o rastreador de objetos infravermelho disponível no Wiimote.

5.2 Protótipo

Para o desenvolvimento da LDI, como proposto por esta pesquisa, são necessários alguns recursos de hardware e software, sendo assim, é necessário o uso de um hardware capaz de visualizar o objeto a ser rastreado e repassar os dados ao computador.

A escolha deu-se para o uso do hardware mencionado, o Wiimote, o controle de um console de videogame, capaz de rastrear leds infravermelhos em tempo hábil para o processamento dos dados obtidos. Como recurso de hardware também é imprescindível o uso de um dispositivo para projeção da tela da lousa digital, podendo ser qualquer projetor de quaisquer marca disponível.

Para que não haja problemas com interferências ou o mau uso dos recursos fazendo ineficaz o rastreamento do LED infravermelho, deve-se tomar algumas medidas para o bom

funcionamento da lousa digital. A disposição dos equipamentos. o cenário deve ficar parecido com a Figura 4, tendo o projetor multimídia (a), a lousa projetada por ele (b), a caneta emitindo luz infravermelho (c) para a câmera rastreadora (d) que emite por Bluetooth para o computador (e).

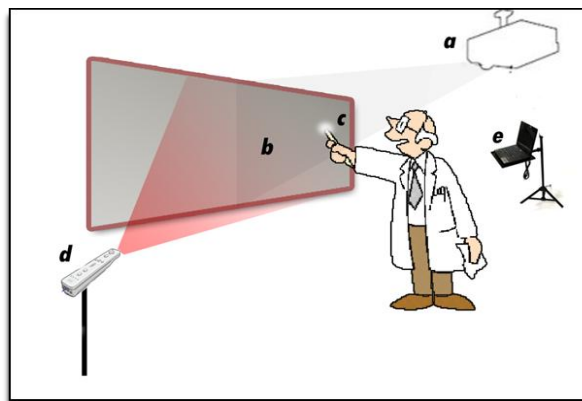


Figura 4 Ilustração do Cenário

5.2.1. Utilização do Viewport

Para criação do protótipo, deve-se mapear a coordenada do LED do espaço físico e representá-los na tela do computador. Sabe-se que o computador tem sua resolução em diferentes monitores, diferentes sistemas, diferentes padrões de resolução e etc. Neste protótipo, utilizou-se da resolução 1024x768, que é a mais comum utilizada pelos projetores e que é a resolução da câmera rastreadora infravermelho.

A API demonstra o ponto do LED através de um painel chamado IRPANEL. O IRPANEL cria uma imagem virtual representativa do ponto rastreado. Nesse painel, tem-se duas coordenadas, a coordenada calculada que cria um plano cartesiano em meio à representação, e o valor da coordenada absoluta, ou seja, 1024x768px, a coordenada absoluta nativa da câmera rastreadora, que é a que será usada para fazer o mapeamento.

Precisa-se, então, abstrair os valores, as nomenclaturas, os tamanhos entre as janelas para que possa ser colocado na fórmula, conforme pode-se verificar na Figura 5.

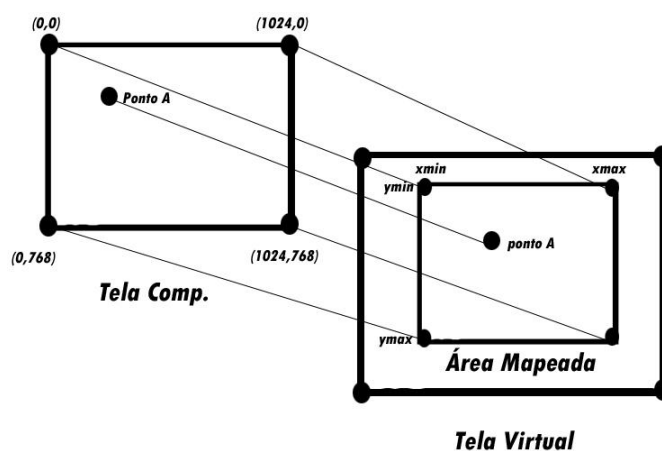


Figura 5 Modelo de Viewport Sobre o Protótipo

Entretanto, para se obter esses valores, deve-se calibrar as coordenadas que devem ser obtidas rastreando as pontas extremas da tela de projeção e editando no código fonte. Logo depois de calibradas as coordenadas, deve-se alterar o código com a expressão vista anteriormente neste artigo, como mostra a Figura 6

```

3. if (robot != null) { // if mouse control activated
4.
5.     cordx = (int) (((arg0.getAx() - xvfinal) * 1024) / (xvinicial-xvfinal));
6.     cordy = (int) (((arg0.getAy() - yvfinal) * 768) / (yvinicial-yvfinal));
7.
8.     robot.mouseMove(((cordx-1024)*-1), ((cordy-768)*-1));

```

Figura 6 Inserção da Formula no Programa

Note-se que quando posto na função *robot*, que é a biblioteca que possibilita mover o mouse, é subtraído o MostraMax é multiplicado por 1 negativo, pois as coordenadas entre a imagem virtual e a real são opostas.

5.3. Resultados Obtidos

O protótipo desenvolvido possibilitou alcançar todos os objetivos propostos. O software conecta a câmera rastreadora, recebe os dados via Bluetooth, interpreta os dados de coordenadas e, usando de técnicas de mapeamento gráfico, transforma o sistema em uma LDI.

O software resultante pode ser utilizado tanto em SO Linux quanto em Windows. Por ser feito em Java, ele possui característica multiplataforma. Entretanto há 4 arquivos que sempre devem acompanhar o software em sua pasta raiz, sendo que o *libwiiuse.so* e *libWiiuseJ.so* responsáveis para o funcionamento no Linux e *wiiuse.dll*, *WiiUseJ.dll* para funcionamento no Windows.

Em 27 de Abril de 2011, deu-se o primeiro teste do protótipo em uma Escola Estadual do município de Criciúma: a Escola de Ensino Fundamental São Cristóvão, que teve suas aulas de informática com o uso do recurso da LDI. A professora da escola, em regime de contratação temporária, Fabiana Milanez, teve ao seu dispor o recurso. A docente fez uso da LDI em suas aulas de ensino em informática e tecnologia para os alunos da 3ª série do Ensino Fundamental. O protótipo teve um bom funcionamento e em momento algum se fez ineficaz ou o acontecimento de erros inesperados, o resultado foi uma aula diferenciada e mais interativa.

5. Conclusão

O estudo realizado ao longo da pesquisa promoveu o entendimento dos conceitos de rastreamento de objetos, bem como o mapeamento destes objetos de um suporte para as representações na LDI e na tela do computador por intermédio da técnica de viewport. O trabalho abordou a representação de objetos, formas de identificá-los, rastreá-los e com o uso de mapeamento de coordenadas representar da SRU para SRD.

Desse modo, a partir deste trabalho é possível aprimorar e conceber novas aplicações que envolvam hardware e softwares não só da área da educação, como também segurança, entretenimento, etc. Para a educação, que é a área aplicada dessa pesquisa, o estudo possibilitou a criação de uma LDI.

A utilização de software livre no desenvolvimento deste projeto facilitou alcançar os objetivos, devido ao grande compartilhamento e socialização de conhecimento da comunidade ao criar e disponibilizar, a API *WiiuseJ*, que serviu como cerne do projeto de LDI.

Para trabalhos futuros na área de rastreamento, espera-se que este trabalho seja motivador como objeto de estudo e material de referência para o desenvolvimento de novos protótipos para outras áreas. E em se tratando de educação, pode-se propor um aprimoramento do software, automatizando algumas funções que conduzam à facilidade de uso e à praticidade, e aperfeiçoar a técnica de mapeamento, usando outras funções de *viewport*.

Referências

- BERNARDIN, Keni; EKENEL, Hazim Kemal; STIEFELHAGEN, Rainer; Multimodal identity tracking in a smart room. p. 387 (2009)
- BARBOSA, Maria Lúcia Marangon. Utilizando o computador como ferramenta pedagógica para vencer a resistência do professor: o caso da 38ª Superintendência Regional de Ensino de Ubá-MG. Florianópolis : UFSC, 2002.
- BLACK, J., A Novel Method for Video Tracking Performance Evaluation, in Proc. IEEE PETS Workshop, Oct. 2003.
- BLINN, James; *Pixel coordinates*; IEEE Computer Graphics and Applications; 1991. BLINN, James: A trip down the graphics pipeline: Grandpa, what does viewport mean? IEEE Computer Graphics; Jan 1992.
- BLINN, James, NEWELL, Martin. Clipping using homogeneous coordinates. Computer Graphics; Agosto 1978.
- BLINN, James. A trip down the graphics pipeline: The homogeneous perspective transform. IEEE Computer Graphics and Applications; Maio 1993.
- BUCKINGHAM, América L. Informática e educação: Rio de janeiro: Vozes, 2008.
- CASTELLS, Manuel. **A Era da Informação: Economia, Sociedade e Cultura** - Volume I: A Sociedade em Rede. Tradução de Roneide Venâncio Majer. São Paulo: Paz e Terra, 2005.
- JAVED, Omar; SHAH, Mubak; YILMAZ, Alper. ObjectTracking: A Survey. Disponível em: <www.cs.ucf.edu/vision/public_html/.../Object%20Tracking.pdf>. Acesso em 9 de abril. 2010.
- LEE, Chung Lee. HackingtheNintendo Wii Remote. Disponível em: <www.cs.cmu.edu/~15-821/CDROM/PAPERS/lee08.pdf>. Acesso em 13 de Maio de 2010.
- LEMAY, Laura; cadenhead, Rogers. Aprenda em 21 dias: Java. Rio de Janeiro: Campus, 1999.
- LÉVY, Pierre. As tecnologias da inteligência: o futuro do pensamento na era da informática. Tradução: Carlos Irineu da Costa. São Paulo: 34, 1993.
- MORAN, José Manuel. Informática na Educação: Teoria & Prática. Porto Alegre, vol. 3, 2000
- MORAES, Cristiane Campos de Oliveira. Integração da Informática na Educação: a experiência da rede municipal de ensino de Juiz de Fora - Minas Gerais. Florianópolis: UFSC, 2002.
- ODOBEZ, J.; PEREZ, D.; BA S. O.. Embedding motion in model-based stochastic tracking, IEEE T-IP, 2006.
- SHALOM, Y. Bar; RONG, L. Xiao; KIRUBARAJAN, T.. Estimation, Tracking and Navigation: Theory, Algorithms and Software. John Wiley and Sons, 2001.

- SHALOM, Y Bar; JAFFER, A. G. Tracking and Data Association. Academic Press, San Diego, CA, 1998.
- SHIN, Jaewon; LEE, Nelso; THRUN, Sebastian; GUBIA, S Leonidas. Lazy inference on object identities in wireless sensor networks. Stanford, 2006.
- SMITH, Kevin; PEREZ, Daniel Gatica; ODOBEZ, Jean-Marc; BA, Sileye. Evaluating Multi-Object Tracking, Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Workshops, p.36, June 20-26, 2005
- STAUFFER, C. ; GRIMSON, E.. Adaptive background mixture models for real-time tracking. CVPR, vol2, p246--252, 1999.
- VEIGA, Ilma Passos Alencastro. Projeto político-pedagógico da escola: uma construção possível. 11. ed. Campinas: Papirus, 1995.
- VIDAL, F. B.; ALCALDE V.H.C. , Motion segmentation in sequential images based on the differential optical flow, 2nd International Conference on Informatics in Control, Automation and Robotics -ICINCO,, 2005.
- WANG, Yiwei; DOHERTY, John F; Moving Object Tracking in Video. Disponível em <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.97.7856&rep=rep1&type=pdf>>. Acesso em 22 de Abril de 2010.
- WILDES, R.; KUMAR, R.; SAWHNEY, H.; SAMASEKERA, S. ; HSU, S. ; TAO, H.; GUO, Y.; HANNA, K. ; POPE, A.; HIRVONEN, D.; HANSEN, M.; BURT, P. Aerial video surveillance and exploitation, Proceedings of the IEEE, vol. 89, no. 10, p. 1539, 2001.