

FUNDAÇÃO DE ENSINO “EURÍPIDES SOARES DA ROCHA”
CENTRO UNIVERSITÁRIO “EURÍPIDES DE MARÍLIA” - UNIVEM
PROGRAMA DE MESTRADO EM CIÊNCIA DA COMPUTAÇÃO

LUIZ FERNANDO BRAGA LOPES

**O ESTUDO E A IMPLEMENTAÇÃO DE INTERFACES PARA
UTILIZAÇÃO EM SISTEMAS DE REALIDADE AUMENTADA**

Marília
2005

LUIZ FERNANDO BRAGA LOPES

O ESTUDO E A IMPLEMENTAÇÃO DE INTERFACES PARA
UTILIZAÇÃO EM SISTEMAS DE REALIDADE AUMENTADA

Dissertação apresentada ao Programa de
Mestrado do Centro Universitário
Eurípides de Marília, mantido pela
Fundação de Ensino Eurípides Soares da
Rocha, para obtenção do Título de
Mestre em Ciência da Computação.

Orientador:
Prof. Dr. Antonio Carlos Sementille

Marília
2005

LOPES, Luiz Fernando Braga

O Estudo e a Implementação de Interfaces para Utilização em
Sistemas de Realidade Aumentada / Luiz Fernando Braga Lopes;
Orientador: Antonio Carlos Sementille.

Marília, SP: [s.n.], 2005.

108 f.

Dissertação (Mestrado em Ciência da Computação) - Centro
Universitário Eurípides de Marília, Fundação de Ensino Eurípides
Soares da Rocha.

1. Interfaces 2. Realidade Aumentada

CDD: 006

*Dedico este trabalho ao meu Avô,
que, através de sua determinação
pela vida, mostrou que, apesar de
todas as dificuldades, nunca
devemos desistir. Só Deus sabe
como sinto sua falta.*

AGRADECIMENTOS

A realização deste trabalho não poderia ter acontecido de forma satisfatória se não fosse o empenho e suporte irrestrito prestado pelo professor Dr. Antonio Carlos Sementille, que sempre se mostrou atencioso no decorrer deste trabalho, apontando sempre a direção. A ele, meu muito obrigado.

Agradeço a todos os professores que foram além de simplesmente expor a matéria em suas aulas, a essência real da grandeza dos homens não está escrita em livros nem pode ser encontrada em universidades, está dentro da consciência daqueles que procuram se aperfeiçoar de maneira criativa e compreensiva.

Agradeço à minha família e namorada que, presentes, sempre me estimularam a atingir meus objetivos, e, ausentes, serviram como inspiração para que eu sempre desse o melhor de mim.

Aos meus amigos, companheiros e colegas que entenderam e respeitaram a minha ausência em certos momentos, meus sinceros agradecimentos.

A todos vocês que tanto me ensinaram, ajudaram e apoiaram em cada momento, o meu sincero agradecimento.

Agradeço aos membros da banca por disporem de seu precioso tempo para a leitura deste trabalho.

O valor das coisas não está no tempo que elas duram, mas na intensidade com que acontecem. Por isso, existem momentos inesquecíveis, coisas inexplicáveis e pessoas incomparáveis.

Fernando Pessoa

LOPES, Luiz Fernando Braga. **O Estudo e a Implementação de Interfaces para Utilização em Sistemas de Realidade Aumentada**. 2005. 105 f. Dissertação (Mestrado em Ciência da Computação) - Centro Universitário Eurípides de Marília, Fundação de Ensino Eurípides Soares da Rocha, Marília, 2005.

RESUMO

A comunicação entre o usuário e o *software* é uma etapa fundamental em qualquer projeto de um Sistema de Interação. Nos sistemas de Realidade Virtual, essa comunicação se faz por meio de uma interface gráfica que tem como objetivo fornecer uma representação visual das principais entidades e funções presentes no Ambiente Virtual. É nesse contexto que se situa este trabalho, o qual apresenta um estudo sobre as principais características da Realidade Aumentada, um levantamento comparativo das mais expressivas tecnologias desta área, uma conceituação da biblioteca empregada no desenvolvimento da pesquisa e uma abordagem da utilização da Realidade Aumentada na implementação de técnicas avançadas de Interfaces de Interação. Após o estudo preliminar desses assuntos, este trabalho descreve um suporte desenvolvido para interação em Ambientes de Realidade Aumentada baseada em captura óptica de marcadores passivos, utilizando a biblioteca Artoolkit. São descritos, também, os protótipos implementados a fim de validar o suporte desenvolvido, analisando-se os principais aspectos de desempenho.

Palavras-chave: Realidade Aumentada, Realidade Virtual, Artoolkit, Interação, Interfaces.

LOPES, Luiz Fernando Braga. **O Estudo e a Implementação de Interfaces para Utilização em Sistemas de Realidade Aumentada**. 2005. 105 f. Dissertação (Mestrado em Ciência da Computação) - Centro Universitário Eurípides de Marília, Fundação de Ensino Eurípides Soares da Rocha, Marília, 2005.

ABSTRACT

The communication between user and software is a basic stage in any Interaction System project. In Virtual Reality systems, this communication is established by the means of a graphical interface, whose objective is to supply a visual representation of the main entities and functions present in the Virtual Environment. It is in this context that this paper is set, and presents a study about the main characteristics of Augmented Reality, a comparative review of the most expressive technologies in this area, a conception of a library used in the development of this research, and an approach for the use of Augmented Reality in the implementation of advanced techniques of Interaction Interfaces. After a preliminary study of these themes, this work describes a support system developed for the interaction of Augmented Reality Environments based on optical capture of passive markers, using the Artoolkit library. The implemented prototypes to validate the support system developed are also being described by the analysis of the main aspects of its performance.

Keywords: Augmented Reality, Virtual Reality, Artoolkit, Interaction, Interfaces.

LISTA DE ILUSTRAÇÕES

Figura 2.1: Representação da progressão da RV	23
Figura 2.2: Imagem projetada diretamente na retina	25
Figura 2.3: Dispositivos de visualização	27
Figura 2.4: <i>Palm</i> com tecnologia virtual	27
Figura 2.5: Telefone celular com tecnologia virtual.....	28
Figura 2.6: Equipamento móvel de RA	31
Figura 2.7: Formas de oclusão.....	32
Figura 3.1: Imagem do funcionamento do Tinmith-evo5.....	34
Figura 3.2: Imagem do funcionamento do DWARF	35
Figura 3.3: Estrutura hierárquica do ImageTclAR	36
Figura 3.4: Imagem do funcionamento do STAR	38
Figura 3.5: Fases de funcionamento do ARToolKit.....	39
Figura 3.6: Imagem do funcionamento do VHD++	40
Figura 3.7: Exemplo do sistema ARCHEOGUIDE	41
Figura 3.8: Imagem do funcionamento do ARAS.....	42
Figura 3.9: Imagem do funcionamento do Studierstube.....	42
Figura 4.1: Arquitetura em camadas do ARToolKit	46
Figura 4.2: Montagem de um marcador	47
Figura 4.3: Reconhecimento de um marcador.....	47
Figura 4.4: Trecho de código para assimilação dos marcadores	48
Figura 4.5: Módulos de uma aplicação ARToolKit	49
Figura 4.6: Matriz identidade 4x4	51
Figura 4.7: Variáveis de translação	51
Figura 4.8: Variáveis de escala.....	52
Figura 4.9: Matrizes de rotação	52
Figura 4.10: Sentido positivo das rotações sobre os eixos de coordenadas	52
Figura 4.11: Coordenadas do marcador.....	53

Figura 4.12: Matrizes referente às coordenadas da câmera, marca e matriz final.....	54
Figura 4.13: Sistemas de coordenadas do ARToolKit	54
Figura 4.14: Situações comuns de oclusão.....	55
Figura 5.1: Usuário indicando objeto virtual distante	62
Figura 5.2: Projetor de um controle de interação	63
Figura 5.3: Ambiente ARWin	68
Figura 5.4: Interação por proximidade no ambiente ARWin	68
Figura 5.5: Ambiente do sistema de Química Aumentada.....	69
Figura 5.6: Cenário urbano virtual	70
Figura 5.7: Manipulação de um edifício virtual	70
Figura 6.1: Diagrama da arquitetura do ARISupport	73
Figura 6.2: Adição de imagens virtuais ao ambiente real	74
Figura 6.3: Tonalidade de cores em objetos virtuais.....	74
Figura 6.4: Objeto virtual transparente.....	75
Figura 6.5: Vídeo adicionado ao ambiente.....	75
Figura 6.6: Diversas formas geométricas	76
Figura 6.7: Teclas numéricas adicionadas a um marcador.....	76
Figura 6.8: Objeto virtual importado adicionado ao ambiente.....	77
Figura 6.9: Eliminação de superfícies em objetos virtuais.....	77
Figura 6.10: Função para encontrar a distância entre os centros dos marcadores.....	78
Figura 6.11: Interface de Controle 2D.....	79
Figura 6.12: Pseudocódigo do protótipo de Interface de Controle 2D.....	80
Figura 6.13: Imagem da placa do ARMouse.....	81
Figura 6.14: Interação com dispositivo não-convencional.....	81
Figura 6.15: Pseudocódigo do protótipo ARMouse	82
Figura 6.16: Vários marcadores espalhados para evitar oclusão.....	82
Figura 6.17: Componentes de interação utilizando marcadores.....	83
Figura 6.18: Ilustração de um teclado virtual.....	83
Figura 6.19: Trecho do código fonte do protótipo Teclado Virtual	84

Figura 6.20: Pseudocódigo do protótipo Teclado Virtual	84
Figura 6.21: Interação entre esferas.....	85
Figura 6.22: Interação entre esferas distantes dos marcadores.....	85
Figura 6.23: Pseudocódigo do protótipo Pinças Virtuais	86
Figura 7.1: Ambiente de trabalho para validação dos protótipos	87
Figura 7.2: Marcador utilizado para teste inicial.....	89
Figura 7.3: Gráfico de desempenho do teste de referência.....	90
Figura 7.4: Gráfico de desempenho da Interface de Controle 2D.....	91
Figura 7.5: Marcadores espalhados no interior de uma biblioteca	92
Figura 7.6: Modelo de teste para o protótipo ARMouse	92
Figura 7.7: Gráfico de desempenho do teste de marcadores	93
Figura 7.8: Modelo do teste do protótipo Teclado Virtual.....	96
Figura 7.9: Protótipo Teclado Virtual utilizado para teste	96
Figura 7.10: Avaliação do protótipo Teclado Virtual.....	96
Figura 7.11: Polígonos adicionados a cena	98
Figura 7.12: Avaliação do protótipo Pinças Virtuais	98
Figura 8.1: Futura implementação de uma Luva Virtual.....	103

LISTA DE TABELAS

Tabela I: Sistemas de RA	43
Tabela II: Medidas sugeridas.....	56
Tabela III: Características das versões do ARToolKit	57
Tabela IV: Critérios para um projeto de uma interface gráfica de interação	59
Tabela V: Critérios de desempenho.....	60
Tabela VI: Interação em sistemas de RV	67
Tabela VII: Resumo das tecnologias abordadas.....	71
Tabela VIII: <i>Softwares</i> utilizados	88
Tabela IX: Relação entre marcadores e a distância útil.....	94
Tabela X: Resumo dos protótipos	99

LISTA DE ABREVIATURAS E SIGLAS

2D	Bidimensional
3D	Tridimensional
3DS	<i>Three-Dimensional Studio</i>
ARAS	<i>Augmented Reality Aided Surgery</i>
ARCHEOGUIDE	<i>Augmented Reality-Based Cultural Heritage On-Site Guide</i>
ARISupport	<i>Augmented Reality Interaction Support</i>
ARMouse	<i>Augmented Reality Mouse</i>
ARToolKit	<i>Augmented Reality Tool Kit</i>
ASCII	<i>American Standard Code for Information Interchange</i>
AV	Ambientes Virtuais
AVI	<i>Audio Video Interleave</i>
BMP	<i>Bit Map</i>
DWARF	<i>Distributed Wearable Augmented Reality Framework</i>
GLUI	<i>Graphic Library User Interface</i>
HMD	<i>Head Mounted Display</i>
JPEG	<i>Joint Photographic Experts Group</i>
OpenGL	<i>Open Graphic Library</i>
PC	<i>Personal Computer</i>
RA	Realidade Aumentada
RAM	<i>Random Access Memory</i>
RM	Realidade Misturada
RV	Realidade Virtual
STAR	<i>Service and Training through Augmented Reality</i>
StbAPI	<i>Studierstube Application Programmer's Interface</i>
Tcl	<i>Tool Command Language</i>
Tk	<i>Tool Kit</i>
VA	Virtualidade Aumentada
VRML	<i>Virtual Reality Modeling Language</i>
WAV	<i>Windows Audio Volume</i>
WIMP	<i>Windows Icon Menu Pointer</i>
XML	<i>Extensible Markup Language</i>

SUMÁRIO

1. INTRODUÇÃO.....	16
1.1 Definição de Realidade Virtual.....	17
1.2 Importância das interfaces nos diversos sistemas de RA	20
1.3 Propósito do trabalho	21
1.4 Descrição dos capítulos	22
2. REALIDADE AUMENTADA	23
2.1 Definição e finalidade	23
2.2 Equipamentos necessários	24
2.3 Requerimentos encontrados em tecnologias de RA	29
2.3.1 Posicionamento	29
2.3.2 Foco	29
2.3.3 Registro.....	30
2.3.4 Luminosidade.....	30
2.3.5 Contraste	30
2.3.6 Mobilidade	31
2.3.7 Oclusão	32
3. PRINCIPAIS SISTEMAS DE RA.....	33
3.1 Tinmith-evo5	33
3.2 DWARF - Distributed Wearable Augmented Reality Framework.....	34
3.3 ImageTclAR.....	35
3.4 STAR - Service and Training through Augmented Reality.....	37
3.5 ARToolKit - Augmented Reality Tool Kit.....	38
3.6 VHD++	40
3.7 ARCHEOGUIDE - Augmented Reality-Based Cultural Heritage On-Site Guide.....	40
3.8 ARAS - Augmented Reality Aided Surgery.....	41
3.9 Studierstube - Environment for Collaboration in Augmented Reality	42
3.10 Uma comparação entre as tecnologias levantadas.....	43
4. UMA BIBLIOTECA PARA APLICAÇÕES DE RA: ARTOOLKIT	45
4.1 Definição e funcionamento	45
4.2 Estrutura interna de uma aplicação ARToolKit	49
4.3 Sistemas de coordenadas	51
4.4 Projeção em perspectiva	53
4.5 Limitações de sistemas de RA utilizando o ARToolKit.....	55
4.6 Compatibilidades e versões	56

5. INTERAÇÃO EM SISTEMAS DE RA	58
5.1 Tipos de interação	59
5.2 Técnicas de manipulação	60
5.3 Técnicas de seleção.....	61
5.3.1 Interação através de menus	62
5.4 Técnicas de navegação.....	64
5.5 Interação em sistemas imersivos.....	65
5.5.1 Interação direta	65
5.5.2 Interação com controles físicos.....	66
5.5.3 Interação com controles virtuais	66
5.6 Comparação do tipo de interação em sistemas de RV	66
5.7 Exemplos de sistemas de RA e comparação do tipo de interação com o usuário	67
5.7.1 ARWin - Gerenciador de Ambientes Operacionais.....	67
5.7.2 Química Aumentada: um ambiente de trabalho interativo e educacional	69
5.7.3 FingARTips: manipulação direta baseada em gestos em RA.....	70
5.8 Considerações sobre as tecnologias abordadas.....	71
6. BIBLIOTECA PARA INTERAÇÃO EM RA	72
6.1 Objetivos principais	72
6.2 Estruturação do suporte	73
6.3 Protótipos desenvolvidos	78
6.3.1 Interface de Controle 2D.....	79
6.3.2 ARMouse	80
6.3.3 Teclados Virtuais	82
6.3.4 Protótipo Pinças Virtuais	84
7. TESTE DOS PROTÓTIPOS E ANÁLISE DOS RESULTADOS OBTIDOS	87
7.1 Recursos utilizados	88
7.2 Testes realizados e resultados obtidos	89
7.3 Resumo dos protótipos apresentados.....	99
8. CONCLUSÕES	101
8.1 Aspectos gerais do trabalho	101
8.2 Contribuições.....	102
8.3 Pesquisas futuras.....	102
8.4 Comentários Finais	104
9. REFERÊNCIAS.....	105

1. INTRODUÇÃO

Atualmente, o alto grau de complexidade imposta por tarefas em diversas áreas da Ciência está exigindo mais do homem do que seus sentidos naturais podem lhe oferecer.

Dessa mesma maneira, é notado o crescente número de trabalhos que incorporaram o computador para o seu desenvolvimento, proporcionando, assim, um aumento no grau de dependência do homem à máquina. Sendo necessário a criação de ambientes mais interativos, a fim de que o homem minimize possíveis dificuldades na manipulação do computador e, por conseguinte, execute seu trabalho de modo mais eficiente e satisfatório.

Um ambiente interativo define-se como uma interface com o usuário, sendo formada por apresentações de informações, de dados, de controles e de comandos. É esta interface, também, que solicita e recebe as entradas de dados, de controles e de comandos. Finalmente, ela controla o diálogo entre as apresentações e as entradas. Uma interface tanto define as estratégias para a realização da tarefa, como conduz, orienta, recebe, alerta, ajuda e responde ao usuário durante as interações.

O principal requisito para um projeto de uma interface gráfica computacional é o favorecimento da tarefa de visualização, ou seja, dos meios que permitem ao usuário acessar o conteúdo do sistema. Um projeto de interfaces dispõe de instrumentos básicos muito semelhantes aos meios utilizados na arquitetura, como dimensões estéticas ou plásticas. Mas também podem ser considerados e classificados aspectos de investigação e inovação tecnológica, viabilização construtiva, racionalização do projeto, dos recursos e meios

disponíveis para sua execução, funcionalidade e o atendimento das necessidades básicas de seus usuários (TISSIANI et al., 2004).

Levando em consideração este contexto, na presente dissertação é proposta uma especificação de interfaces de interação em Ambientes Virtuais (AV).

1.1 Definição de Realidade Virtual

O termo Realidade Virtual (RV) possui diversas interpretações. Segundo Isdale (1998), RV é um conjunto de equipamentos específicos, como capacetes de visualização, dispositivos como luvas de interação e dispositivos de áudio, para interagir no AV, sendo considerada a interatividade em tempo real.

Segundo Burdea e Coiffet (2003), a RV é uma simulação na qual a computação gráfica é usada para criar uma visão de mundo real. No entanto, o mundo sintetizado não é estático e responde às entradas do usuário (gestos, comandos verbais, etc). Logo RV é a imersão total no AV, ao utilizador é completamente negado o acesso ao mundo real. O usuário pode interagir com o mundo e diretamente manipular objetos dentro desse ambiente sintético.

Existe uma grande diversidade de aplicações desenvolvidas que se utilizam das técnicas de RV, desde jogos e arquiteturas a planejamento de negócios. Muitas aplicações tentam simular com precisão as características de ambientes reais, como, por exemplo, a modelagem de edificações. Algumas aplicações fornecem formas de acrescentar perspectivas impossíveis no mundo real, tais como simuladores científicos, sistemas de telepresença ou controle de tráfego aéreo. Outras aplicações apresentam uma nova maneira para a experimentação da informação. Essas últimas aplicações podem ser definidas como complexas e são dedicadas a apenas um nível de interesse, como visualização do fluxo de informações do mercado financeiro ou navegar sobre uma grande base de dados de uma

corporação (ISDALE, 2004). Pode-se ainda considerar RV como a junção de três conceitos: imersão, interação e navegação.

O conceito de imersão está ligado ao sentimento de se estar dentro do ambiente. Um sistema imersivo é obtido com o uso de equipamentos de visualização, como, por exemplo, salas com projeções das visões diretamente expostas no recinto, como também dispositivos ligados aos outros sentidos, como som ou o tato.

O conceito de interação direciona-se à capacidade do computador de detectar as entradas do usuário e modificar instantaneamente o mundo virtual e as ações sobre ele, definindo uma capacidade de reação, ou seja, uma ação dirigida a um objeto define ao mesmo um novo comportamento (KIRNER, 1997).

O conceito de navegação em um ambiente 3D (tridimensional) caracteriza-se pelo deslocamento de um observador virtual dentro do modelo geométrico do cenário que se está interessado em visualizar (PINHO, 2000).

Junto à RV, está associado o conceito de AV. Estes ambientes englobam a RV, pois esta é a transposição de elementos e aplicações reais para este universo virtual. Os AV podem consistir em qualquer tipo de interação entre homem e computador, sendo criados a partir da idéia e imaginação do projetista. Este tipo de ambiente pode ter ou não referência ao mundo real com o qual o usuário pode interagir como se estivesse imerso nele.

As formas de interação são as mais diversas. Um exemplo desta diversidade é o fato de um participante de um AV poder trafegar por ele, caminhando ou flutuando.

O AV pode ser não-imersivo, semi-imerso e imersivo. Um AV não-imersivo é aquele em que a aplicação não envolve o usuário a ponto deste imaginar que esteja completamente presente dentro do ambiente. Normalmente, este tipo de aplicação não utiliza nenhum *hardware* dedicado a RV, tais como HMD (*Head Mounted Display*) e luvas.

O AV semi-imersivo é capaz de prover uma sensação de imersão maior do que o não-imersivo, através do emprego de uma ou mais telas de projeção ou de vários monitores. As telas são maiores do que o monitor, permitindo maior mobilidade do usuário. Podendo ser um benefício considerável em aplicações educacionais enquanto permite a experiência simultânea do AV. Nesse tipo de ambiente, o usuário começa a sair do mundo real e entrar no AV. Já no ambiente imersivo, os sentidos do usuário são aguçados intensamente em pelo menos uma categoria, para que ele tenha a sensação de estar realmente dentro do AV (COSTELLO, 1997).

O sentido da visão é o mais utilizado em RV. O estímulo da visão pode ser feito por intermédio de um HMD, que pode isolar o usuário do ambiente real. Normalmente, ele também pode proporcionar o estímulo dos sentidos auditivos (LUZ, 1997).

Uma interface de RV envolve um controle 3D altamente interativo de processos computacionais. O usuário entra no espaço virtual das aplicações, visualiza, manipula e explora os dados da aplicação em tempo real, usando seus sentidos, particularmente os movimentos naturais do corpo. A grande vantagem desse tipo de interface é que o conhecimento intuitivo do usuário a respeito do mundo físico pode ser transferido para a manipulação do AV. Para suportar esse tipo de interação, o usuário utiliza dispositivos não-convencionais como HMD, luva e outros. Estes dispositivos dão ao usuário a impressão de que a aplicação está funcionando no ambiente 3D real, permitindo a exploração do ambiente e a manipulação natural dos objetos com o uso das mãos, por exemplo, para apontar, pegar, e realizar outras ações (KIRNER, 1997).

1.2 Importância das interfaces nos diversos sistemas de RV

O projeto de uma interface entre um ser humano e um computador deve ser dirigido aos objetivos do usuário. Se os objetivos deste não forem a base do projeto de uma interface, então o usuário terá dificuldades em utilizá-la, fazendo com que a tarefa a ser realizada só possa ser executada com um esforço desnecessário, o que implica em maiores custos e maior consumo de tempo (COOPER, 1997).

A RV traz ao uso do computador um novo paradigma de interface com o usuário. Neste paradigma, o usuário não estará mais em frente ao monitor, mas sim, sentir-se-á dentro da interface. Com dispositivos especiais, a RV busca captar os movimentos do corpo do usuário, em geral braços, pernas, cabeça e olhos e, a partir destes dados, realizar a interação homem-máquina.

A RV constitui um salto qualitativo em relação às interfaces gráficas ora em utilização. Com a RV o usuário percebe, através de um ou mais de seus sentidos, dados vindos da máquina, gerados em dispositivos especiais, em uma simulação interativa. A interface de RV procura ser semelhante à realidade, buscando a sensação de presença em um ambiente informacional por meio de uma ilusão gerada por computador. Esta sensação de presença ou imersão constitui-se na principal característica da RV. A qualidade desta imersão, ou grau de ilusão, ou quão real esta ilusão parece ser, depende da interatividade e do grau de realismo que o sistema é capaz de proporcionar.

A interatividade é dada pela capacidade que o sistema tem de dar respostas às ações do usuário. Se o sistema responder de maneira instantânea, gerará no usuário o sentimento de que a interface está viva, criando assim, uma forte sensação de realidade. Por isto, a RV é um sistema computacional que deve utilizar-se de técnicas de tempo real para a interação com o usuário. O grau de realismo é dado pela qualidade destas respostas. Quanto mais parecida

com uma cena real for uma imagem apresentada ou um som emitido ao usuário, mais envolvido pelo sistema este usuário ficará (PINHO, 2000).

1.3 Propósito do trabalho

As manipulações de objetos em um AV são muitas vezes inábeis e inconvenientes. A falta de um retorno tátil, uma interferência no rastreamento, o comportamento carente de técnicas de interação e outros fatores podem fazer de uma simples tarefa, como segurar e mover um objeto virtual, uma experiência frustrante. Vários estudos têm focalizado a maneira como as pessoas manipulam objetos em um mundo real e em como ferramentas, locais de trabalho, entre outros, deveriam ser projetados para que se consiga uma manipulação efetiva. O desenvolvimento de aplicações de RV requer um entendimento de manipulações imersivas, em particular, qual ferramenta virtual e técnicas deveriam ser usadas e como elas deveriam ser projetadas para serem de uso simples e eficaz (POUPYREV et al., 1997).

Sendo a RV considerada a tecnologia de interface computacional mais avançada, pelo menos do ponto de vista da interatividade, deve-se garantir a sua funcionalidade e ainda projetar interfaces gráficas que possuam valor estético e funcional. A dificuldade no desenvolvimento de interfaces com usabilidade se deve ao fato delas constituírem fundamentalmente, sistemas abertos, probabilísticos, não determinísticos, sujeitos às influências do ambiente e às interpretações dos usuários. Suas entradas e saídas podem significar coisas diferentes para pessoas diferentes, em função de seu conhecimento, do momento, e do ambiente que as cercam.

O propósito desta dissertação é levantar os principais requisitos e tecnologias para o desenvolvimento de interfaces para AV, principalmente aquelas relacionadas à Realidade Aumentada (RA) contextualizada no próximo capítulo. Ao final deste levantamento é

apresentado um projeto para o estudo e implementação destes tipos de interface homem-máquina.

1.4 Descrição dos capítulos

Além desta introdução, a presente dissertação está organizada em mais oito capítulos, descritos a seguir.

No Capítulo 2 discorre-se sobre RA, suas definições e finalidades, equipamentos necessários e os desafios encontrados.

O Capítulo 3 apresenta um levantamento sobre os principais sistemas, bibliotecas e suportes para RA encontrados na literatura, uma breve descrição sobre o funcionamento, características principais e ao final, uma comparação entre os mesmos.

O Capítulo 4 aborda abrangentemente o funcionamento e as características gerais da biblioteca ARToolKit, bem como os princípios matemáticos envolvidos, o sistema de projeção das coordenadas e uma visão geral da sua estrutura.

As interações em sistemas de RA, os tipos de interações necessárias, as interações em sistemas imersivos, exemplos de sistemas em que a forma de interação com o usuário é exposta são os itens abordados no Capítulo 5.

O sexto Capítulo apresenta uma visão geral da arquitetura da biblioteca proposta, as funcionalidades providas e as implementações dos protótipos.

No Capítulo 7 descreve-se o ambiente de trabalho, os principais recursos de *hardware* e *software* utilizados, os testes realizados para cada protótipo e a análise dos resultados obtidos.

As conclusões, contribuições e pesquisas futuras são apresentadas no Capítulo 8.

Por fim, no Capítulo 9, são apresentadas as referências que forneceram a base teórica utilizada neste trabalho.

2. REALIDADE AUMENTADA

Neste capítulo, são abordados os conceitos de RA e os aspectos relativos a sua tecnologia, tais como: suas definições e finalidades, as tecnologias empregadas (equipamentos necessários) e os desafios encontrados.

2.1 Definição e finalidade

Milgram e Drasic (1996) descrevem em seu trabalho que entre o ambiente real e a RV é definida uma faixa conhecida como Realidade Misturada (RM), na qual a visão do mundo real é combinada gradativamente ao modo de visão do AV. Interagindo diretamente com a visão estereoscópica, por vídeo e por imagens gráficas, a RA fornece uma classe de padrões de vídeo que objetivam o realce ou o acréscimo de informações. A RA fornece uma sensação de envolvimento, adicionando elementos virtuais ao ambiente real. Como ilustrado na Figura 2.1, a RM refere-se à classe de todas as combinações entre o ambiente real e a RV.

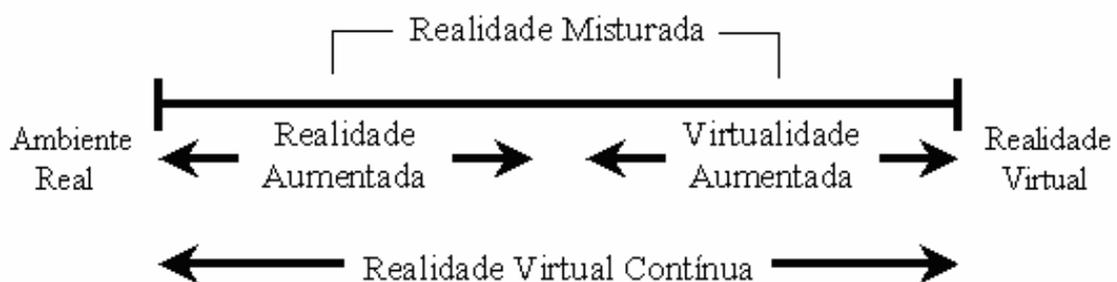


Figura 2.1: Representação da progressão da RV (MILGRAM & DRASIC, 1996)

Neste contexto, o significado do termo RA aplica-se quando o ambiente real é realçado ou complementado com imagens geradas por computador. Fazendo uso de um equipamento, como um HMD, é possível que uma imagem virtual de qualquer tipo, apareça para o usuário em um local espacial especificado. Estas imagens podem mostrar informações, ou servir como ferramentas interativas para o controle do ambiente.

Em contraste, a Virtualidade Aumentada (VA) representa que o AV é realçado por meio do acréscimo de imagens ou sensações advindas do mundo real. Este acréscimo de objetos vem de maneira direta à visão do usuário, fazendo com que ele possa ver seu próprio corpo como algo presente nas simulações geradas por computador, como um sistema típico de RV.

A VA pode também interagir com a RV com imagens estereoscópicas, quando, por exemplo, a vista de fora de uma janela virtual pode ser do mundo real em uma posição distante. Ferramentas de VA estão sendo desenvolvidas para solucionar as maiores limitações da RA, nas quais, tem-se a ausência do sentido de toque, por exemplo (MILGRAM & COLQUHOUN JR, 1999).

2.2 Equipamentos necessários

Descrito por Stradiotto (2002), um sistema de RV deve dar ao usuário a sensação de presença no AV e também permitir a interação deste usuário com o ambiente. Para proporcionar estas sensações, é necessária uma interface entre o usuário e o ambiente sintético. Ela deve ser capaz de, no mínimo, compreender os estímulos, passá-los para a forma de sinais digitais e fornecer uma resposta perceptível para o usuário. Esta transformação de sinais humanos para sinais digitais e vice-versa é feita pelo *hardware* periférico conectado à máquina.

Dependendo do tipo e formato de sinal e do tipo de tradução que o *hardware* específico fará durante a transferência de dados homem-máquina, uma pré-programação deve ser feita na máquina, indicando como serão processados os dados analógicos ou digitais provenientes do usuário e como retornarão estes dados.

Em sua maioria, as aplicações de RA são baseadas na estimulação do sentido da visão. Os dispositivos visuais e a qualidade de imagem gerada influenciam o nível de imersão em um sistema de RA. Cada tipo de dispositivo visual possui suas vantagens e desvantagens, mas todos têm o mesmo objetivo: combinar o ambiente real com o virtual.

Os equipamentos de exposições, como os HMD, foram tecnologias tradicionais de saída para aplicações de RA por mais de quarenta anos. Entretanto, ainda sofrem diversos inconvenientes tecnológicos e ergonômicos, que impedem que sejam usados eficazmente em diversas áreas de aplicações.

Os dispositivos de projeção de imagens diretamente na retina utilizam varreduras moduladas com feixes de *laser* de muita baixa potência, para criar a percepção da imagem virtual, que é desenhada diretamente na retina do usuário (Figura 2.2).

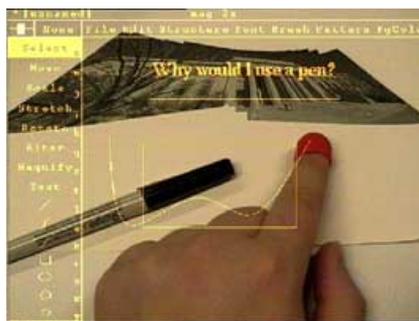


Figura 2.2: Imagem projetada diretamente na retina

Este tipo de dispositivo fornece maior potencial do campo de visão do que o previamente possível em um dispositivo como um HMD. Entretanto, algumas desvantagens adicionais podem ser identificadas para as versões existentes: somente imagens monocromáticas são apresentadas, ainda não existem *lasers* coloridos de baixa potência e de baixo custo (AZUMA et al., 2001).

O HMD de visão fechada inibe a visão completa do ambiente, fornecendo imagens reais através de câmeras de vídeo acopladas no exterior deste equipamento. A imagem projetada ao usuário é mostrada em tempo real e a imagem virtual é sobreposta ao vídeo. Um dos problemas com esta tecnologia de vídeo são os retardos dos quadros quando o usuário move seu eixo de visão rapidamente (BONSOR, 2004).

O funcionamento deste dispositivo é composto por um gerador de cenas que receberá os dados e fará os cálculos necessários, gerando imagens virtuais que são enviadas aos monitores dos óculos que, por sua vez, reproduzirão as imagens virtuais sobrepostas ao mundo real vistas através das lentes do visor do HMD. Um exemplo que ilustra essa tecnologia pode ser observado na Figura 2.3 (a). Opcionalmente, o sistema pode possuir um rastreador que captura os movimentos da cabeça do usuário (AZUMA, 1997).

O HMD de visão parcialmente transparente exposto na Figura 2.3 (b) permite que o usuário tenha uma visão direta do ambiente real, sendo apenas projetado nas lentes do HMD o objeto virtual. Normalmente esta visão não se torna nítida, por ter um teor opaco. Tal opacidade tem a finalidade de promover uma menor distinção do objeto virtual com o ambiente real.

O grau de processamento envolvido em um equipamento como este é menor que em outros sistemas de RA, pois somente é necessário consumir tempo de processamento com a geração das imagens virtuais para serem visualizadas, não necessitando da reprodução do mundo real. Quanto ao grau de resolução das imagens reais, este é o maior possível, pois a imagem é captada diretamente pelo olho do usuário (AZUMA, 1997).

A montagem de um sistema de projeções baseado em monitores é semelhante aos sistemas de RA por vídeo, com a diferença de que, ao invés do combinador de cenas enviar o resultado da combinação das imagens (reais e virtuais) para um HMD, ele irá transmiti-las para um monitor, como ilustrado na Figura 2.3 (c) (AZUMA, 1997).

Os óculos micro-ópticos ilustrados na Figura 2.3 (d) criam uma tela de vídeo virtual flutuando no espaço, na frente do usuário, por meio de um prisma encaixado na lente, refletindo a imagem de um pequeno vídeo colorido. Pode ser utilizado em aplicações nos quais se necessita de uma visão ampla, como, por exemplo, em simuladores de torre de controle de tráfego aéreo. A imagem da tela aparenta estar a cerca de 0,6 m do usuário e ocupa cerca de 10 graus (horizontais) de seu campo de visão (AZUMA, 2001).



a) HMD de visão fechada



b) HMD parcialmente transparente



c) Sistema baseado em monitor



d) Óculos micro-ópticos

Figura 2.3: Dispositivos de visualização

Atualmente, a tecnologia desenvolvida na área de computação móvel tem revolucionado a maneira de como são vistos os sistemas de computação. No passado, um computador era destinado apenas para o uso fixo em uma mesa de escritório. Hoje, cada vez mais, equipamentos podem ser carregados pelo usuário, como telefones celulares ou *palms* (Figura 2.4), para acessar as últimas informações e serem utilizados na rotina de trabalho.



Figura 2.4: Palm com tecnologia virtual (RASKAR & BIMBER, 2004)

Um recente telefone celular foi desenvolvido utilizando um sistema de RA no desenvolvimento de um jogo chamado “Caçada ao Mosquito Virtual” (Figura 2.5). Nele, o *software* tira uma fotografia através da câmera embutida na parte traseira do telefone e a imagem é processada em tempo real. A imagem virtual aparece, na tela do telefone celular e os mosquitos virtuais são projetados nela. O objetivo do jogo é caçar, em várias direções, todos os mosquitos (CeBIT, 2003).



Figura 2.5: Telefone celular com tecnologia virtual (CeBIT, 2003)

Tal tecnologia se distingue das outras por possuir a característica de ser móvel. Ao mesmo tempo, também apresenta a característica de ser de baixo custo, se comparado com os computadores atuais. Porém, para estes equipamentos encontram-se limitações de recursos no que se refere à baixa capacidade de processamento de imagens, atrasos nas taxas de quadros geradas, além de um campo de visão limitado ao tamanho do visor do dispositivo (AZUMA et al., 2001). Em contraste com os equipamentos acoplados ou carregados pelo usuário, como um HMD ou celulares, tecnologias de projeções espaciais integram-se ao ambiente real, explorando assim um maior espaço de visão. Em diversas situações, estas projeções podem superar limitações tecnológicas e ergonômicas dos sistemas convencionais de RA (RASKAR & BIMBER, 2004).

2.3 Requerimentos encontrados em tecnologias de RA

Assim como há diversas tecnologias de visualização empregadas em sistemas de RA, cada uma com suas características e tipos específicos de projeções, encontrou-se para as mesmas, alguns tipos de desafios que serão apresentados sucintamente nos próximos itens.

2.3.1 Posicionamento

A RA com dispositivos de visão limitada ou por vídeo possuem problemas quanto à utilização de câmeras. A qualidade da imagem depende da resolução da câmera que captura as imagens do mundo real. Além disso, a utilização de somente uma câmera para desempenhar a função dos olhos anula a sensação de profundidade. O ideal, então, é usar duas câmeras. Neste caso, porém, surge um outro problema, que é como acertar o posicionamento entre as câmeras de forma a se obter a mesma distância entre as pupilas de um usuário. Não obstante, essa distância varia de acordo com as características individuais de cada usuário, aumentando a complexidade ao se utilizar este tipo de sistema (AZUMA, 1997).

2.3.2 Foco

Segundo Azuma (1997), o foco pode ser um problema para aproximações ópticas e de vídeo. Idealmente, o virtual deve combinar-se com o real. Em um sistema baseado em vídeo, a imagem virtual e real combinada será projetada na mesma distância pelo monitor ou pelo sistema óptico do HMD. Entretanto, dependendo dos ajustes de profundidade do campo de visão e do foco da câmera de vídeo, as partes do mundo real podem não estar no foco.

No caso óptico, a imagem virtual é projetada a alguma distância do usuário. Esta distância pode ser ajustável, embora seja frequentemente fixa. Conseqüentemente, quando os objetos reais estiverem em distâncias variáveis do usuário, todos os objetos virtuais serão projetados à mesma distância. Se as distâncias virtuais e reais não forem combinadas para os

objetos que o usuário está olhando, pode não ser possível ver claramente a ambos simultaneamente.

2.3.3 Registro

Para o correto funcionamento de um sistema de RA, é imprescindível que objetos reais e virtuais estejam devidamente alinhados para que se tenha uma perfeita ilusão de coexistência dos dois ambientes. Esse alinhamento é chamado de registro e tem como objetivo informar ao sistema sobre a posição e as dimensões de objetos reais que irão compor o cenário visualizado (AZUMA, 1997).

2.3.4 Luminosidade

Uma limitação imposta por aplicações que utilizam dispositivos com combinadores ópticos de visão parcial ou total é o grau de luminosidade no cenário real. A maioria dos combinadores reduz a quantidade de luz vinda do mundo real. Contraditoriamente, em ambientes com muita iluminação, o grau de qualidade das imagens virtuais fica prejudicado, fazendo com que elas pareçam semi-transparentes.

2.3.5 Contraste

O contraste é um outro exemplo de desafio por causa da grande escala dinâmica entre ambientes reais e o que o olho humano pode detectar. Idealmente, o brilho dos objetos reais e virtuais deve ser apropriadamente combinado. Se o ambiente real for muito claro, ilumina demasiadamente a imagem virtual. Se o ambiente real for muito escuro, a imagem virtual pode não aparecer no mundo real (AZUMA, 1997).

2.3.6 Mobilidade

Algumas aplicações de RA, precisam fornecer ao usuário um suporte caso haja a necessidade deste se movimentar. As aplicações requerem que o usuário esteja realmente no lugar onde a tarefa deve ocorrer. Portanto, alguns sistemas de RA devem prover a característica da mobilidade (habilidade de andar ao redor e ao ar livre), com o usuário estando afastado dos ambientes controlados.

O maior problema desse tipo de sistema é o desconforto, devido ao excesso de aparelhos que devem ser carregados junto ao corpo, pois os equipamentos de RA ainda são pesados, apesar de já estarem sendo desenvolvidos dispositivos para evitar ou reduzir esses problemas. A Figura 2.6 ilustra um sistema de RA que demonstra essa característica.



Figura 2.6: Equipamento móvel de RA

De acordo com Luz (1997), os equipamentos específicos ainda estão bem longe de serem confortáveis, principalmente para que o usuário os utilize durante horas, da forma como utiliza atualmente um monitor, um teclado e um *mouse*. O que há de disponível, a baixo custo, como as luvas e o HMD, ainda são desconfortáveis. Deve-se acrescentar que o HMD, segundo alguns estudos, geralmente causa mal estar na maioria dos usuários quando utilizado por um tempo superior a vinte minutos.

2.3.7 Oclusão

Descrita por Poupyrev et al. (1997), a oclusão influencia o desempenho do usuário de duas maneiras. Primeiramente, quando um objeto está ocluído, sua dimensão visual diminui. A esse respeito, a seleção de um objeto ocluído é a mesma que a seleção de um objeto não-ocluído com tamanho visível menor. Em segundo lugar, a oclusão obstrui parcialmente o acesso ao objeto-alvo e dificulta seu acesso, sendo que a proximidade de outros objetos aumenta a probabilidade de que haja a seleção do objeto errado. Os casos de oclusão do ponto de vista do usuário são ilustrados pela Figura 2.7:

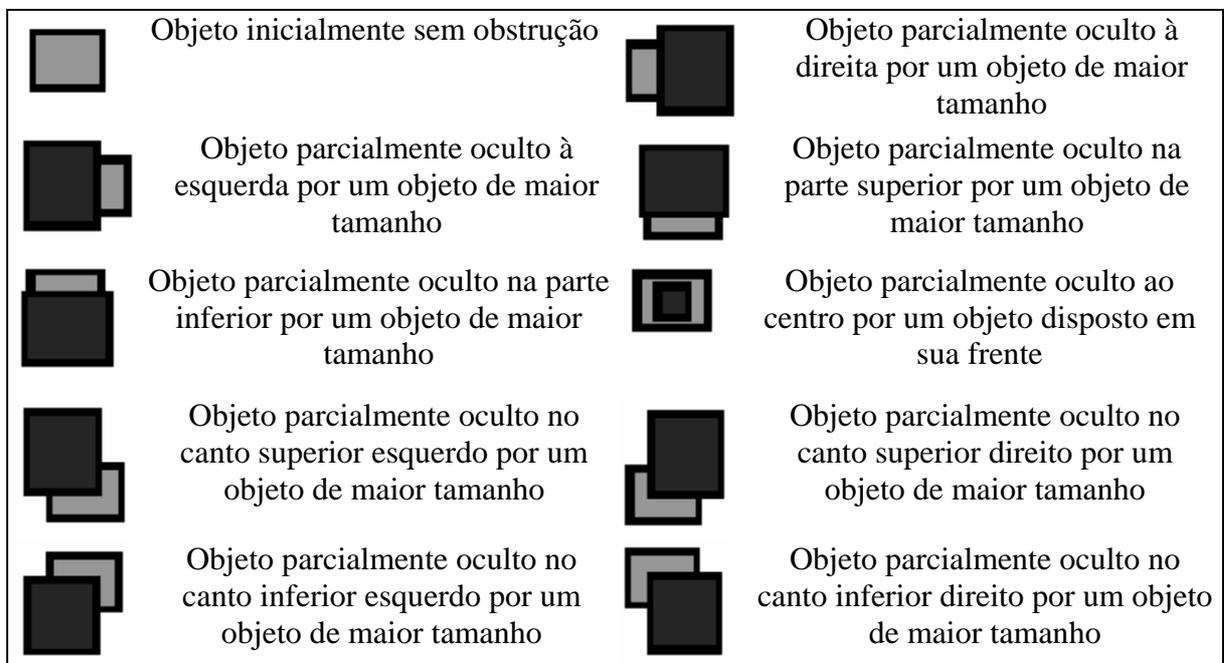


Figura 2.7: Formas de oclusão

Após a resolução dos problemas básicos com aplicações de RA, o objetivo final é gerar objetos virtuais que são virtualmente indistinguíveis do ambiente real (AZUMA, 1997).

3. PRINCIPAIS SISTEMAS DE RA

Para o desenvolvimento de aplicações de RA, é fundamental considerar o aspecto da sensação de tempo real. Para isso, o ambiente de desenvolvimento deve apresentar características como: prover desempenho no processamento, a confiabilidade dos equipamentos a serem utilizados e um sistema operacional adequado. Um sistema de RA, uma vez finalizado, deverá ser testado e validado, a fim de averiguar se o resultado reflete o realismo esperado. Alguns deles são apresentados a seguir.

3.1 Tinmith-evo5

A versão descrita por Piekarski & Thomas (2001), do Tinmith-evo5 é baseada em um projeto completamente novo, orientado a objetos e desenvolvido em linguagem C++ pela Universidade do Sul da Austrália. Os objetivos principais são: o aumento do desempenho de velocidade e da eficiência das aplicações de RA.

O sistema Tinmith-evo5 foi projetado para implementar pesquisas de RA sem barreiras físicas. Para este sistema foram desenvolvidas técnicas de interface para ambientes 3D, nos quais equipamentos como teclado e *mouse* não poderiam ser usados. Usando essas interfaces, foram desenvolvidas aplicações complexas que permitem aos usuários interagirem com seu ambiente, provendo a incorporação de novas informações no sistema sobre o mundo real.

A arquitetura do Tinmith-evo5 inclui atualmente uma base para menus de controle, seleção e manipulação de objetos. O painel do menu é acoplado ao fundo da tela de vídeo, e sua visualização não depende de onde as mãos ou a cabeça do usuário estejam localizadas. Este ambiente é inteiramente controlado usando-se apenas o menu e as luvas, não havendo nenhum teclado ligado ao sistema como mostra a Figura 3.1.



Figura 3.1: Imagem do funcionamento do Tinmith-evo5 (PIEKARSKI & THOMAS, 2001)

3.2 DWARF - Distributed Wearable Augmented Reality Framework

A estrutura do DWARF, desenvolvido pela Universidade Técnica de Munique (BRUEGGE & KINKLER, 2001), é baseada no conceito de serviços colaborativos distribuídos (Figura 3.2). Os serviços são interdependentes e expõem suas exigências, chamadas *Needs*, e ofertas, chamadas *Abilities*, com a ajuda de gerentes de serviço. Em cada nó de rede de um sistema DWARF, há um gerente de serviço, não havendo nenhum componente central. Um gerente de serviço controla seus serviços locais e mantém as descrições destes. Cada gerente de serviço coopera com o outro na rede para ajustar as conexões entre os serviços. Cada serviço tem uma descrição XML (*Extensible Markup Language*) de suas *Abilities*, *Needs* e conexões (protocolos de comunicação).



Figura 3.2: Imagem do funcionamento do DWARF (BRUEGGE & KINKLER, 2001)

Cada *Abilitie* possui certos atributos que descrevem os parâmetros de qualidade desse serviço. Do mesmo modo, cada *Need* especifica um atributo sobre a qualidade do serviço que espera. Este atributo é usado pelo gerente de serviço para selecionar as habilidades que podem fornecer uma qualidade suficiente do serviço para satisfazer a uma necessidade especificada. Este atributo pode também ser usado em tempo de execução para assegurar-se que a qualidade desejada do serviço seja fornecida (BRUEGGE & KINKLER, 2001).

3.3 ImageTclAR

O ImageTclAR, desenvolvido na Universidade de Michigan, é um ambiente dirigido ao desenvolvimento de aplicações de RA, tendo como uma de suas características prover aos programadores com diferentes níveis de habilidades de programação, formas simplificadas e avançadas ao projetarem aplicações de RA, providas por funções deste ambiente (OWEN et al., 2003).

A flexibilidade oferecida pela linguagem permite que os programadores inexperientes desenvolvam aplicações de RA sem se envolverem com códigos complexos, utilizando simples trechos de *scripts*. Fornece, ainda, aos programadores mais experientes, os meios para implementar e desenvolver componentes utilizando a linguagem C++ e funções da biblioteca gráfica OpenGL (*Open Graphic Library*).

O ImageTclAR é uma extensão do ambiente de desenvolvimento do algoritmo de multimídia ImageTcl. É baseado nas ferramentas: Tcl (*Tool Command Language*) e Tk (*Tool Kit*). Logo, o sistema foi projetado para prover um ambiente de programação para o desenvolvimento de uma aplicação de RA, oferecendo características como a interpretação de *scripts* ou a geração de códigos compilados. A Figura 3.3 ilustra os componentes deste ambiente de programação.

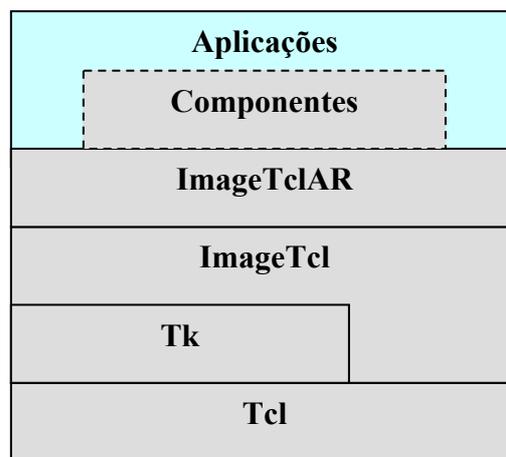


Figura 3.3: Estrutura hierárquica do ImageTclAR (OWEN et al., 2003)

A Tcl é uma linguagem simples, de fácil entendimento e utilização e é direcionada para programadores sem experiência de desenvolvimento em programação. Essa linguagem pode ser utilizada interativamente, fornecendo ao programador um ambiente com facilidades de aprendizagem e implementação. Também suporta trechos de programas desenvolvidos em partes separadas, como *scripts*, além de oferecer recursos de programação estruturada em procedimentos.

A Tk é uma versão estendida da linguagem Tcl, que fornece ao desenvolvedor de uma aplicação de RA os recursos necessários para a criação de interfaces amigáveis, trata-se de uma forma simples de programação para criar interfaces tais como: menus, botões, caixas de diálogos ou qualquer outro componente visual.

O ImageTcl é uma outra versão estendida da linguagem Tcl que fornece um ambiente de desenvolvimento multimídia, como por exemplo, componentes básicos de leitura e

composição de imagens e sons, direcionamento e processamento de informações armazenadas em componentes e suporte para desenvolvimento de aplicações gráficas.

O ImageTclAR é um sistema desenvolvido a partir das composições da linguagem ImageTcl, acrescido de características direcionadas ao desenvolvimento de aplicações de RA.

Estas composições definem o sistema de desenvolvimento da linguagem ImageTclAR, além de integrar funções de suporte aos dispositivos de RA, como HMD, câmeras e procedimentos para calibração de posições (OWEN et al., 2003).

3.4 STAR - Service and Training through Augmented Reality

O STAR é um sistema de RA, desenvolvido por institutos de pesquisas europeus, como a Universidade de Genebra, e corporações americanas, como a Siemens. Segundo Badiqué e Raczynski (2002), o sistema foi projetado com finalidades industriais, permitindo que informações adicionais sejam observadas em tempo real. A informação varia de textos de ajuda simples à geometrias 3D complexas. Com a ajuda de técnicas de RA, é possível combinar estas informações de acordo com o movimento de câmeras. O resultado é uma extensão da realidade. O foco do sistema STAR é o treinamento, a documentação e o planejamento de projetos industriais. Para o sistema STAR foram desenvolvidos os seguintes componentes:

1. Reconstruções 3D de instalações industriais;
2. Seleção do ponto de visão e da câmera;
3. Manipulação de objetos combinados por humanos virtuais; e
4. Aproximações com diferentes rastreadores e reconhecimento dos objetos do vídeo.

Empregando-se estes elementos, realiza-se a seguinte simulação a título de exemplo: um trabalhador está em determinado local e se prepara para executar uma tarefa de manutenção. Para sua assistência, possui um *laptop* equipado com uma câmera e uma

conexão sem fio à uma rede local de computadores. As imagens da câmera filmam o espaço de trabalho em que o trabalhador está operando. A imagem de vídeo da câmera é enviada pela rede a um perito. A posição 3D da câmera com a cena é determinada automaticamente pelo sistema nas formas originais encontradas nas imagens. Assim, o perito pode acrescentar ao vídeo uma variedade de informações, como textos e imagens. A imagem aumentada é emitida pela rede ao *laptop* do trabalhador que usará estas informações para decidir que etapas executar em seguida (Figura 3.4).

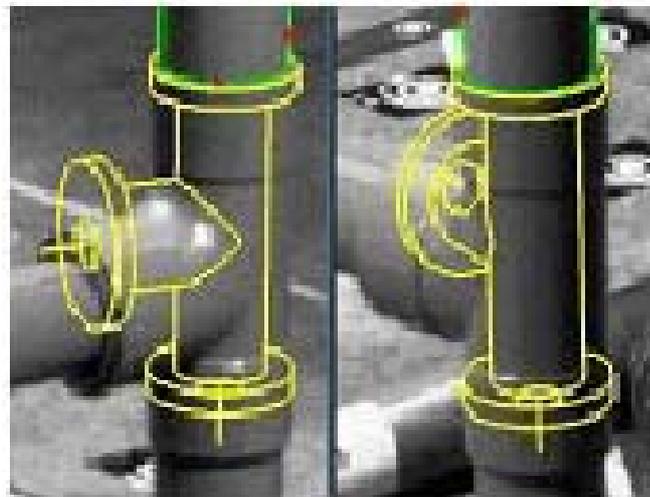


Figura 3.4: Imagem do funcionamento do STAR (BADIQUÉ & RACZYNSKI, 2002)

3.5 ARToolKit - Augmented Reality Tool Kit

O ARToolKit é um conjunto de bibliotecas desenvolvidas na Universidade de Washington, usando a linguagem de programação C. Foi projetado para o desenvolvimento rápido de aplicativos de RA, provendo técnicas de visão computacional para calcular a posição e a orientação de marcadores ou padrões impressos em cartões, sendo esta posição e orientação capturados por uma câmera digital ou um HMD, de forma que estes marcadores venham a ser corretamente revestidos por objetos virtuais 3D.

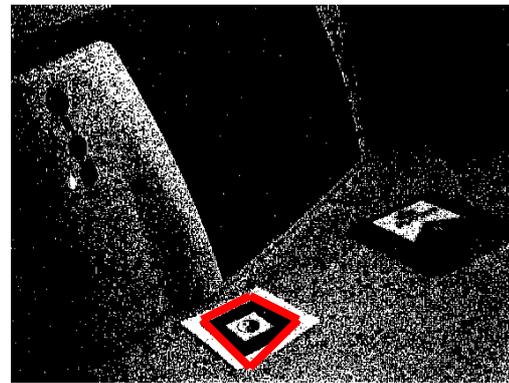
Descrito por Kato (2004), o ARToolKit utiliza-se de técnicas de visão computacional para calcular o ponto de vista da câmera em relação a um marcador existente no mundo real.

Neste sistema, cada quadro de imagem do ambiente capturado é binarizado para facilitar o reconhecimento e a posição espacial do marcador a ser localizado. Feito isso, para cada imagem de vídeo, o desenho no interior do marcador reconhecido é comparado com gabaritos de marcadores pré-definidos. Para cada quadro, o marcador é sobreposto por uma imagem.

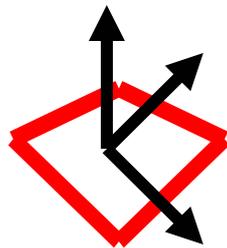
A Figura 3.5 (a)-(d), ilustra a forma como o ARToolKit projeta a imagem virtual sobre um marcador. A imagem gerada pode ser de um objeto desenvolvido dentro de uma aplicação utilizando bibliotecas gráficas como OpenGL, ou importando objetos desenvolvidos em linguagem VRML (*Virtual Reality Modeling Language*).



a) Ambiente real



b) Localização do marcador



c) Posição espacial



d) Imagem virtual sobre o marcador

Figura 3.5: Fases de funcionamento do ARToolKit

3.6 VHD++

Desenvolvido pelo Instituto Federal de Tecnologia Suíço e pela Universidade de Genebra, o ambiente VHD++, após diversos anos de intensas pesquisas, projetos, e dos esforços em seu desenvolvimento, encontra-se atualmente em sua fase de validação. De acordo com Ponder et al. (2003), foi projetado como uma forte estrutura orientada a objetos, e desenvolvida inteiramente em linguagem C++. Consiste atualmente em aproximadamente 500 classes agrupadas em aproximadamente 35 componentes.

A estrutura deste sistema fornece aos programadores uma aplicação eficaz direcionada a simulações habituais, sendo caracterizada como uma tecnologia avançada no desenvolvimento de simulações humanas.

Em consequência de sua arquitetura de grande escala e da reutilização do código (componentes), o sistema provê uma forma de desenvolvimento eficiente para aplicações virtuais robustas. A Figura 3.6 ilustra a interação do usuário com o AV em uma simulação médica:



Figura 3.6: Imagem do funcionamento do VHD++ (PONDER et al., 2003)

3.7 ARCHEOGUIDE - Augmented Reality-Based Cultural Heritage On-Site Guide

Projetado pelo Centro de Computação Gráfica de Darmstadt na Alemanha, o ARCHEOGUIDE é um sistema que introduz o estudo da arte em tecnologias de visualização e de computação móvel, empregado na ilustração de recomposições de informações

históricas. Ele proporciona aos usuários percursos personalizados de reconstruções virtuais de ruínas de elevado interesse cultural, conforme ilustra Figura 3.7.



Figura 3.7: Exemplo do sistema ARCHEOGUIDE (GLEUE & DÄHNE, 2001)

Os três componentes fundamentais do ARCHEOGUIDE são: as unidades computacionais portáteis, transportadas pelos visitantes durante os percursos na área arqueológica, redes de comunicação e uma base de dados central. A base de dados é o repositório central do material multimídia (imagens virtuais) que são apresentados ao usuários durante o percurso entre as ruínas (GLEUE & DÄHNE, 2001).

3.8 ARAS - Augmented Reality Aided Surgery

Desenvolvido por Splechtna et al. (2002), ARAS é uma aplicação para ajudar o médico cirurgião em operações, indicando dados tomográficos e dados de ultra-som, como objetos 3D, desenhados na superfície (região de interesse) do paciente. O médico cirurgião tem a vantagem de uma visualização detalhada do órgão e de sugestões adicionais colocadas pelo radiologista, permitindo a localização exata da patologia.

Este sistema apresenta aos profissionais de saúde uma forma de avaliar a situação do paciente, planejar ou definir a estratégia pré-operatória e possibilitar uma avaliação precisa dos resultados pós-operatórios (Figura 3.8).



Figura 3.8: Imagem do funcionamento do ARAS (SPLECHTNA et al., 2002)

3.9 Studierstube - Environment for Collaboration in Augmented Reality

Studierstube é um ambiente para o desenvolvimento de aplicações colaborativas de RA. Desenvolvido no Instituto de Computação Gráfica da Universidade de Tecnologia de Viena, o conceito deste ambiente baseia-se em um conjunto particular de equipamentos como dispositivos de apresentação e interação, assim como uma estrutura de *software* chamada StbAPI (*Studierstube Application Programmer's Interface*). A estrutura StbAPI é composta por um conjunto de classes da linguagem C++, programada com as melhores ferramentas gráficas da linguagem *Open Inventor*.

Neste projeto, o usuário, utilizando um painel personalizado e um HMD transparente (Figura 3.9), pode interagir com um experimento e ao mesmo tempo visualizar os demais colaboradores, podendo, desta forma, conversar, apontar e gesticular a fim de discutir o fenômeno observado (SZALAVÁRI et al., 1996).



Figura 3.9: Imagem do funcionamento do Studierstube (SZALAVÁRI et al., 1996)

3.10 Uma comparação entre as tecnologias levantadas

Muitos trabalhos encontrados na literatura dedicam-se a introduzir novas técnicas de interação ou descrever configurações particulares de *hardware* e *software*, assim como seu uso em situações distintas. A Tabela I mostra um resumo dos ambientes apresentados neste capítulo, sendo eles plataformas de desenvolvimento, bibliotecas que oferecem flexibilidade na concepção de sistema de RA, ou arquiteturas compostas de equipamentos específicos para sua utilização, assim como as interfaces de interação providas por esses ambientes.

Tabela I: Sistemas de RA

Nome do Sistema	Ano	Qualificação	Características da Interface de Interação do Usuário
ARToolKit	1999	Biblioteca de desenvolvimento	Não oferece serviços diretos de interação, apenas valores de rastreamento do espaço visual baseados em marcadores.
DWARF	2000	Plataforma de serviços colaborativos	Ferramenta multi-usuário de interação, tendo o usuário interagindo como moderador de informações.
ImageTclAR	2000	Biblioteca de desenvolvimento	Ferramenta mono-usuário que oferece suportes a rastreadores e provê adição de módulos (componentes para a aplicação) como importar objetos 3D.
STAR	2000	Sistema completo de treinamento em aplicações industriais	Sistema multi-usuário caracterizando-se pelos equipamentos móveis, câmera e redes sem fio, apenas captura a imagem, envia para o técnico, recebendo instruções para continuidade do trabalho.
Archeoguide	2001	Dispositivo móvel	Aplicação mono-usuário, composta de um computador portátil onde as informações são dispostas, equipamentos de navegação (interações por dispositivos físicos como teclado) e um HMD (traje especial).
Studierstube	2001	Plataforma de desenvolvimento de projetos colaborativos	Plataforma multi-usuário composta por dispositivos como HMD, tendo como ferramentas de interação um painel (adicionado de informações como controles virtuais para navegação e seleção) e uma caneta para seleção.
ARAS	2002	Sistema de guia cirúrgica por imagem	Sistema multi-usuário ilustrativo, disposto de câmeras espalhadas no recinto cirúrgico, tendo o médico cirurgião usando um HMD e um radiologista recebendo informações em um vídeo, analisando e retornando informações.
Tinmith-evo5	2002	Sistema completo (Aplicação móvel)	Aplicação mono-usuário, baseada em um sistema de navegação, HMD e luvas específicas para visualização e interação com a informação. A interface possui um suporte por menus de controles, seleção e manipulação do objeto virtual.
VHD++	2003	Biblioteca de desenvolvimento de simulações humanas	Tecnologia mono-usuário de simulação humana em tempo real por avatares, com comandos reconhecimento de voz, animação comportamental dos avatares, possui uma camada de navegação por sensores magnéticos.

Nesta tabela, foram apresentadas diferentes tecnologias de interação em ambientes de RA e o contexto de suas utilizações. Devido às suas características, propriedades e funções, os sistemas como DWARF (item 3.2), STAR (item 3.4), ARCHEOGUIDE (item 3.7), ARAS (item 3.8), Tinmith-evo5 (item 3.1) se tornam carentes no desenvolvimento de novas aplicações, pois são integrados com equipamentos específicos para seu funcionamento, dirigidos a somente uma forma de utilização.

A biblioteca VHD++ (item 3.6) é fechada a uma quantidade de simulações restritas, implementadas em sua arquitetura. A biblioteca ImageTclAR (item 3.3) aparentemente apresenta-se como um projeto descontinuado, não encontrando atualização de novas versões como também novas pesquisas.

A plataforma Studierstube (item 3.9) utiliza-se da biblioteca ARToolKit como suporte de reconhecimento em algumas de suas aplicações, bem como de outros utilitários para seu funcionamento.

Através das arquiteturas levantadas, objetiva-se a implementação de interfaces para utilização em sistemas de RA, sendo que a ferramenta de desenvolvimento para as aplicações de interação será feita através da biblioteca ARToolKit (item 3.5), que se caracteriza como uma inovação mundial, multiplataforma. Novas versões dessa biblioteca têm sido desenvolvidas em vários países e os ambientes gerados são de alto desempenho, além dos equipamentos possuírem baixo custo. Outro motivo para a utilização da biblioteca ARToolKit é que ela se apresenta como *open source*, distribuída gratuitamente, e que a linguagem C da qual se utiliza é considerada dentre as opções existentes, como a mais adequada por sua eficiência e robustez, com a característica de rápido processamento e ótima portabilidade. Outras características desta tecnologia serão abordadas no capítulo seguinte.

4. UMA BIBLIOTECA PARA APLICAÇÕES DE RA: ARTOOLKIT

Neste capítulo, são abordadas algumas das características e estruturas da biblioteca de desenvolvimento de aplicações de RA, ARToolKit. Como também uma visão geral de seu funcionamento.

4.1 Definição e funcionamento

O ARToolKit é uma biblioteca projetada em linguagem C que permite aos programadores desenvolver facilmente aplicações de RA. Uma das partes mais difíceis do desenvolvimento de uma aplicação em RA é calcular precisamente o ponto de vista do usuário em tempo-real para que imagens virtuais estejam exatamente alinhadas com os objetos do mundo real.

O ARToolKit usa técnicas de visão computacional (captar a informação relativa à cena) para calcular a posição real da câmera e orientação relativa a cartões marcadores, permitindo ao programador sobrepor objetos virtuais sobre estes cartões.

Os requisitos básicos de *hardware* para desenvolver e executar aplicações do ARToolKit são: uma câmera de vídeo digital, um monitor para a visualização das imagens e um dispositivo de aquisição de vídeo (placas gráficas) com seus respectivos *drivers* (KATO, 2004).

A biblioteca ARToolkit compõe sua estrutura nas seguintes camadas mais significativas:

Ar.lib: apresenta-se como responsável por fornecer funções referentes aos cálculos matemáticos através do reconhecimento dos marcadores, como matrizes, valores de distorção de imagem, além de outras funções.

libARgsub: responsável pela configuração do ambiente de vídeo para apresentação das imagens 3D utilizando-se da biblioteca OpenGL.

Arvml: utiliza-se esta camada na importação de objetos VRML.

libARvideo: responsável por exibir a imagem (criar janela), bem como fornecer funções de filtros de vídeo, como exemplo utilizado na binarização dos quadros.

A disposição desta estrutura básica utilizada no ARToolKit é mostrada na Figura 4.1.

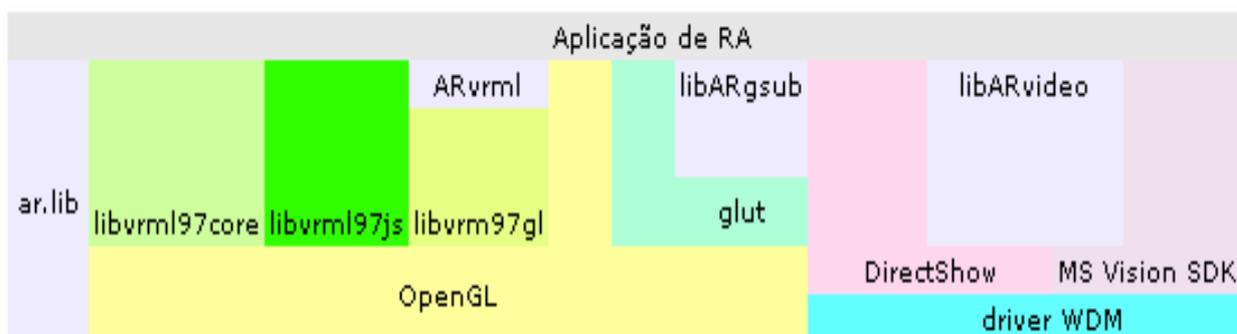


Figura 4.1: Arquitetura em camadas do ARToolKit (KATO, 2004)

Inicialmente, para o desenvolvimento de uma aplicação de RA utilizando o ARToolKit, é necessário criar um marcador, ou seja, uma imagem para reconhecimento e sobreposição do objeto virtual. Para isso, o ARToolKit fornece juntamente com sua instalação uma imagem padrão, em forma de um quadro de bordas pretas e interior vazio. Esse quadro necessita ser editado por algum programa gráfico, no seu interior deverá ser inserido uma ilustração. A Figura 4.2 explana esta etapa.

Esta ilustração deverá ser única em qualquer orientação, assimétrica e sem muitos detalhes, deve-se evitar também o desenvolvimento de marcadores muito semelhantes.

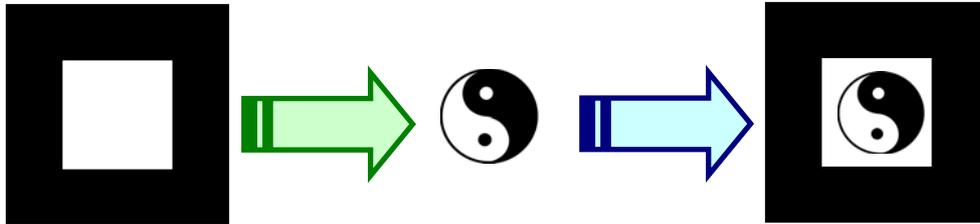


Figura 4.2: Montagem de um marcador

Após a criação do marcador é necessário que a aplicação de RA assimile-o, através da execução de um utilitário de reconhecimento, o usuário posiciona o marcador impresso em frente à câmera. O utilitário filma o marcador com a borda esquerda e inferior desenhada na cor verde, e a borda direita e superior na cor vermelha (Figura 4.3), informando ao usuário que o marcador foi reconhecido e está pronto para ser armazenado.

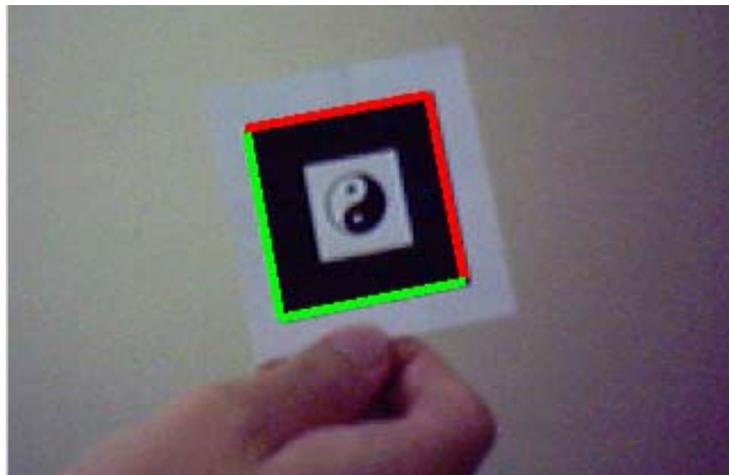


Figura 4.3: Reconhecimento de um marcador

O resultado final deste reconhecimento é um arquivo texto com valores que vão de 0 a 255. As partes mais claras da imagem representam os valores mínimos e os maiores valores, as partes escuras. Para um melhor reconhecimento, estes valores dependem da iluminação inicial que deverá permanecer semelhante à iluminação final. Este arquivo deve acompanhar a aplicação de RA.

Além desta ferramenta de identificação de marcadores, o ARToolKit possui ainda utilitários para calibração da câmera, além de diversos marcadores prontos para impressão e de exemplos de sistemas de RA.

O próximo passo é a associação do arquivo contendo o marcador reconhecido com o programa de RA. Para isso, um novo arquivo com qualquer editor de texto padrão ASCII (*American Standard Code for Information Interchange*) deve ser empregado. Esse arquivo pode conter uma ou mais marcações. É nele que é definido qual marcador corresponderá ao objeto virtual. A Figura 4.4 ilustra esta estrutura. Esse arquivo também deve acompanhar a aplicação de RA.

```
#Número de Marcadores a serem Reconhecidos
1

# Nome do Arquivo gerado pelo programa de reconhecimento do marcador
MarcaA
# Nome utilizado no programa de RA para associação com o marcador
PontoA
# Escala: Tamanho do Marcador (milímetros)
10
# X e Y coordenadas referentes ao centro do marcador
0 0
```

Figura 4.4: Trecho de código para assimilação dos marcadores

Ao configurarem-se os passos descritos, um programa de RA estará apto a ser executado. Como exposto resumidamente no item 3.5, inicialmente a imagem de vídeo é capturada e convertida em uma imagem binária (preta ou branca), nesta imagem, então, se inicia uma busca por linhas paralelas.

Para cada quadro encontrado, a imagem inserida no interior é identificada e comparada com o arquivo de marcadores que acompanha a aplicação. Se existe uma porcentagem (de média a alta) de igualdade, então o ARToolKit encontrou um dos marcadores de rastreamento de RA.

Usa-se o tamanho, a posição e a orientação do marcador encontrado para calcular a posição de sobreposição do objeto virtual.

O próximo item descreverá o funcionamento interno de uma aplicação de RA utilizando-se funções da biblioteca ARToolKit.

4.2 Estrutura interna de uma aplicação ARToolKit

Uma aplicação de RA, utilizando-se da biblioteca ARToolKit segue os seguintes módulos ilustrados na Figura 4.5. Explanam-se estes módulos a seguir.

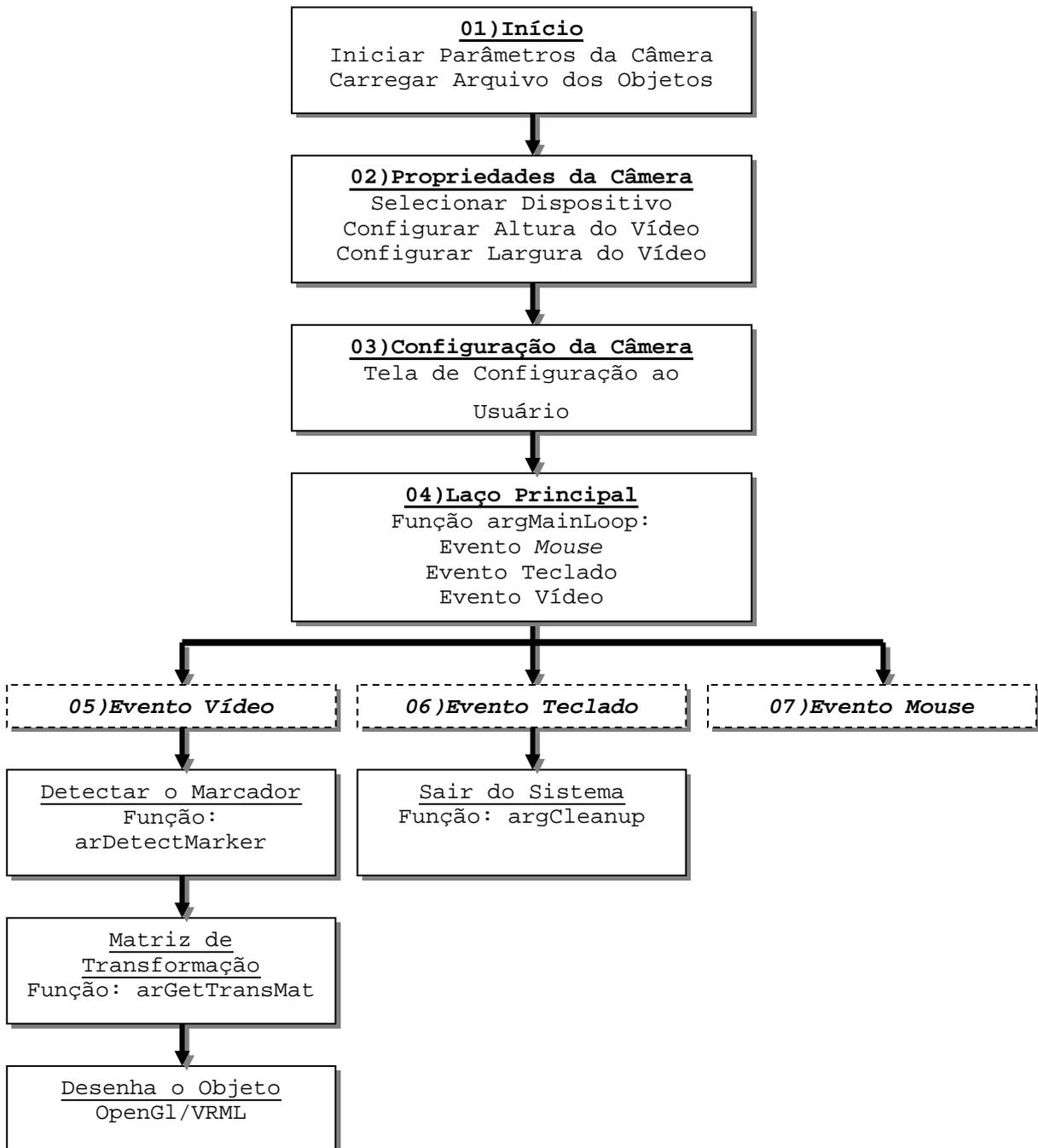


Figura 4.5: Módulos de uma aplicação ARToolKit

- 01) Módulo Início:

Neste módulo, são inicializados os parâmetros da câmera ou de qualquer outro dispositivo de captura de vídeo para funções RA, interpretam-se os arquivos de padrões dos marcadores, associa-se uma variável referente ao objeto virtual com o marcador, projeta-se o campo de visão do usuário. Caso o sistema de RA encontrar-se usando um monitor como dispositivo de saída, o mesmo cria uma janela para a disposição do vídeo, senão projetam-se as imagens diretamente no HMD.

- 02) Módulo de Propriedades da Câmera:

Neste módulo são selecionados os dispositivos de vídeo registrados no sistema operacional, além de configurar a altura e largura da projeção do vídeo.

- 03) Módulo de Configuração da Câmera:

Neste módulo é apresentado ao usuário um formulário de configuração da câmera ou do dispositivo de captura, nele são listadas as propriedades do *driver* deste dispositivo.

- 04) Módulo Laço Principal:

A função utilizada no laço principal é chamada para iniciar os eventos do teclado, *mouse* e vídeo.

- 05) Módulo Evento Vídeo:

Este módulo é dividido em três partes: a primeira função, denominada “arDetectMarker” é responsável por detectar os marcadores no quadro capturado pela câmera, a segunda função, “arGetTransMat” é destinada a calcular a matriz de transformação entre um marcador detectado e o dispositivo de captura, isto é, a posição e a orientação da câmera relativa ao marcador de rastreamento. E, por último, as funções da biblioteca gráfica OpenGL são usadas para configurar as coordenadas de projeção, além de desenhar os objetos virtuais. Os objetos virtuais também podem ser importados de um arquivo VRML.

- 06) Módulo Evento Teclado:

Este módulo é utilizado para eventos de controle do teclado, normalmente a função “argCleanup” é chamada para finalizar o processamento e desconectar o dispositivo de vídeo, liberando-o para outras aplicações.

- 07) Módulo Evento *Mouse*:

Este módulo é responsável por programar possíveis eventos nos quais o *mouse* seja necessário em uma aplicação de RA.

4.3 Sistemas de coordenadas

Nos gráficos 3D, as matrizes 4x4 são usadas para representar a posição, a rotação e a escala de um objeto. A matriz contém 16 valores que devem ser armazenados com exatidão a fim de assegurar que a transformação seja representada precisamente. Inicialmente, uma matriz que não executa nenhuma transformação é chamada de matriz identidade, demonstrada na Figura 4.6.

$$\begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Figura 4.6: Matriz identidade 4x4

Para a movimentação de um objeto, nos eixos X, Y ou Z a partir de seu ponto de origem, tem-se exposto na Figura 4.7 a matriz de translação, em que as variáveis **T_x**, **T_y** e **T_z** referem-se aos valores que fornecerão estas novas coordenadas de posição.

$$\begin{vmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Figura 4.7: Variáveis de translação

Alterando-se a escala de um objeto virtual ao longo de cada um de seus três eixos, as variáveis S_x , S_y e S_z referem-se aos valores que fornecerão as coordenadas de tamanho. A Figura 4.8 expõe essa matriz.

$$\begin{vmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Figura 4.8: Variáveis de escala

Ao executar-se uma rotação de um objeto virtual sobre um eixo central, as operações tornam-se presentemente complexas, necessitando de três matrizes para girar sobre X, Y, e Z. A Figura 4.9 (a)-(c) ilustra essas matrizes, nas quais atribui-se para a variável C o valor de $\cos(\theta)$ e para a variável S o valor de $\sin(\theta)$.

$$\begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & c & -s & 0 \\ 0 & s & c & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad \begin{vmatrix} c & 0 & s & 0 \\ 0 & 1 & 0 & 0 \\ -s & 0 & c & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad \begin{vmatrix} c & -s & 0 & 0 \\ s & c & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

a) Rotação em X b) Rotação em Y c) Rotação em Z

Figura 4.9: Matrizes de rotação

Verifica-se na Figura 4.10 que, θ é o ângulo de rotação sobre os eixos de coordenadas X, Y e Z, respectivamente. Esses ângulos são positivos quando medidos segundo o sentido inverso ao dos ponteiros de um relógio.

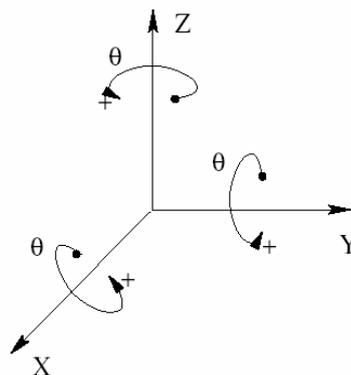


Figura 4.10: Sentido positivo das rotações sobre os eixos de coordenadas

4.4 Projeção em perspectiva

A exibição de um objeto 3D em uma tela de computador exige o mapeamento de um sistema de coordenadas 3D, ou seja, operações de projeção. Para que as projeções possam ser geradas em computador é necessário que sejam definidas as matrizes de transformações que, aplicadas ao conjunto de pontos de um objeto 3D, permitam a obtenção da figura projetada do objeto.

Inicialmente, os marcadores podem ser usados como interfaces tangíveis para lidar com artefatos virtuais ou como elementos de interface de interação do usuário.

As posições dos quadrados desenhados nos marcadores são utilizadas como base para a projeção inicial de um sistema de coordenadas. Através dessas coordenadas realiza-se, o processo de mudança do espaço real para o espaço da câmera, logo cria-se o efeito de perspectiva na formação da imagem (correspondente a uma representação 2D da cena) perdendo-se, portanto, uma das dimensões. A Figura 4.11 ilustra um marcador, na qual as variáveis X_m , Y_m e Z_m representam essas coordenadas. Independente de qual seja o valor da variável Z_m , o objeto virtual nunca será projetado atrás do marcador, pode, apenas, variar sua escala (diminuir ou aumentar seu tamanho).

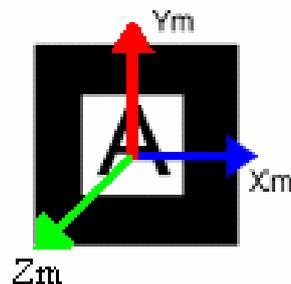


Figura 4.11: Coordenadas do marcador

Após o reconhecimento desses valores através da imagem capturada pelo vídeo, a matriz de transformação das coordenadas desse marcador para as coordenadas da câmera são representadas pela matriz na Figura 4.12. Na qual as variáveis X_c , Y_c , e Z_c representam as coordenadas da câmera. As variáveis R e T , como exposto anteriormente (item 4.3) referem-

se aos valores de rotação, translação ou escala. O resultado final deste relacionamento é a matriz definida como T_{CM} .

$$\begin{bmatrix} X_C \\ Y_C \\ Z_C \\ 1 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_1 \\ R_{21} & R_{22} & R_{23} & T_2 \\ R_{31} & R_{32} & R_{33} & T_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_M \\ Y_M \\ Z_M \\ 1 \end{bmatrix} = T_{CM} \begin{bmatrix} X_M \\ Y_M \\ Z_M \\ 1 \end{bmatrix}$$

Figura 4.12: Matrizes referentes às coordenadas da câmera, marcador e matriz final

Neste momento, encontram-se pontos decisivos nos reconhecimentos de marcadores no campo de visão do usuário. Quanto mais marcadores forem encontrados pelo sistema de RA, maior será o processamento envolvido em todo o procedimento, bem como, se um marcador apresentar uma ampla dimensão, os valores das coordenadas apresentarão maior precisão, porém todo o processo de reconhecimento da imagem será lento. Por outro lado, um marcador pequeno oferece uma baixa exatidão na identificação das coordenadas, porém um rápido processamento da imagem (KATO & BILLINGHURST, 1999).

Em um plano 2D, exclui-se a profundidade (Eixo Z) e apenas os valores das coordenadas de projeção (Eixo X e Y) representadas pelas variáveis x_c e y_c são atribuídos ao monitor do vídeo. Todo o processo de transformação dos valores das coordenadas dos marcadores até as coordenadas de projeção do vídeo são ilustrados pela Figura 4.13.

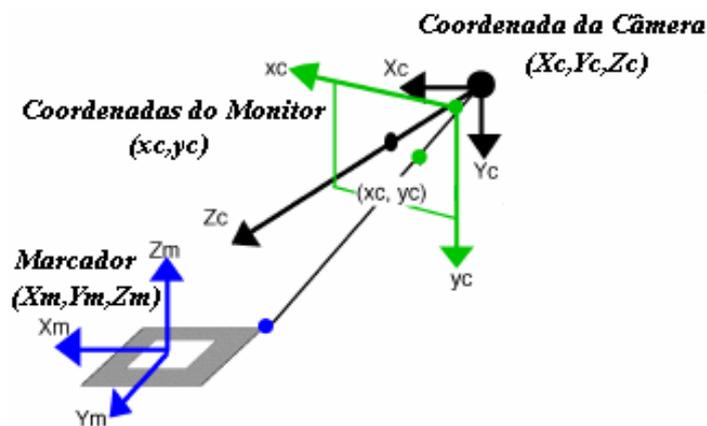


Figura 4.13: Sistemas de coordenadas do ARToolKit (KATO & BILLINGHURST, 1999)

4.5 Limitações de sistemas de RA utilizando o ARToolKit

Os resultados de rastreamento são afetados por condições de iluminação. A luminosidade deve ser constante e suave em toda área onde o rastreamento será realizado. Se não existir iluminação suficiente ou se a luz for muito brilhante ou incidir diretamente sobre o marcador, ele não será reconhecido pelo ARToolKit. A saturação luminosa pode criar reflexões e pontos de brilho nas marcas do papel e podem tornar mais difícil a tarefa de encontrar as marcas quadradas. As soluções convencionais para estes problemas são: realizar o rastreamento num ambiente com iluminação adequada e utilizar materiais não-reflexivos (opacos) para a confecção dos marcadores.

Uma outra limitação encontrada é se o marcador não estiver completamente visível para a câmera (LEDERMANN et al., 2002). O marcador deve estar totalmente dentro do ângulo de visão da câmera e não deve sofrer nenhuma obstrução por nenhum tipo de obstáculo ou por outro marcador qualquer, a Figura 4.14 (a)-(b) ilustra essas situações. Para solucionar este problema são usados geralmente mais de um marcador para a projeção de um único objeto virtual, ou a adição de mais uma câmera.



a) Marcador fora do campo de visão



b) Marcadores encobertos por outro marcador

Figura 4.14: Situações comuns de oclusão (LEDERMANN et al., 2002)

O rastreamento dos marcadores feito pelo ARToolKit é perdido sempre que o mesmo não esteja completamente ou claramente visível pela câmera. Isso ocorre devido ao problema de oclusão (item 2.3.7) ocasionado pela limitação do ângulo da câmera ou quando qualquer

obstáculo que cobre parte do marcador, tornando-se uma barreira, restringe as áreas eficazmente rastreadas.

Outra limitação é causada por marcadores muito pequenos ou distantes. Se a câmera não consegue focalizar totalmente ou claramente o marcador, essa imagem não será reconhecida ou os resultados obtidos serão imprecisos. Uma solução para esse tipo de problema é o uso de marcadores com tamanhos apropriados, segundo os autores, as medidas sugeridas relativas às bordas pretas e as distâncias do campo de visão são demonstradas na Tabela II.

Tabela II: Medidas sugeridas

Tamanho do Marcador (em centímetros)	Espaço de Interação Útil (em centímetros)
6,985275	40,6416
8,89035	63,5025
10,795425	86,3634
18,720537	127,005

4.6 Compatibilidades e versões

O rastreamento rápido e preciso oferecido pelo ARToolKit propicia um desenvolvimento dinâmico de diversas aplicações interessantes de RA. O código fonte de suas bibliotecas é completamente disponibilizado, tornando possível portá-lo para diversas plataformas ou adaptá-lo para resolver as especificidades das aplicações.

Atualmente, o ARToolKit está disponível para plataformas Irix, Linux e Windows 95/98/NT/2000/XP. Há versões separadas para cada uma dessas plataformas. A funcionalidade de cada versão da biblioteca é a mesma, mas o desempenho pode variar conforme as diferentes configurações de *hardware*. Ainda há implementações do ARToolKit para Matlab e para Java (esta última conhecida como JARToolKit) e também iniciativas de versões reduzidas para *palms*.

A Tabela III demonstra algumas versões oficiais do ARToolKit, nela são destacados a data de lançamento, o número da versão, o sistema operacional compatível e o suporte ao tipo de objeto gráfico que pode ser utilizado em um sistema de RA.

Tabela III: Características das versões do ARToolKit

Data	Versão	Sistema Operacional	Objeto Gráfico
1999	1.0	Windows – SGI Irix	OpenGL
1999	2.11	Windows – SGI Irix – Linux	OpenGL
2000	2.33	SGI Irix – Linux	OpenGL
2001	2.40	SGI Irix – Linux	OpenGL
2001	2.43	Windows – SGI Irix – Linux	OpenGL - VRML
2002	2.52	Windows	OpenGL - VRML
2002	2.60	Linux	OpenGL - VRML
2002	2.61	MacOs – SGI Irix – Linux	OpenGL - VRML
2004	2.68	Linux – Windows – MacOs – SGI Irix	OpenGL - VRML
2004	2.69	Linux – Windows – MacOs – SGI Irix	OpenGL - VRML
2004	2.70	Linux – Windows – MacOs – SGI Irix	OpenGL - VRML
2005	2.71	Linux – Windows – MacOs – SGI Irix	OpenGL - VRML

No próximo capítulo, serão discutidas as interações em sistemas de RA, enfatizando características como: as formas de interação com o usuário, as facilidades de utilização. Serão expostas aplicações nas quais a biblioteca ARToolKit foi utilizada, tecnologia abordada no presente capítulo.

5. INTERAÇÃO EM SISTEMAS DE RA

Os sistemas de RA têm o propósito de interação em ambientes 3D visando estimular o maior número de sentidos humanos para conectar o usuário de forma mais próxima possível a realidade. Porém, para fazer essa conexão, é necessário que o usuário visualize, compreenda e execute as tarefas necessárias no AV. Um AV é um cenário dinâmico em 3D, modelado computacionalmente através de técnicas de computação gráfica e usado para representar a parte visual de um sistema de RV. O AV nada mais é do que um cenário onde os usuários de um sistema de RV podem interagir (PINHO, 2000).

Um ambiente 3D é um cenário armazenado em computador e exibido por meio de técnicas de computação gráfica. Esses ambientes podem ser divididos em duas categorias básicas: ambientes 3D imersivos e não-imersivos.

O processo de interação em um AV, seja ele imersivo ou não, é considerado como um sistema de atuação no qual cada ação do usuário deve ser respondida com uma ação do controlador do sistema do AV.

Na interação imersiva, considera-se que o usuário esteja utilizando dispositivos de RV capazes de impedir que veja o mundo real que o cerca, passando a enxergar apenas o ambiente gerado pelo sistema gráfico. Na interação não-imersiva e semi-imersiva, toma-se o usuário como alguém que observa um cenário 3D através da tela de um computador (PINHO, 2000).

Muitos critérios de projeto de interfaces 2D podem ser facilmente aplicados às interfaces 3D. Considerando a metáfora e a manipulação direta como os dois princípios mais importantes do projeto de interfaces, é fácil entender porque as interfaces gráficas são comercializadas de maneira cada vez mais intensa hoje em dia, e por que deve-se aplicar estes conceitos na concepção de gráficos interativos 3D (TISSIANI et al., 2001).

Porém, além de vários outros critérios de projeto gráfico de interfaces, segundo TISSIANI et al. (2001), há algumas considerações que devem ser entendidas para possibilitar o desenvolvimento das tarefas a serem executadas pelos usuários e para suprir as necessidades de interação. Os principais critérios são descritos na Tabela IV.

Tabela IV: Critérios para um projeto de uma interface gráfica de interação

Critérios	Necessidades
Simplicidade	Implica no não comprometimento da usabilidade em prol da função.
Suporte	Manter o usuário com o controle das ações.
Familiaridade	Concepção baseada no conhecimento prévio dos usuários.
Clareza	Construção de objetos e de seus controles de modo visível e intuitivo.
Premeditação	Estabelecer ações de acordo com as expectativas dos usuários.
Satisfação	Criação de um sentimento confortável perante a interação.
Disponibilidade	Colocar todos os objetos disponíveis durante todo o tempo, o que implica em disponibilizar todos os níveis dos menus virtuais presentes quando acessados pelo usuário.
Afinidade	Transformar os objetos em figuras visuais, sempre que possível, que referenciem aqueles da realidade.

A informação exibida como um sistema visual aumenta a produtividade por causa do estilo de manipulação direta com os elementos da interface. Em um AV, elementos visuais como ícones, símbolos, cores, entre outros, compõem a interface como um todo (TISSIANI et al., 2001).

5.1 Tipos de interação

Não há provavelmente técnicas de interações que se apliquem para todas tarefas possíveis. As diferentes características de cada tarefa impõem diferentes necessidades quanto

às técnicas de manipulação. Em um estudo ideal, podem-se avaliar técnicas de interação para todos os trabalhos possíveis (POUPYREV et al., 1997). Alguns critérios de desempenho para a avaliação de uma técnica de interação são descritos na Tabela V.

Tabela V: Critérios de desempenho

Critérios	Definição
Tempo de Atuação	O tempo necessário para realizar com sucesso as tarefas.
Exatidão	A avaliação da proximidade do objeto para posição ou orientação desejada.
Taxa de Erro	O número de tentativas falhas quando se realiza uma tarefa.
Facilidade de Uso	O estímulo racional direciona o usuário enquanto estiver usando técnicas de interação.
Facilidade de Aprendizagem	É a habilidade do usuário melhorar seu desempenho com a experiência adquirida.
Sensação de Imersão	O usuário conscientemente sente-se imerso.

A interação com objetos em ambientes 3D é realizada com objetivo de efetuar operações de manipulação, seleção e navegação.

Como manipulação entende-se a tarefa de alterar algum parâmetro de um objeto 3D que acarrete mudança de suas propriedades geométricas (tamanho, posição ou orientação). Como seleção entende-se o processo de definir, dentre os objetos de um ambiente 3D, sobre qual deles se deseja operar. A navegação, por sua vez, é o processo de mudar a posição do observador do ambiente 3D (PINHO, 2000).

5.2 Técnicas de manipulação

Uma das mais importantes formas de interação é a especificação da posição ou orientação de um objeto no mundo virtual. Essa interação pode ser realística, com o usuário segurando e movendo um objeto virtual, como se o fizesse em um mundo real, ou não, com o usuário movendo-o de uma forma sem paralelo no mundo real (PINHO, 2000).

De acordo com Poupyrev et al. (1997), identificar as tarefas básicas, entretanto, não é suficiente. Para cada tarefa básica, existem vários parâmetros que afetam o desempenho do

usuário e que precisam ser considerados. O desempenho de um usuário para uma tarefa de seleção de objetos, por exemplo, depende da distância e do tamanho do objeto a ser selecionado. Um objeto localizado próximo ao usuário pode ser fácil de selecionar, diferente de um objeto localizado fora de seu alcance. Geralmente, os parâmetros das tarefas de manipulação são todos os fatores influenciados pelo desempenho do usuário enquanto realizador de alguma atividade. Segundo Poupyrev et al. (1997) os parâmetros podem ser classificados em função da dependência com:

- **Usuário:** experiência, capacidade cognitiva, percepção e habilidades motoras, diferenças físicas e outras;
- **Dispositivo de entrada e saída:** atributos dos dispositivos, como grau de liberdade, resolução, campo de visão e dentre outros;
- **Técnicas de interação:** metáforas de técnicas, suas sugestões e implementações;
- **Aplicação:** configuração do AV, tamanho, forma e emprego dos objetos, cores, iluminações e outras; e
- **Contexto:** precisão requerida, condições finais e iniciais das tarefas, reação das tarefas e outras.

Em geral, dois parâmetros precisam ser especificados para a manipulação de um objeto: a mudança na posição e a mudança na orientação. Para alterar esses parâmetros há algumas possibilidades, entre elas especificar um controle virtual girando e posicionando os objetos no espaço ou através de controles físicos.

5.3 Técnicas de seleção

A interação com objetos virtuais requer formas de seleção de objetos, isto é, meios de indicar o alvo de interação desejado. Para todos os casos em que exista a seleção de objetos, é requerido algum mecanismo para identificação do objeto a ser selecionado e algum sinal ou

comando para indicar o ato de seleção. Este sinal pode ser tipicamente representado na forma de um botão pressionado, um gesto, ou algum tipo de comando de voz (PINHO, 2000).

Segundo Poupyrev et al. (1997), encontram-se os seguintes parâmetros para tarefas de seleção: números de objetos a serem selecionados; distância do objeto alvo; tamanho do objeto alvo; direção do objeto alvo e a oclusão do objeto alvo.

Há duas técnicas primárias para as tarefas de seleção: local e à distância. Na seleção local, existe o contato do usuário com o objeto desejado. Nesse modo, os objetos são escolhidos pela movimentação do cursor até o objeto. A técnica de seleção à distância, por sua vez, é adotada quando não há o contato direto do usuário com o alvo da seleção. Nesses casos, a seleção pode ser realizada usando-se dispositivos como apontadores a *laser* (Figura 5.1), que lançam raios que interceptam o objeto no mundo virtual (PINHO, 2000).

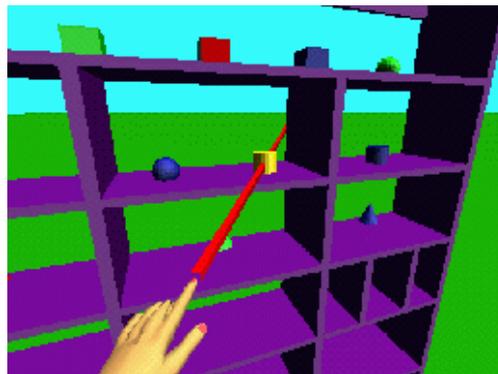


Figura 5.1: Usuário indicando objeto virtual distante

5.3.1 Interação por meio de menus

Certamente a manipulação direta dos objetos é a forma dominante e mais natural de interação em AV. Entretanto, nessas aplicações ainda há espaço para o uso de menus. Tal uso aplica-se essencialmente onde a interação 3D não é natural. Por exemplo: a mudança na cor de um objeto, a definição do modo de trabalho ou de navegação podem ser muito mais facilmente especificados em um menu do que na interação direta com os objetos.

Há diversas formas de incorporar menus em um AV. A diferença primária entre os diversos projetos está na dimensionalidade do mecanismo de seleção (número de dimensões a

ser especificada pelo usuário para a seleção de diferentes opções) e no local de apresentação desses menus ao usuário.

Menus de uma dimensão são ideais para a seleção de um simples item com um conjunto de opções. Menus de duas ou três dimensões adicionam poder para a aplicação e, ao mesmo tempo, problemas na tarefa de interação com o menu. Componentes 3D de interface levam a uma maior interação com o ambiente e ajudam a aumentar o controle sobre os objetos. Por outro lado, a adição de uma dimensão torna mais complicado ao usuário interagir com os menus, pois terá de posicionar corretamente sua mão acertando as coordenadas do alvo em mais de uma dimensão (PINHO, 2000). A Figura 5.2 ilustra um dispositivo de projeção de um menu 2D sobre uma mesa.

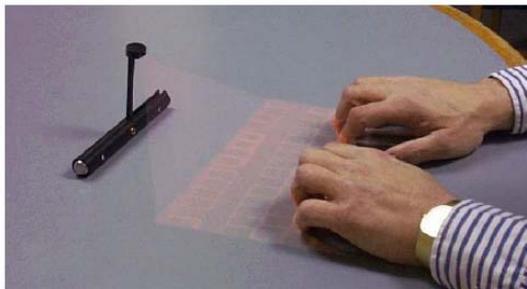


Figura 5.2: Projetor de um controle de interação

O posicionamento do menu, por sua vez, irá definir como apresentar o menu ao usuário. Uma opção é ter o menu flutuando no espaço, como se fosse outro objeto virtual. Quando o usuário desejar movimentar o menu, deve segurá-lo e movê-lo para um outro local no espaço. Com um menu no espaço, pode ocorrer que este bloqueie a visão do usuário no mundo virtual. Uma solução é ter algum meio adicional de fazê-lo aparecer ou desaparecer rapidamente.

Os princípios de projeto de um menu virtual estão diretamente relacionados aos princípios de projeto de uma interface 2D. Menus virtuais, entretanto, tem problemas adicionais resultado do trabalho em um ambiente 3D e das limitações (como ruídos e distorções) nos equipamentos existentes atualmente para rastreamento.

5.4 Técnicas de navegação

O mapeamento do movimento físico para o virtual é uma das maneiras mais intuitivas de se realizar uma animação em um AV. Feito desta forma, o movimento não requer nenhuma ação especial por parte do usuário (BOWMAN, 1999).

A desvantagem de usar um movimento físico para mover-se no AV, é que o alcance do movimento do usuário depende diretamente da tecnologia de rastreamento utilizada. Quando se opta por rastrear o movimento do corpo do usuário com rastreadores, fica-se restrito, na maioria dos casos, a uma área de rastreamento, que muitas vezes, não proporciona uma adequada mobilidade do usuário no ambiente.

Os dispositivos de entrada comuns (*mouse*, teclado, etc.) operam apenas em um plano. Para a navegação em um AV 3D, é necessário um artifício que mapeie de 2D para 3D o que, em geral, não é fácil para qualquer usuário assimilar. Assim, dispositivos de entrada que oferecem três ou mais graus de liberdade são mais adequados para a navegação em um espaço 3D. Além disso, a escolha do dispositivo deve considerar vantagens e limitações do mesmo bem como conceitos de ergonomia.

Ao invés de utilizar-se dispositivos físicos, uma alternativa é a implementação de dispositivos virtuais para controlar a navegação em um AV. Tal implementação tem como vantagem flexibilidade de poder modelar qualquer controle de navegação.

Bowman (1999) identifica três tarefas básicas que devem ser analisadas para qualquer técnica de navegação: seleção da direção do objeto alvo, que define como o usuário indica a direção ou o ponto final do movimento, seleção da velocidade e aceleração sendo as técnicas permitem que o usuário varie a velocidade e a aceleração do movimento e condição de entrada que significa como o usuário inicia, continua e finaliza o movimento.

5.5 Interação em sistemas imersivos

Para que os AV do tipo imersivo possam ser utilizados em sua plenitude, os componentes mínimos são: *hardware* de aceleração gráfica; HMD de imersão visual e auditiva; luvas com sensores de posicionamento; computador que gerencie e execute a aplicação (ambiente) e o *software* que gere o AV.

A interconexão entre esses componentes permite ao usuário usufruir de uma sensação muito real, possibilitando-o imaginar que está fisicamente presente em outro ambiente (LUZ, 1997).

Na análise de qualquer das tarefas interativas, pode-se definir três categorias, de acordo com tipo de controle exercido pelo usuário. Essas categorias são observadas a seguir de acordo com Mine (1995).

5.5.1 Interação direta

A interação direta com objetos em um AV é condição essencial para a implementação de aplicações que traduzem ações do mundo real para objetos virtuais. Essa categoria inclui as técnicas interativas que utilizam o corpo do usuário (mãos, braços, cabeça, entre outros), atuando diretamente sobre o objeto por meio de um “toque virtual” sobre este. Para tanto, se faz necessário que o sistema de RV possua funções de suporte ao rastreamento das mãos e da direção do olhar, reconhecimento de gestos e detecção de mirar um objeto.

O sucesso das técnicas de interação direta depende da capacidade do sistema em realizar um mapeamento natural e intuitivo entre a ação do usuário e a ação resultante no mundo virtual.

Para Poupyrev et al. (1997), pode-se entender que manipulações de tarefas básicas em RV são as mesmas tarefas que são executadas no mundo real, quando se realizam alguns movimentos. Movimentos são alcançados (ou obtidos) através da mudança de posições e

orientações a todo tempo, no momento em que é movida alguma coisa para outro local. Esses movimentos são as combinações de alcançar, agarrar, mover e orientar os objetos.

5.5.2 Interação com controles físicos

Essa categoria inclui o uso de botões, luvas, *joysticks*, pedais, entre outros. Usar controles físicos para interagir com um mundo virtual (como um volante, em um simulador de carro) pode aumentar muito a sensação de presença do usuário no mundo virtual, pois permite ao usuário algum tipo de sensação tátil não disponível na interação direta. Dispositivos físicos também são úteis para o controle preciso da tarefa de interação. Esses dispositivos, no entanto, nem sempre oferecem um mapeamento natural que facilite a tarefa de interação no mundo virtual.

5.5.3 Interação com controles virtuais

A idéia, nesse caso, é representar visualmente um dispositivo físico. Qualquer coisa que se imagine pode ser implementada como um controle virtual. Essa grande flexibilidade é a maior vantagem dos controles virtuais. Entretanto, as desvantagens incluem a falta de um retorno sensorial e a dificuldade de interação com o objeto virtual (PINHO, 2000).

5.6 Comparação do tipo de interação em sistemas de RV

Devido ao alto nível de imersão exigido pelos AV, estes devem exibir as imagens eficientemente, procurando gerá-las em tempo real para que seja possível a interação com o usuário que deve perceber as modificações no ambiente no momento em que são realizadas as ações.

Na Tabela VI, são descritas técnicas de interação, sendo que os componentes básicos são: interfaces lógicas, que especificam como os parâmetros do ambiente e de seus objetos

podem ser alterados, e interfaces físicas, que consistem em um ou mais equipamentos visuais, auditivos ou táteis. Para um AV, é necessária existência de ferramentas que permitam ao usuário realizar a tarefa atribuída.

Tabela VI: Interação em sistemas de RV

Técnica	Descrição	Vantagens	Desvantagens
Direção pela mão	Usuário aponta na direção a ser seguida.	Intuitivo e de fácil operação.	Enquanto o usuário se movimenta, não pode manipular objetos; gera fadiga muscular.
Direção ao olhar	Usuário movimenta-se à medida que move a cabeça.	Intuitivo e de fácil operação.	O usuário não consegue olhar à sua volta enquanto se movimenta.
Controles Físicos	<i>Joystick, trackball, mouse, botões.</i>	Baratos e fáceis de utilizar.	Pouco intuitivo, de difícil operação.
Controle Virtual	Objetos virtuais utilizados para controlar o movimento.	Flexível, pode ser desenvolvido conforme a necessidade.	Dificuldade de operação devido à falta de retorno tátil.
Dirigido por Objetos	Controle é feito por objetos e não pelo usuário (carro, relógio, entre outros).	Não necessita da interação do usuário.	Não possibilita a interação do usuário.

5.7 Exemplos de sistemas de RA e comparação do tipo de interação com o usuário

Neste item, serão exemplificadas algumas aplicações de RA, consideradas importantes, que foram desenvolvidas utilizando-se a biblioteca ARToolKit. Serão apresentadas, também, algumas considerações dos projetos e arquitetura dos sistemas, exibindo a funcionalidade com que podem ser aplicados.

5.7.1 ARWin - Gerenciador de Ambientes Operacionais

Nos últimos vinte anos, o gerenciamento das interfaces operacionais do computador foi dominado pelo paradigma 2D, WIMP (*Windows-Icon-Menu-Pointer*). Com os avanços computacionais e as melhorias consideráveis no processamento gráfico 3D, atualmente é possível desenvolver-se ambientes 3D aumentados com facilidades sem precedentes. Neste

contexto, o projeto ARWin foi desenvolvido na Universidade da Califórnia, tratando-se de um ambiente operacional 3D aumentado, que realça especificamente visualizações de dados, além do mesmo utilizar-se de uma variedade de técnicas híbridas de manipulação de dados e de interação.

Esse novo ambiente permite ao usuário trabalhar em uma forma familiar com aplicações tradicionais e, ao mesmo tempo, introduzir as aplicações que são desenvolvidas especificamente para o espaço de trabalho 3D. O ambiente aumentado pode imitar ou estender objetos tradicionais, tais como um relógio de parede ou um calendário, ou pode visualizar uma informação de forma espacial, como demonstra Figura 5.3.



Figura 5.3: Ambiente ARWin (HÖLLERER et al., 2003)

Para obter um maior aproveitamento do espaço de trabalho 3D, o ARWin estende a interação tradicional da aplicação fornecendo um evento de proximidade: quando duas (ou mais) aplicações são colocadas próximas, cada aplicação recebe um evento que notifica os membros de sua vizinhança. O que as aplicações fazem com esse evento é definido pelo desenvolvedor. A Figura 5.4 ilustra a forma como o ambiente gerado administra uma informação de proximidade (HÖLLERER et al., 2003).

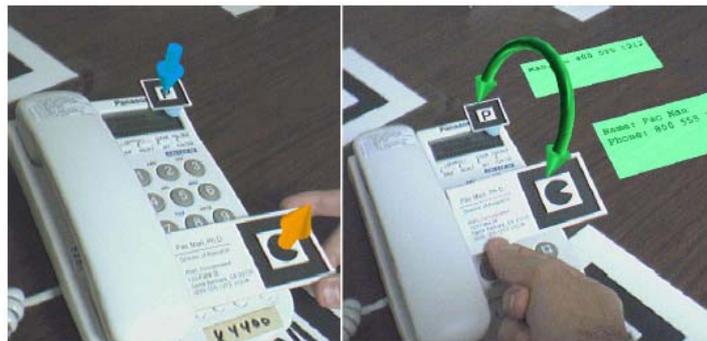


Figura 5.4: Interação por proximidade no ambiente ARWin (HÖLLERER et al., 2003)

5.7.2 Química Aumentada: um ambiente de trabalho interativo e educacional

O seguinte projeto demonstra como a RA pode interagir nas diversas áreas da Ciência, especificamente na área de Química. Esse trabalho apresenta-se como um conjunto de ferramentas interativas, no qual o usuário pode escolher elementos de um pequeno livro e compor modelos moleculares 3D. Tais ferramentas podem ser usadas por vários usuários simultaneamente.

Baseado na biblioteca do ARToolKit, um dos objetos destas ferramentas é um cubo com marcadores diferentes impressos em todos os lados, sua função é conectar e integrar os modelos moleculares 3D, onde as composições dessas moléculas seguem regras de formulação e integração, seguindo a natureza das estruturas moleculares básicas.

O projeto de Química Aumentada é composto por uma área de trabalho constituída de uma mesa e de uma projeção de vídeo em uma tela de vidro acrílico (Figura 5.5), caracterizado como um dispositivo de projeção espacial (item 2.2). Abaixo dessa tela situa-se uma câmera que captura a imagem do espaço real projetando-a no vídeo.



Figura 5.5: Ambiente do sistema de Química Aumentada (FJELD & VOEGTLI, 2002)

Além do cubo rotacionador, o sistema é composto por outras ferramentas, como uma pequena pá e um pequeno livro com os marcadores. Usando essa pá, o usuário pode retirar um elemento molecular do livro e com a rotação do cubo operada pela outra mão, pode determinar como esse elemento irá se conectar com a molécula (FJELD & VOEGTLI, 2002).

5.7.3 FingARTips: manipulação direta baseada em gestos em RA

O presente trabalho apresenta uma técnica de interação em ambientes de RA denominada FingARTips, na qual foi utilizada a biblioteca ARToolKit para seu desenvolvimento. Marcadores são acoplados às pontas dos dedos com a finalidade de seguir os gestos do usuário e estabelecer uma intenção: manipular objetos virtuais em um sistema de planejamento urbano. Este cenário virtual é ilustrado na Figura 5.6.

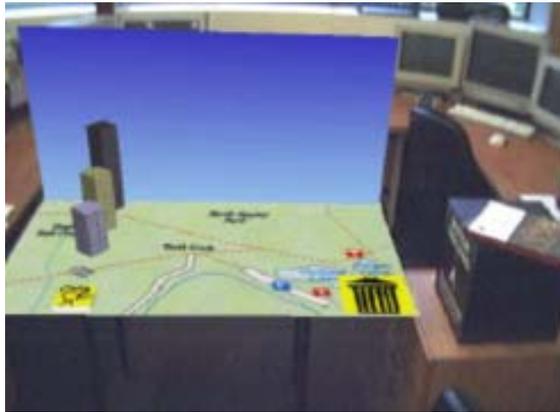


Figura 5.6: Cenário urbano virtual (BUCHMANN et al., 2004)

O reconhecimento da intenção dos gestos é baseado em uma de três medidas: a posição dos dedos relativa a um objeto virtual, a posição dos dedos relativa ao cenário virtual, ou a posição dos dedos relativa às distâncias dos mesmos. Gestos de agarrar são usados para manipular os edifícios. À partir desses gestos, edifícios podem ser criados, removidos, mudados de posição e redimensionados. Para criar um novo edifício, o usuário simplesmente agarra um modelo de um menu. Isso criará uma cópia, que faz com que o edifício virtual fique preso à mão do usuário (BUCHMANN et al., 2004), como ilustrado na Figura 5.7.

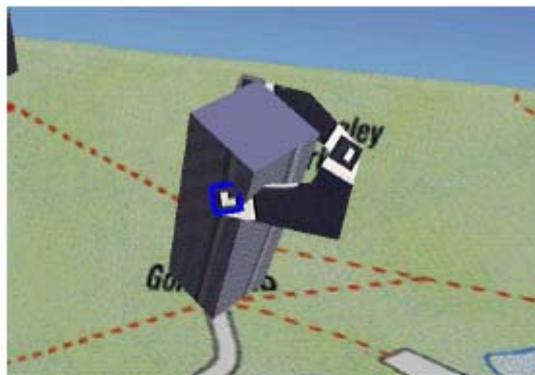


Figura 5.7: Manipulação de um edifício virtual (BUCHMANN et al., 2004)

5.8 Considerações sobre as tecnologias abordadas

A Tabela VII descreve os sistemas desenvolvidos, ilustrando uma breve descrição de seu funcionamento e as técnicas utilizadas nas tarefas de interação pelo usuário no AV.

Tabela VII: Resumo das tecnologias abordadas

Sistema	Descrição	Técnica de Interação
ARWin	Ambiente operacional 3D aumentado	O usuário visualiza a informação espacial e realiza a interação pela proximidade entre marcadores
Química Aumentada	Manipulação de objetos virtuais (modelos moleculares) visando estudos científicos.	Ferramentas não-convencionais baseadas em marcadores: uma ilustrativa, uma para mover o objeto e outra para manusear.
FingARTips	Sistema de modelagem urbana	Manipulação direta do objeto virtual baseada em marcadores acoplados aos dedos, simulando o gesto de pegar.

A partir deste estudo sobre as técnicas de interação, foi possível delinear um panorama sobre as técnicas existentes e, também, escolher as mais apropriadas para os experimentos deste trabalho. Na implementação de uma técnica, o que se busca, é que seja intuitiva para o usuário, posto que o interesse deste deve estar voltado para o ambiente e não para a técnica.

Os sistemas apresentados propiciam ao usuário interações de maneira intuitiva com o AV, usando as mãos com marcadores ou com simples objetos de controle, rastreados por uma câmera ou por um HMD, dado que interfaces mais intuitivas e interativas requerem cada vez menos pessoas altamente especializadas para sua manipulação, não restringindo, assim, o seu uso.

O próximo capítulo ilustra as características e funções do suporte desenvolvido, que oferece serviços para a implementação de interfaces de interação em um AV.

6. BIBLIOTECA PARA INTERAÇÃO EM RA

Este capítulo apresenta o suporte para interação denominado ARISupport (*Augmented Reality Interaction Support*), que provê aos desenvolvedores funções para criação de interfaces sensíveis para aplicações de RA baseadas no ARToolKit. Tendo este trabalho como foco o desenvolvimento de uma ferramenta para a avaliação e experimentação de manipulação em interfaces imersivas.

6.1 Objetivos principais

Este trabalho focou-se no problema da utilização de metáforas de interação para aplicações de RA. Nesse contexto, são objetivos deste trabalho:

- Estudar os aspectos gerais sobre RA, como: características dessa tecnologia, equipamentos necessários e desafios;
- Realizar um levantamento das aplicações de RA mais relevantes, segundo os aspectos de interação;
- Estudar recursos e funcionalidades da biblioteca ARToolKit, bem como bibliotecas relacionadas, como OpenGL e GLUI;
- Desenvolver um suporte de interação para RA baseado nos principais modelos de interação de aplicações levantadas; e
- Implementar protótipos para validação do suporte.

6.2 Estruturação do suporte

O suporte foi estruturado como uma biblioteca dividida em três módulos integrados, a saber (Figura. 6.1):

- **Módulo de Ambiente:** responsável pela importação de figuras como imagens de fundo (*background*), texturas de superfície e funções como som e vídeo.
- **Módulo de Formas Geométricas:** fornece objetos gráficos complexos a serem adicionados às aplicações de RA, bem como permite a importação de objetos 3D e a adição de textos 2D.
- **Módulo de Interação:** oferece funções matemáticas úteis para a alteração do posicionamento e orientação dos objetos virtuais.

A implementação de um suporte, como uma biblioteca, aumenta a produtividade e evita esforços desnecessários de desenvolvimento. Disponibiliza um conjunto de elementos já elaborados que podem, de maneira fácil e rápida, serem utilizados para construir um AV.

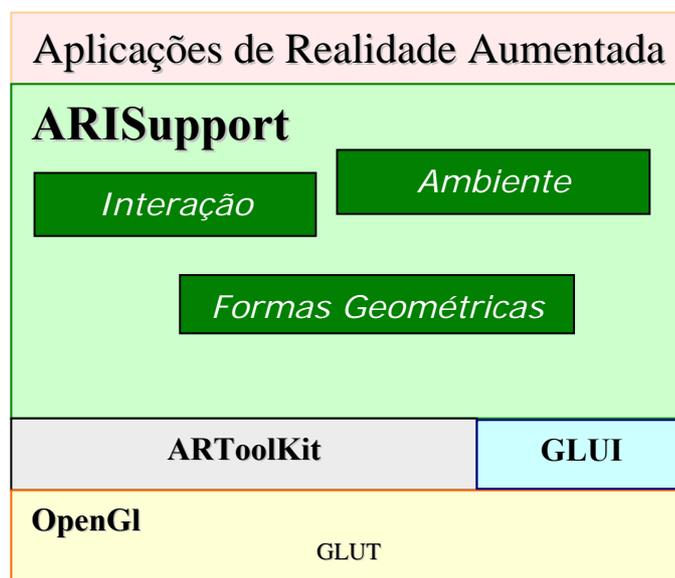


Figura 6.1: Diagrama da arquitetura do ARISupport

Um objeto de interação é definido como um objeto de *software* cujo processamento gera uma imagem que é apresentada ao usuário e com a qual ele pode interagir. Para o projeto visual de um sistema, o projetista pode se valer de metáforas do mundo real como forma de

tornar a estrutura do sistema de informação reconhecível intuitivamente (TISSIANI et al., 2004).

Os componentes providos por essa biblioteca e suas funcionalidades são explanados a seguir:

a) Módulo de Ambiente

São oferecidos neste módulo:

- Funções de carregamento de imagens: permite o carregamento de imagens no formato BMP (*Bit Map*) e JPEG (*Joint Photographic Experts Group*) como texturas de superfícies e a adição de texturas aos objetos virtuais em formatos diferentes. A Figura 6.2 ilustra um exemplo dos efeitos da utilização desta função.



Figura 6.2: Adição de imagens virtuais ao ambiente real

- Função de aparência: oferece o controle de colorização do objeto virtual (procurando fornecer aspectos mais naturais ao objeto virtual projetado no ambiente real) em diferentes níveis de cores como demonstra a Figura 6.3 (a)-(c).



a) Coloração em Vermelho



b) Coloração em Verde



c) Coloração em Azul

Figura 6.3: Tonalidade de cores em objetos virtuais

- Função de transparência: introduz efeitos de transparência a fim de evitar que um objeto virtual possa obstruir a visibilidade do usuário ao acesso a um objeto alvo. Um exemplo do resultado da utilização dessa função é demonstrado na Figura 6.4, na qual se consegue ver o marcador sobreposto por um cubo virtual.

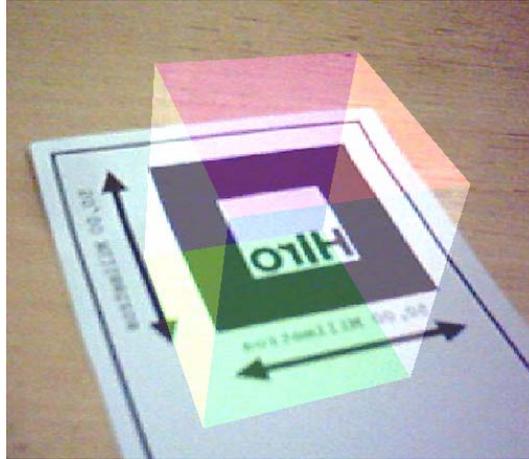


Figura 6.4: Objeto virtual transparente

- Função de som: permite a adição de sons em formato WAV (*Windows Audio Volume*), armazenados previamente em arquivos, aos objetos virtuais.
- Função de carregamento de vídeo: funções de importação de vídeos e manipulação dos quadros em formato AVI (*Audio Video Interleave*). A Figura 6.5 ilustra os efeitos do uso dessa função.

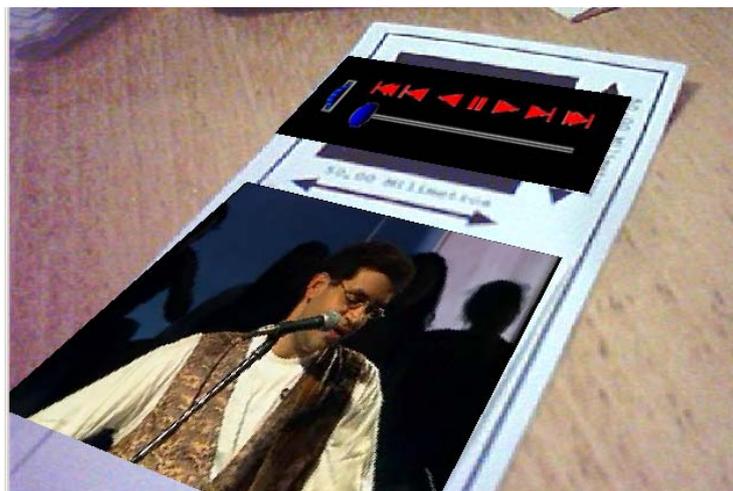


Figura 6.5: Vídeo adicionado ao ambiente

b) Módulo de Formas Geométricas

Este módulo oferece os seguintes recursos gráficos:

- Funções para a criação de formas geométricas básicas 2D e 3D: cria objetos gráficos que possuem características direcionadas às aplicações de RA com parâmetros que agilizam a construção do mesmo, como, por exemplo: quadrados, cilindros, esferas (malha), entre outros (Figura 6.6).

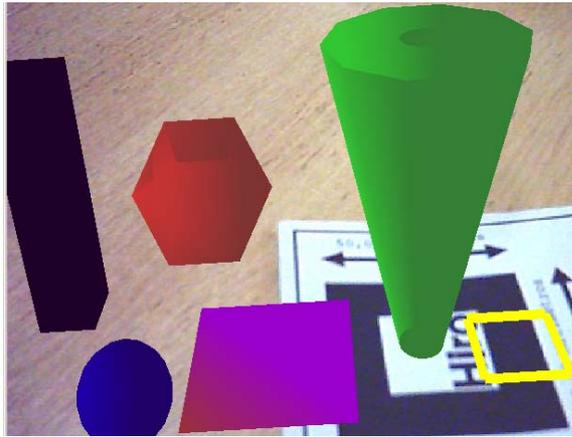


Figura 6.6: Diversas formas geométricas

- Função para criação de desenho de textos 2D e 3D: projeção de informações no campo de visão do usuário através do emprego de textos 2D e 3D.
- Função para criação de teclas numéricas 3D: oferecer objetos numéricos 3D para a construção de teclados virtuais em um ambiente virtual (Figura 6.7).

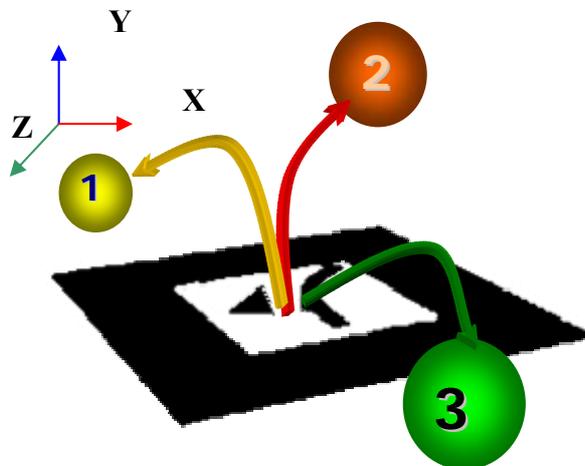


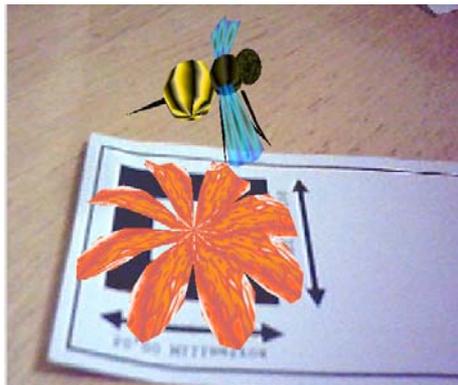
Figura 6.7: Teclas numéricas adicionadas a um marcador

- Função para importação de objetos 3D: realiza a importação de modelos no formato 3DS (*Three-Dimensional Studio*). A Figura 6.8 ilustra o efeito da utilização dessa função.

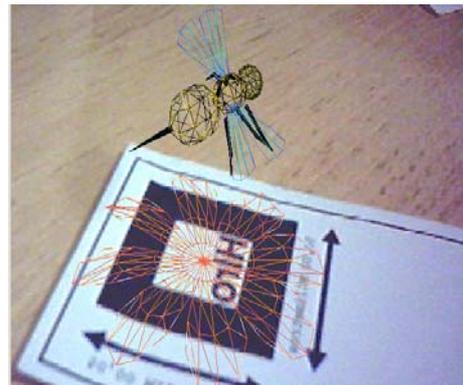


Figura 6.8: Objeto virtual importado adicionado ao ambiente

- Função de exibição em *wireframe*: habilita a eliminação das superfícies dos objetos 3D desenvolvidos em OpenGL e VRML. A Figura 6.9 (a)-(b) ilustra o emprego dessa função em um objeto VRML animado.



a) Objetos com as superfícies preenchidas



b) Objetos em *wireframe*

Figura 6.9: Eliminação de superfícies em objetos virtuais

c) Módulo de Interação

Neste módulo estão presentes as seguintes funções:

- Função de colisão entre esferas: um dos modos mais clássicos de detecção de colisão é aproximar cada objeto, ou parte deste, a uma esfera e verificar se esta esfera intercepta outra. Esse método é extremamente utilizado por exigir pouco esforço computacional. É necessário apenas verificar se a distância entre os centros das duas esferas é menor do que a soma dos raios de ambas, o que indica a ocorrência da colisão. A detecção de colisão entre dois objetos é importante para preservar o realismo.

- Função que calcula a distância entre dois pontos no espaço 3D: usada principalmente para calcular a distância entre os centros de dois marcadores reconhecidos (Figura 6.10).

```
float CMarcas(double Ax,double Ay,double Az,double Bx,double By,double Bz)
{
    float Dist;

    // Teorema de Pitágoras A*A = B*B + C*C
    Dist = sqrt((Ax-Bx)*(Ax-Bx) + (Ay-By)*(Ay-By) + (Az-Bz)*(Az-Bz));

    return Dist;
}
```

Figura 6.10: Função para encontrar a distância entre os centros dos marcadores

- Função de quadros por segundo: calcula a taxa de quadros por segundo de uma aplicação de RA.

6.3 Protótipos desenvolvidos

Segundo Tissiani et al. (2004), o projeto da interface com o usuário se encerra com a escolha e a configuração da apresentação e do comportamento dos objetos de interação relacionados às ferramentas e dos objetos que caracterizam uma unidade de apresentação. A descrição dos objetos deve incluir, além de sua imagem, o seu comportamento em termos de habilitação (quando se torna habilitado e quando se torna desabilitado), o foco (quando possui o foco das ações do usuário ou se torna uma opção padrão), e a condução ao usuário (conteúdo e momento das mensagens de orientação que são enviadas ao usuário).

As várias formas de interação podem ser usadas de diversas maneiras, dependendo do tipo da aplicação desejada. A combinação dessas técnicas de interação propiciam ao usuário elementos interessantes para o exercício da criatividade na elaboração de aplicações de RA.

Os protótipos a seguir visam oferecer manipulações dos objetos virtuais no AV e como citado no item 5, os critérios levantados para o desenvolvimento de uma interface de interação serviram para o desenvolvimento dos mesmos, sendo projetados e implementados para a validação do suporte descrito no item 6.2.

6.3.1 Interface de Controle 2D

O primeiro protótipo de interação foi desenvolvido utilizando-se a biblioteca GLUTI (*Graphic Library User Interface*), a qual permite a construção de controles como botões, menus, caixas de seleção, entre outros (RADEMARCHER, 2005). Os comandos destinados ao objeto virtual são executados a partir dessa interface, o vídeo é projetado dentro de um formulário e, à direita, foi criado um painel onde os controles são desenhados. A Figura 6.11 ilustra essa interface.

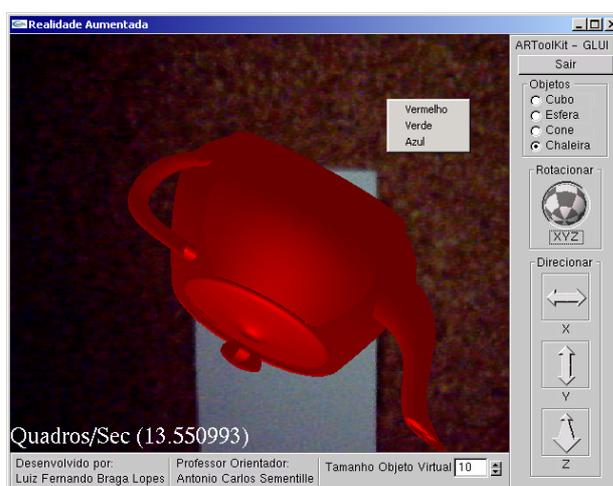


Figura 6.11: Interface de Controle 2D

Os referidos controles fornecem ao objeto virtual ações como a mudança de sua forma entre um cubo, uma esfera, um cone ou uma chaleira, para o mesmo marcador, através de uma caixa de seleção. Outros controles podem rotacionar e transladar, ou seja, mudar a orientação e a posição do objeto virtual. Tais controles são exemplificados a seguir.

- **Caixa de Seleção do Tipo do Objeto Virtual:** implementou-se uma caixa de seleção, na qual o usuário pode escolher qual objeto virtual (Cubo, Esfera, Cone e Chaleira) será projetado sobre o marcador em tempo real.
- **Controle de Rotação:** implementou-se um controle de rotação sobre o objeto virtual escolhido, também conhecido como “*ArcBall*”.
- **Controle de Translação:** foram implementados três controles para o deslocamento do objeto virtual em relação ao marcador, de acordo com os três eixos coordenados.

- **Controle de Escala:** implementou-se um controle que permite a alteração da escala do objeto.
- **Menu de Cores:** ao pressionar o botão direito do *mouse* sobre o vídeo, através de uma caixa de seleção, o usuário muda a cor do objeto virtual.

Nesse protótipo foram utilizados os serviços dos três módulos que compõem o suporte, como ilustra a Figura 6.12.

```

Início do Protótipo
  Declaração das Bibliotecas e das Variáveis:
    Luz (Tonalidade da Iluminação)
    Obj (Tipo de Objeto Virtual)
    Rot (Ângulo de Rotação)
    Tra (Variável de Translação)
    Tam {Variável de Escala}
  Laço Principal
    Criar Janela (Formulário)
    Criar Painel de Controle:
      Caixa de Seleção de Objetos {Obj}
      Controle de Rotação {Rot}
      Controle de Translação {Tra}
      Controle de Escala {Tam}
    Evento Mouse {Luz}
    Evento Vídeo:
      Ilustrar a Taxa de Quadros
      Iluminação dos Objetos (Luz)
      Escolha do Tipo de Objeto (Obj)
      Rotacionar Posição de (Rot)
      Mover Posição de (Tra)
      Desenhar os Objetos Virtuais (Tam)
    Retorna Evento Vídeo
  Fim do Protótipo

```

Figura 6.12: Pseudocódigo do protótipo de Interface de Controle 2D

6.3.2 ARMouse

O protótipo ARMouse (*Augmented Reality Mouse*) permite a alteração em tempo real do objeto virtual, a partir de um *mouse* composto de marcadores. É definido como um dispositivo não-convencional de interação constituído de sete marcadores dispostos em posições distintas (Figura 6.13). Três marcadores centrais são responsáveis pelos eixos X, Y e Z. Dois marcadores nas laterais são responsáveis pela translação do objeto virtual, sendo um responsável pelo aumento de valores e outro pela redução. Outros dois marcadores

posicionados nas laterais opostas, responsáveis pelas rotações, aumentando ou diminuindo os valores dos ângulos.

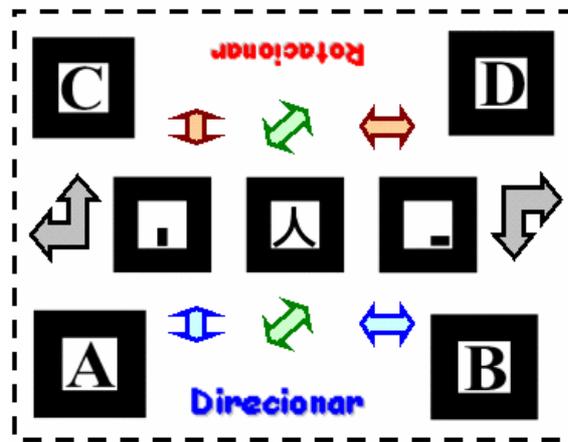


Figura 6.13: Imagem da placa do ARMouse

Este protótipo representa uma forma de interagir com o AV, de modo que o objeto virtual projetado em um marcador execute uma ação no ambiente após a interação com esse dispositivo. O funcionamento básico desse protótipo implica nas oclusões dos marcadores, seguindo uma seqüência lógica para o mesmo, isto é, se o usuário deseja deslocar o objeto virtual, o mesmo deverá ocultar um marcador específico de direcionamento, depois um marcador para qual eixo haverá esse deslocamento.

Seu funcionamento implica, primeiramente, na escolha de qual marcador de mudança de estado, seja ele translação ou rotação, será ocultado. Após a oclusão desse marcador, um segundo marcador, representando o eixo, deverá ser ocultado. Resultando, então, no comportamento do objeto virtual, conforme Figura 6.14 (a)-(b).



a) Direcionador de objetos virtuais



b) Objeto virtual deslocado após interação

Figura 6.14: Interação com dispositivo não-convencional

A Figura 6.15 apresenta o pseudocódigo ilustrando esse protótipo.

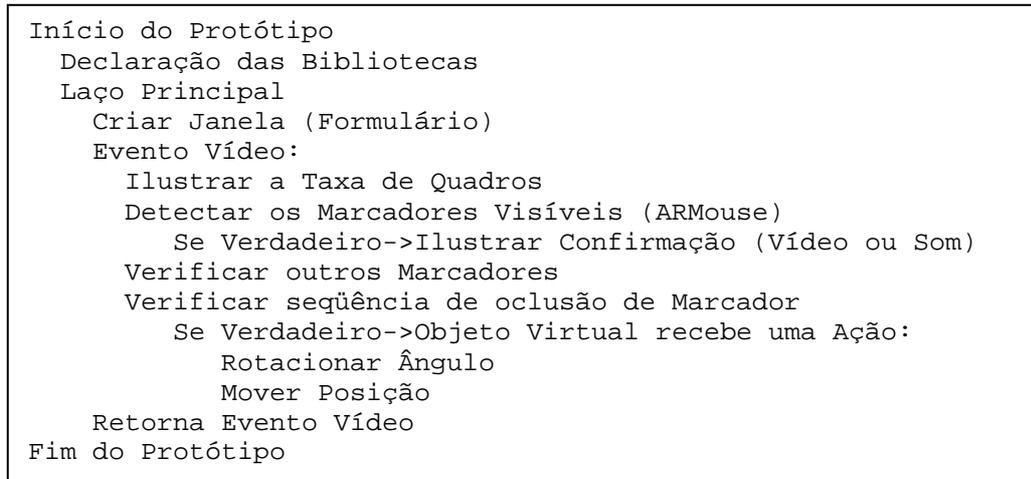


Figura 6.15: Pseudocódigo do protótipo ARMouse

6.3.3 Teclados Virtuais

O protótipo em questão apresenta a criação de teclados virtuais. Primeiramente, foram criados vários marcadores espalhados em um mesmo plano a uma distância pré-determinada entre eles (Figura 6.16). Se existe apenas um marcador e este for oculto pelo usuário, o objeto virtual não será exibido, logo com a utilização de vários marcadores, a aplicação passa para a localização do próximo marcador que ficará responsável pela projeção do objeto virtual.

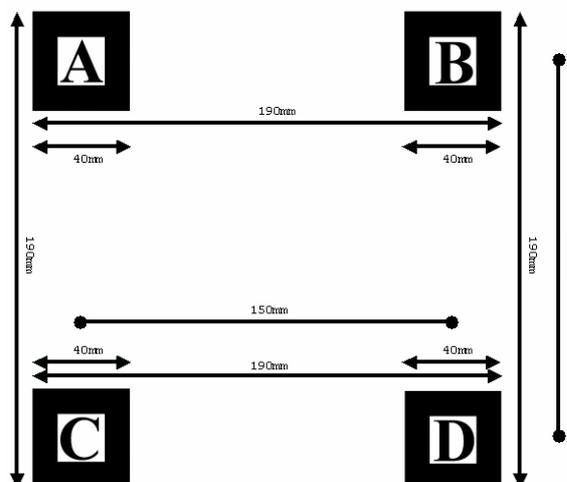


Figura 6.16: Vários marcadores espalhados para evitar oclusão

As formas de interação com esse teclado virtual podem ser feitas utilizando um marcador acoplado ao dedo ou um bastão com um marcador desenhado na ponta, como ilustrado na Figura 6.17 (a)-(b).



a) Marcadores acoplados aos dedos



b) Bastão para interação

Figura 6.17: Componentes de interação utilizando marcadores

Diversos eventos podem ser desenvolvidos para a colisão com o teclado, como, por exemplo, adição de sons, execução de outros programas ou até mesmo a criação de um editor de texto. A Figura 6.18 ilustra uma implementação desse teclado virtual.

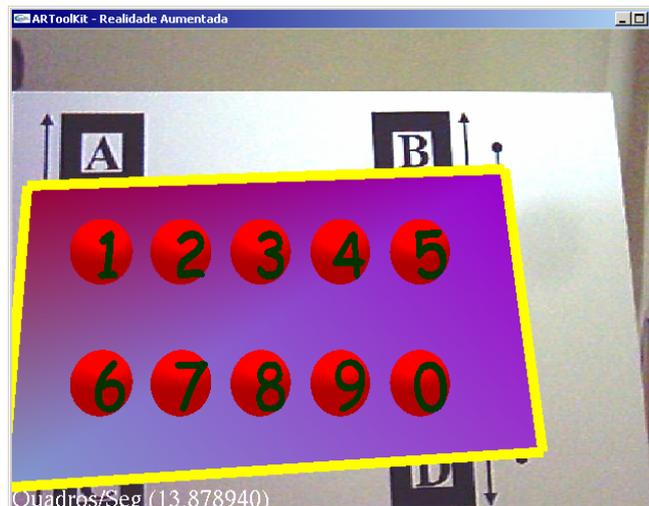


Figura 6.18: Ilustração de um teclado virtual

A primeira função executada nesse protótipo é a verificação de qual marcador está visível, após esta verificação é atribuído um valor para a variável **Painel** responsável pelo marcador ativo. Outras variáveis auxiliares, denominadas de **ValorX**, **ValorY** e **ValorZ**, são utilizadas para sempre posicionar as imagens virtuais ao centro do painel (Figura 6.19). Pode-

se, ao desenhar o painel de seleção, importar uma figura como imagem de fundo para se obter maior realismo da cena.

```

switch (Painel)
{
case 0:
    ValorX = 85; ValorY = 85; ValorZ = 0; break;
case 1:
    ValorX = -85; ValorY = 85; ValorZ = 0; break;
case 2:
    ValorX = 85; ValorY = -85; ValorZ = 0; break;
case 3:
    ValorX = -85; ValorY = -85; ValorZ = 0; break;
}

```

Figura 6.19: Trecho do código fonte do protótipo Teclado Virtual

Ilustra-se o pseudocódigo desse protótipo na Figura 6.20, bem como sua interação com as bibliotecas empregadas.

```

Início do Protótipo
Declaração das Bibliotecas
Laço Principal
Criar Janela (Formulário)
Evento Vídeo:
    Ilustrar a Taxa de Quadros
    Detectar Marcadores no Plano:
        Importar Imagem como Tela de Fundo
        Importar Tecla 01
        Importar Tecla 02
        Importar Tecla 03
        Importar Tecla 04
    Detectar Marcador de Interação:
        Desenhar Objeto Virtual de Interação
        Calcular as Distâncias entre Teclas e Objeto
        Validar as Distâncias (Função de colisão):
            Se Verdadeiro-> Disparar um Evento (Som)
    Retorna Evento Vídeo
Fim do Protótipo

```

Figura 6.20: Pseudocódigo do protótipo Teclado Virtual

6.3.4 Protótipo Pinças Virtuais

Neste protótipo, dois marcadores representam objetos de interação e um marcador representa um objeto fixo. Esses marcadores foram sobrepostos por esferas virtuais onde, após a colisão entre as duas esferas representando os objetos de interação com a esfera fixa, a esfera fixa passava a ter um novo comportamento derivado das ações dos marcadores que

simulam os dedos do usuário: o objeto fixo passa a acompanhar as mesmas mudanças espaciais dos objetos colididos, como o exemplo citado no item 5.7.3. A Figura 6.21 (a)-(b) ilustra esta situação.

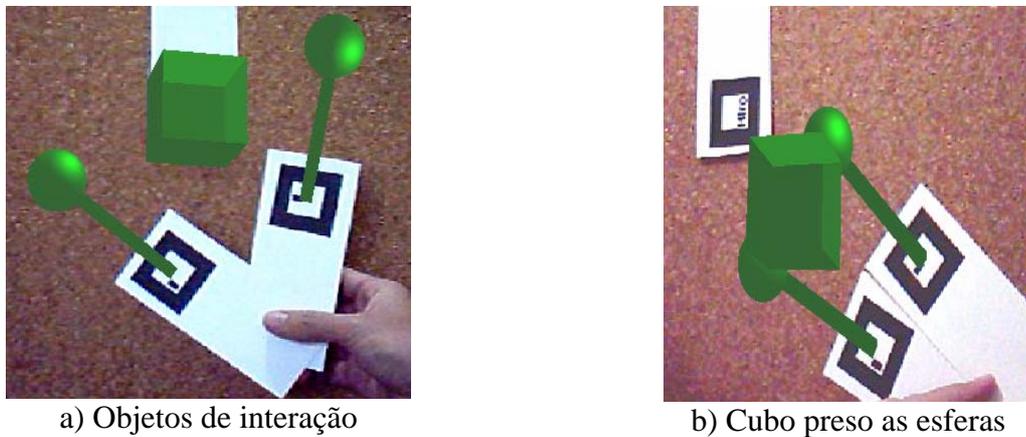


a) Esferas sobrepostas aos marcadores nos dedos b) Objeto virtual sendo movido

Figura 6.21: Interação entre esferas

Uma desvantagem encontrada é relativa aos marcadores e aos objetos virtuais projetados: um objeto virtual terá que ser maior que o marcador, pois a aproximação entre os marcadores acarretará em oclusão de pelo menos um deles.

A fim de solucionar tal problema, no próximo protótipo foram utilizados objetos virtuais de interação projetados a uma certa distância do marcador. Este demonstra duas esferas distanciadas dos marcadores, porém ligadas ao centro do marcador através de um cilindro. O objeto colidido, conforme Figura 6.22 (a)-(b), é um cubo envolvido em uma esfera invisível. Esta esfera envolvente representa as maiores arestas do objeto.



a) Objetos de interação

b) Cubo preso as esferas

Figura 6.22: Interação entre esferas distantes dos marcadores

Ilustra-se o pseudocódigo deste protótipo na Figura 6.23.

```
Início do Protótipo
  Declaração das Bibliotecas
  Laço Principal
    Criar Janela (Formulário)
    Evento Vídeo:
      Detectar Marcador P1
      Desenhar Esfera Virtual
      Detectar Marcador P2
      Desenhar Esfera Virtual
      Detectar Marcador P3
      Desenhar Esfera Virtual
      Calcular as Distâncias: (P1 e P3) (P2 e P3)
      Validar as Distâncias
      Se Verdadeiro  $P3 = P3 - ((P1+P2)/2)$ 
    Retorna Evento Vídeo
Fim do Protótipo
```

Figura 6.23: Pseudocódigo do protótipo Pinças Virtuais

7. TESTE DOS PROTÓTIPOS E ANÁLISE DOS RESULTADOS OBTIDOS

Neste capítulo são apresentados os testes realizados com os protótipos, a fim de analisar a viabilidade e a qualidade dessas aplicações de interação de RA.

Foi montado, para fins de testes, um ambiente de trabalho composto de uma mesa na escala de 40 cm por 80 cm, uma câmera digital, um par de refletores de luz e um painel de material de cortiça para fixação dos marcadores. A Figura 7.1 mostra a imagem desse estúdio e como os elementos estão dispostos. Com a câmera focalizada a uma distância de 60 cm do painel, e centrada com a relação à mesa, foi possível estabelecer um sistema de coordenadas em relação aos marcadores.

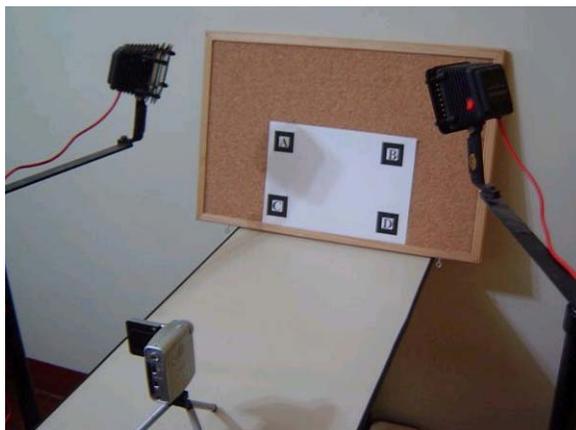


Figura 7.1: Ambiente de trabalho para validação dos protótipos

A resolução de vídeo usada limita-se a 800x600 pontos, frequência de 60hz e profundidade de cores mínima de 32 bits.

7.1 Recursos utilizados

Os principais recursos de *software* utilizados são demonstrados na Tabela VIII.

Tabela VIII: *Softwares* utilizados

Programa	Qualificação	Desenvolvedor	Versão	Objetivo
OpenGL	Biblioteca Gráfica	<i>Silicon Graphics</i>	2.0	Projeção de Matrizes e Renderização de Objetos 3D
DirectX	Biblioteca Multimídia	Microsoft	9b	Suporte aos Dispositivos de vídeo do Sistema Operacional
ARToolKit	Biblioteca de RA	Universidade de Washington	2.52	Desenvolvimento de Sistemas de RA
Visual C++	Gerador de Aplicações	Microsoft	6.0	Desenvolvimento de Sistemas baseados em linguagem C e C++
GLUI	Biblioteca Gráfica	Paul Rademacher	2.0	Desenvolvimento de Interfaces Gráficas para programas em Linguagem C e C++
Windows 2000 Professional	Sistema Operacional	Microsoft	2000	Sistema Operacional utilizado no desenvolvimento do Trabalho

Os principais recursos de *hardware* utilizados são:

- Câmera digital, tipo Webcam, da marca Aiptec, modelo DV3100, de resolução de 3.1 megapixels;
- Placa aceleradora gráfica, da marca Asus, modelo GeForce4 MX, 64 megabytes de memória, otimizada para DirectX e OpenGL;
- Monitor de vídeo da marca LG, modelo Flatron E701S de 17 polegadas; e
- Computador PC (*Personal Computer*) com 512 megabytes de memória RAM (*Random Access Memory*) e processador Athlon 2.4 GHz.

7.2 Testes realizados e resultados obtidos

Como descrito no item 6.2, o suporte apresenta algumas características multimídia como: funções de som, reproduções de vídeo e a introdução de diversas formas geométricas dispostas como objetos virtuais 3D. Além de prover funções matemáticas de colisões, abastecidas pelos pontos de referências (marcadores) a fim de interagir com outros objetos virtuais no AV.

Avaliou-se o quanto o número de polígonos (triângulos) de um objeto virtual pode influenciar a qualidade de uma aplicação de RA, resultando no valor obtido pela taxa de quadros por segundo. Ou seja, prover uma condição interessante de imersão em uma interface genérica, questionando, assim, a viabilidade dos sistemas de RA apresentados.

Para essa avaliação, realizou-se um teste de referência comum aos exemplos que acompanham a biblioteca ARToolKit. Para isso, foi utilizado somente um marcador (Figura 7.2), medindo 10,79 cm, que foi adicionado ao painel de testes. O painel encontra-se a uma distância de 86,36 cm a frente da câmera de vídeo. Essas medidas foram sugeridas pelos autores da biblioteca ARToolKit, conforme citado no item 4.5.



Figura 7.2: Marcador utilizado para teste inicial

Algumas informações técnicas consideradas representativas devem ser citadas: a resolução da câmera de vídeo é de 640x480 *megapixels* e o nível de *threshold* da aplicação é de 100 (taxa de iluminação que varia de 0 a 255).

A cada polígono adicionado à cena, a taxa de quadros por segundo era medida, conforme mostra a Figura 7.3.

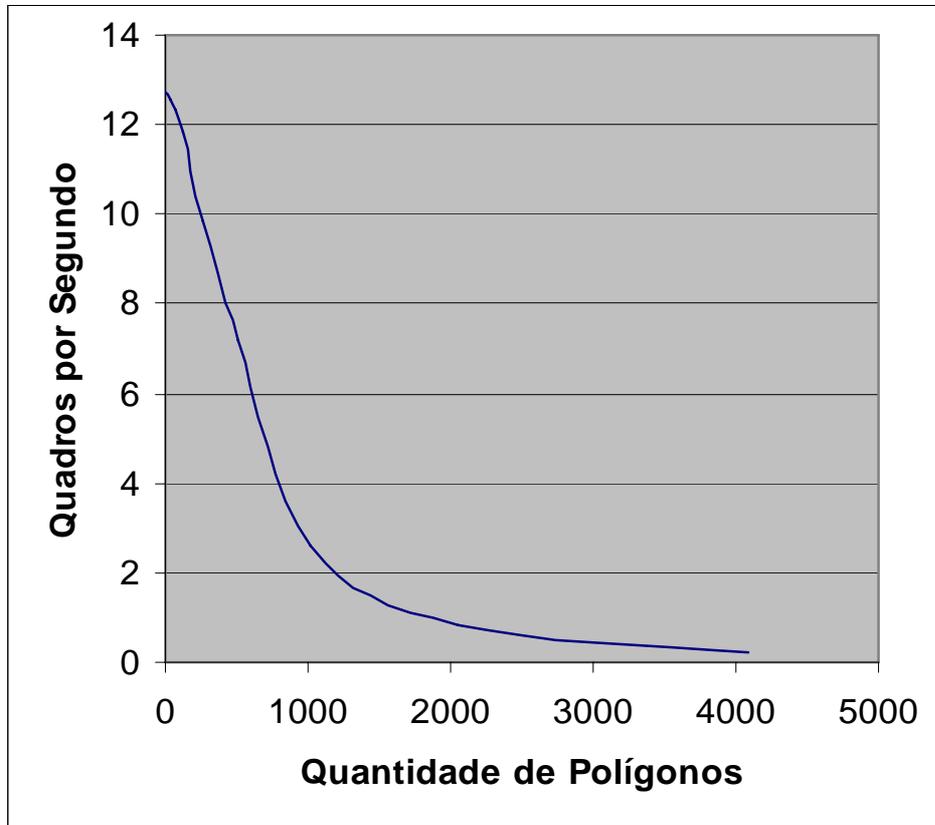


Figura 7.3: Gráfico de desempenho do teste de referência

- **Protótipo Interface de Controle 2D**

Para este protótipo, funções para a criação de componentes de controle (botões, caixa de seleção, painéis) da biblioteca GLUI foram utilizadas, como também a introdução do objeto virtual, medindo-se então a quantidade de polígonos pela taxa de quadros por segundo, empregando-se o mesmo marcador e as mesmas medidas utilizadas no teste de referência.

Comparando-se os resultados deste teste com os valores do teste anterior conclui-se que a taxa de quadros por segundo foi afetada significativamente pela utilização da biblioteca gráfica (GLUI) no desenvolvimento da interface. Os valores obtidos são expressos no gráfico da Figura 7.4.

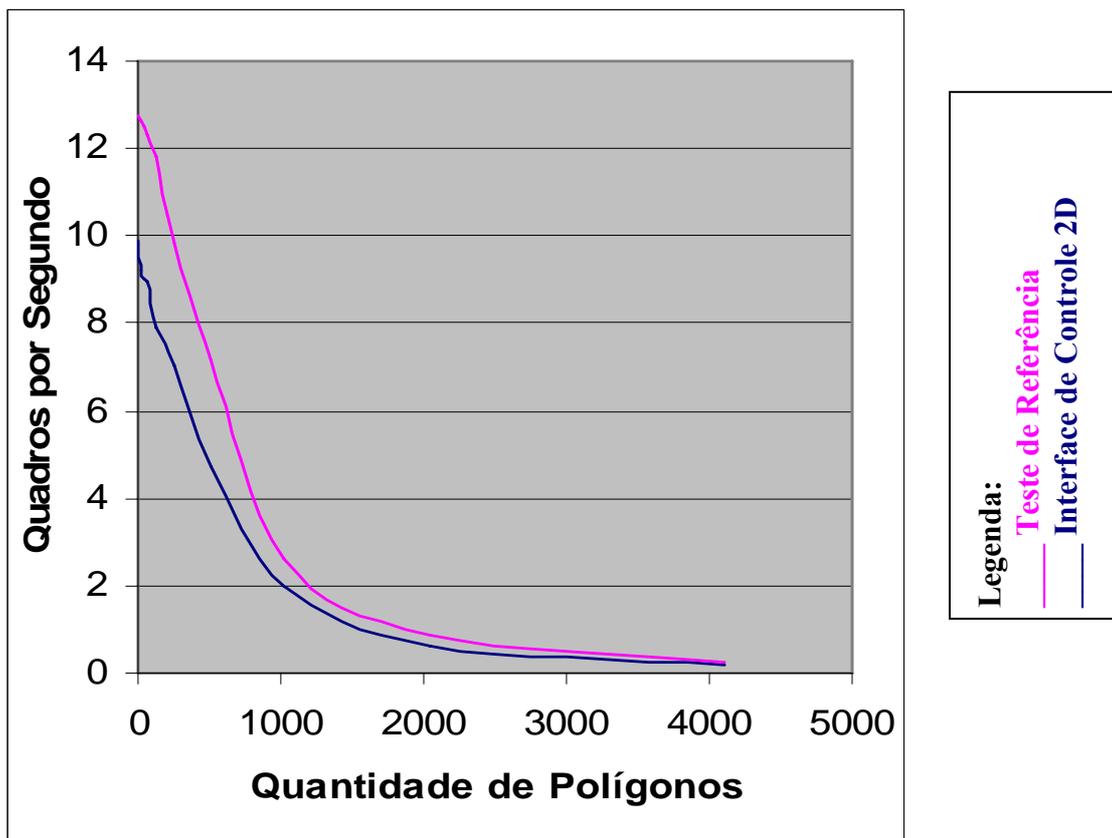


Figura 7.4: Gráfico de desempenho da Interface de Controle 2D

Em relação ao teste desse protótipo, buscou-se também avaliar algumas características que emergiram em função das interações através dos controles externos de manipulação do objeto virtual.

Sobre as simulações realizadas, esse protótipo expõe o impacto da utilização de controles externos de interação, logo o funcionamento desse protótipo apresenta uma desvantagem: os componentes de interação se aplicam a somente um marcador, ou seja, as modificações comportamentais seriam de um só objeto virtual. Torna-se uma aplicação inflexível, pois seu estado não é influenciado por outros objetos virtuais durante a simulação, somente pelos componentes de controle.

Analisando outros aspectos do funcionamento dessa interface, as formas de interação partem do princípio de que o usuário dispõe de outro dispositivo para a interação, como um *mouse* ou um teclado, para executar a modificação no objeto virtual.

- **Protótipo ARMouse**

Em relação a este teste levantou-se uma aplicação descrita por Umlauf (2002), em que marcadores dispostos em regiões específicas de estantes em uma biblioteca auxiliam o usuário identificar a localização de um livro do seu interesse (Figura 7.5). Inicialmente, o usuário conta com um painel virtual para escolha da obra, após a seleção os marcadores espalhados no recinto são sobrepostos por sinais de localização como setas a fim mostrar a posição espacial do item, facilitando assim sua busca.

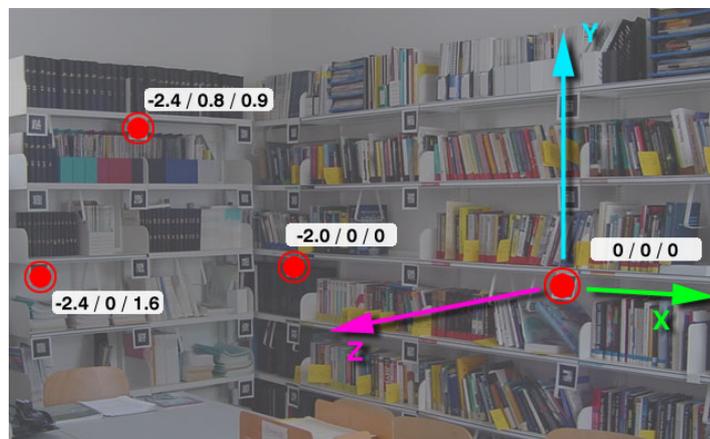


Figura 7.5: Marcadores espalhados no interior de uma biblioteca (UMLAUF, 2002)

Logo para a avaliação deste protótipo, discutiu-se a necessidade de quantos marcadores poderiam ser utilizados em uma aplicação de RA sem prejudicar o realismo da cena, simulando assim mais funções para cada marcador.

A Figura 7.6 ilustra o modelo de teste para esse protótipo, validando até quantas opções pode-se ter em um protótipo como esse.

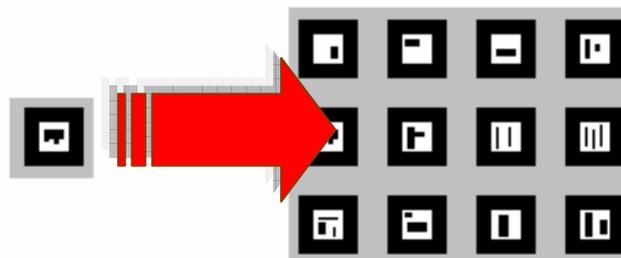


Figura 7.6: Modelo de teste para o protótipo ARMouse

Inicialmente, foram utilizados 30 marcadores com o tamanho de 40mm para este teste, a uma distância de 50cm em relação ao dispositivo de captura. Os marcadores eram adicionados à cena e a taxa de quadros por segundo era medida. A Figura 7.7 ilustra o gráfico de desempenho dos resultados obtidos.

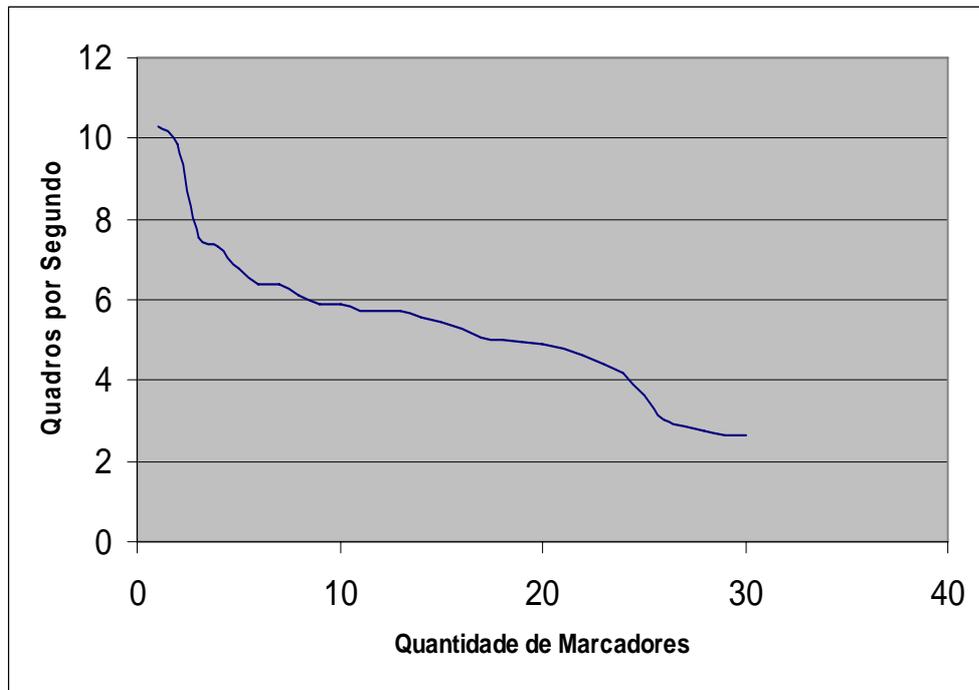


Figura 7.7: Gráfico de desempenho do teste de marcadores

Ao avaliarem-se os valores obtidos, levantaram-se alguns pontos importantes: o resultado do processamento de muitos marcadores é uma questão muito significativa. Devido ao reconhecimento de vários marcadores, ocorreu uma diminuição significativa na taxa de quadros por segundo. Foi avaliada, também, a distância máxima em que os marcadores mostravam um grau de confiabilidade de reconhecimento.

Foram desenhadas, então, esferas de cores distintas sobre os marcadores e, à medida que a distância do painel aumentava, as esferas começaram a trocar de cor (os marcadores se confundiam), resultando na perda do grau de confiabilidade do sistema. A Tabela IX ilustra a distância e quantos marcadores permaneceram confiáveis (sem a mudança do objeto virtual).

Tabela IX: Relação entre marcadores e a distância útil

Distância (cm)	Marcadores
50	30
60	28
70	21
80	20
90	19
100	16
110	10
120	8
130	3

Pode-se concluir que a distância interessante para uma quantidade de 30 marcadores do tamanho utilizado ficou entre 50cm à 80cm.

Ao se trabalhar com vários marcadores, encontram-se alguns pontos desfavoráveis a respeito do uso desse sistema. O primeiro ponto é a necessidade de um processamento maior no reconhecimento das imagens (marcadores). Outro ponto é a instabilidade provocada pela variação na luminosidade do ambiente real, pois, dependendo de sua intensidade, poderá tornar o sistema inoperante, impossibilitando a identificação dos marcadores. Por fim, a oclusão indevida de alguns marcadores, podendo fornecer valores de deslocamento ao objeto virtual, a disposição das teclas tem de estar em posições diferentes para não haver oclusão indevida.

Sobre os resultados referentes à simulação do ARMouse, discute-se a necessidade do mesmo ser projetado sobre um material que ofereça uma superfície rígida, pois pequenas ondulações podem significar o não reconhecimento de um marcador. Assim como a forma de interação não deve ser feita diretamente pelas mãos do usuário, pois qualquer incidência de sujeira ou impregnação de qualquer substância também prejudica o reconhecimento do marcador, tornando o sistema inoperante.

Uma das vantagens encontradas nesse protótipo é a ausência de controladores físicos, como o *mouse*. A maior desvantagem é o processamento de muitos marcadores, pois, além dos marcadores deste objeto de interação, incluem-se os marcadores dos objetos virtuais que sofrerão as mudanças de seu comportamento.

- **Protótipo Teclado Virtual**

Com o protótipo sobre teclados virtuais seguiu-se o mesmo princípio de colisões entre esferas, mas um grande problema foi encontrado: se o objeto virtual fosse projetado no centro do marcador, quando houvesse aproximação de outro marcador, haveria oclusão, impossibilitando a interação com o mesmo. Para resolver essa dificuldade, foram distribuídos uniformemente 4 marcadores distintos, simetricamente afastados.

Para avaliação desse protótipo, foram utilizados 4 marcadores de tamanho 40mm, a uma distância de 85mm entre seus centros, representando o painel para sobreposição do teclado virtual, e 1 marcador de tamanho 40mm, representando o objeto de interação. Todos dispostos a uma distância de 40cm da câmera de captura.

O objetivo desse teste foi de montar um teclado virtual e verificar quantas teclas (quantos números de opções de seleção) podem ser utilizadas sem prejudicar a taxa de quadros por segundo, sem anular o grau de realismo, pois quanto maior a quantidade de esferas utilizados em um sistema de RA como ponto de referência, maior será o número de vezes em que haverá o cálculo de distâncias entre o objeto alvo (esferas) e o objeto de interação.

A finalidade foi de avaliar o tempo geral (em quadros por segundo) em como a quantidade de esferas poderiam ser inseridas em um sistema de RA sem prejudicar o grau de realismo. O cálculo partiu de uma esfera de referência denominada **T** para as esferas de opções que eram inseridas na cena, até uma quantidade **n** de esferas.

A Figura 7.8 ilustra esse modelo de avaliação para o protótipo.

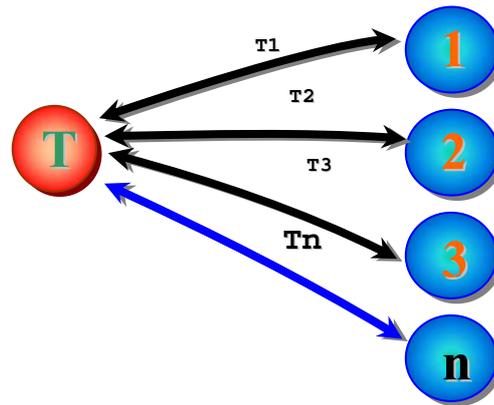


Figura 7.8: Modelo do teste do protótipo Teclado Virtual

A Figura 7.9 ilustra o teclado virtual, os valores referentes a esse teste são ilustrados na Figura 7.10, na qual o gráfico representa uma escala de quantidade de esferas (número de opções que o usuário poderia ter) pela quantidade de quadros por segundo na cena.

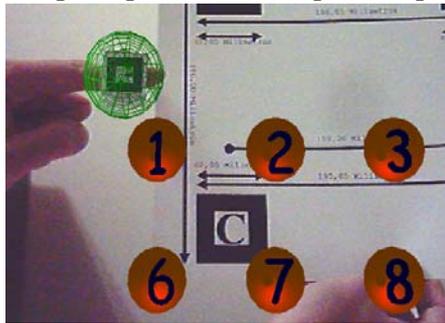


Figura 7.9: Protótipo Teclado Virtual utilizado para teste

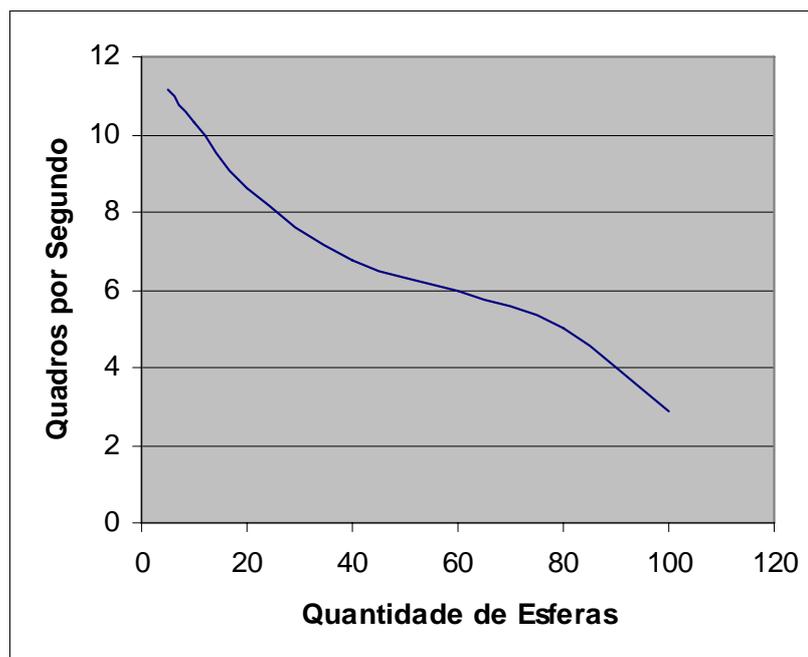


Figura 7.10: Avaliação do protótipo Teclado Virtual

Após os resultados do teste do protótipo teclado virtual, pode-se avaliar que uma quantidade adequada de esferas, ou seja, opções, situa-se na faixa de 20 esferas (considerando a média de 8 à 10 quadros por segundo) para se ter um bom realismo da cena. Algumas outras observações em relação a esse protótipo são descritas a seguir.

A forma de seleção do protótipo teclado virtual se resume ao contato com o objeto, sendo a seleção realizada pela proximidade espacial, na qual o ponto de intersecção do objeto alvo (tecla virtual) e do objeto de interação (dedo) para a tarefa de seleção deve ser óbvio e acessível.

Um das preocupações da utilização de um protótipo como esse é se o usuário do sistema é leigo em informática, pois os objetos virtuais desenhados na cena devem prover uma familiaridade aos usuários.

Seu comportamento deve ser definido como um conjunto de ações que um objeto executa quando exposto a um determinado estímulo, podendo ou não, mudar seu estado. As atribuições de comportamentos aos objetos permitem uma interatividade maior do usuário com o AV. Pode-se ainda utilizar um objeto virtual de interação transparente para evitar oclusão por outro objeto virtual durante a seleção.

É de fundamental importância um retorno visual ou sonoro, ou seja, uma forma através do qual o resultado pode ser expresso e assimilado pelo usuário após a interação. Prover ao usuário um retorno de seleção apropriado, como, por exemplo, um som, uma mudança de cor ou uma mensagem de confirmação se faz necessário.

- **Protótipo Pinças Virtuais**

Para esse protótipo foram utilizados 2 marcadores de tamanhos 20mm e 1 marcador de tamanho 40mm, e a distância em relação à câmera é de 50cm, sendo 1 marcador para o objeto virtual e os outros dois marcadores (sem desenhos de objetos virtuais) apenas utilizados para a

aproximação de colisão entre os marcadores, a fim de não interferir na quantidade de polígonos que eram adicionadas a cena (Figura 7.11). Apenas uma mudança de cor para o objeto virtual foi utilizada para confirmar a colisão.

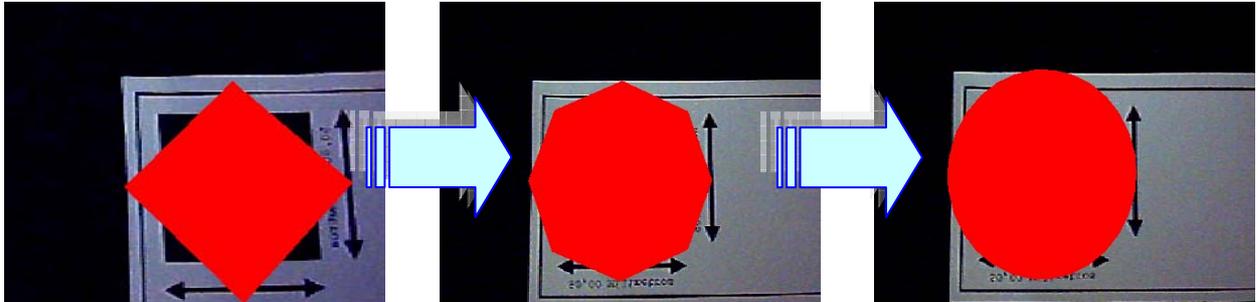


Figura 7.11: Polígonos adicionados a cena

Ao serem analisados os valores finais obtidos, por maior que fosse a quantidade de polígonos e os atrasos na taxa de quadros por segundo, a função de colisão continuava funcionando. Um ponto positivo é a utilização desse modelo de função de colisão, pois demonstrou uma rápida resposta na interação. A Figura 7.12 ilustra a projeção dos valores obtidos.

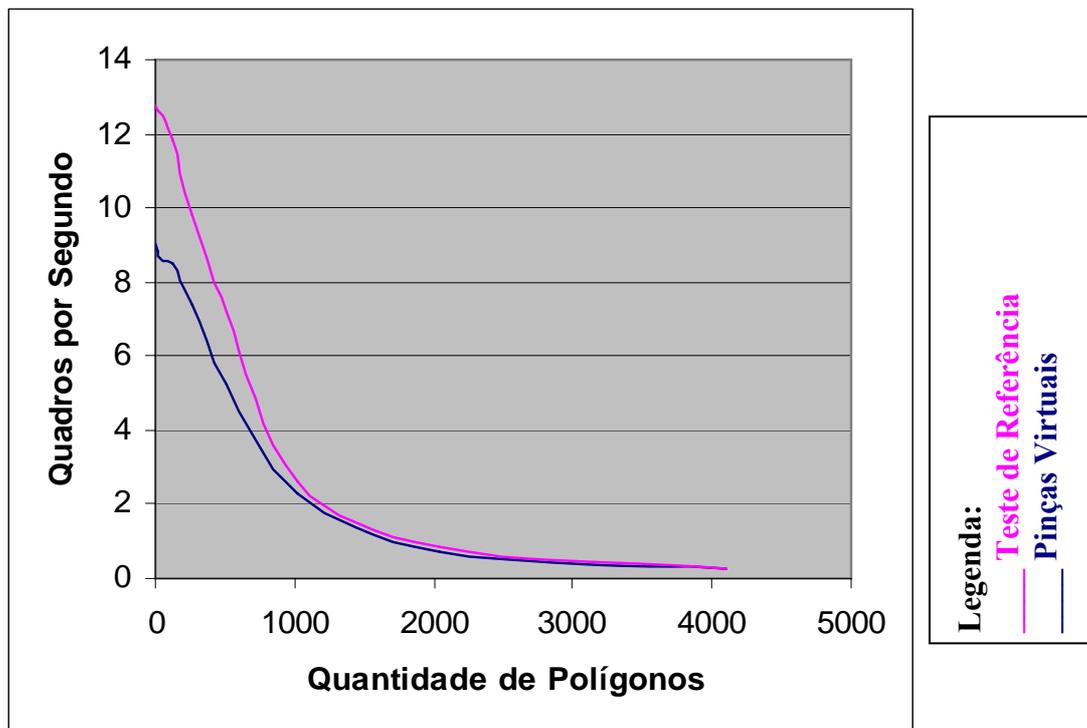


Figura 7.12: Avaliação do protótipo Pinças Virtuais

Um ponto favorável, observado após a execução deste protótipo, é a ausência de *hardwares* específicos de manipulações (como luvas ou sensores ópticos) substituídos por marcadores acoplados aos dedos.

7.3 Resumo dos protótipos apresentados

Uma análise final sobre os valores obtidos nos testes realizados revelou que o sistema sempre perde o grau de qualidade quando, em uma aplicação, o objeto virtual excede 1000 polígonos. A Tabela X demonstra um breve resumo dos protótipos apresentados.

Tabela X: Resumo dos protótipos

Protótipo	Característica	Vantagem	Desvantagem
Interface de Controle 2D	Componentes de controle 2D	Mudança das características do objeto virtual em tempo real, como forma, tamanho, posição espacial e cor.	Necessita a utilização de dispositivo de entrada convencional, como o <i>mouse</i> , além de modificar o comportamento de somente um objeto virtual.
Pinça Virtual	Interação por Colisão	Algoritmo de simples implementação provendo a ausência de luvas para interação.	O objeto virtual deve ser maior que o marcador, além de prover uma colisão aproximada para objetos diferentes de esferas.
Teclado Virtual	Seleção de opções em teclas interativas	Expansão das possibilidades de visualização de informação (representação de dados virtuais 3D).	Implementar um comportamento de retorno adequado.
ARMouse	Dispositivo não-convencional de interação por oclusão	Independência de controles físicos de maior grau de liberdade.	Maior processamento por reconhecer vários marcadores, além da oclusão indevida do marcador remetendo um valor ao sistema.

Com o desenvolvimento dos protótipos aqui apresentados, estendem-se estes resultados a fim de permitir que programadores projetem diferentes comportamentos de interação sobre múltiplos focos de ações, ou seja, na concepção de interfaces de interação

para ambientes de RA. Com a utilização da biblioteca ARToolKit, pode-se criar uma grande diversidade de ambientes apropriados aos mais variados tipos de aplicações de RA.

O maior desafio, entretanto, é metodológico. Ainda há compreensão insuficiente de características essenciais e parâmetros em manipulação na RV. Embora manipulações imersivas sejam similares a manipulações do mundo real, existem também diferenças significativas que devem ser estudadas e entendidas para a exploração de todo o potencial de uma tecnologia de RV (POUPYREV et al., 1997).

Finalmente, a última seção apresenta conclusões a respeito das abordagens propostas e discute a continuidade do trabalho.

8. CONCLUSÕES

Os sistemas de RA propiciam ao usuário interações de maneira intuitiva com o ambiente misturado, usando as mãos com marcadores ou com simples objetos de controle, capturados por uma câmera ou por um HMD, não exigindo nenhum equipamento especial para as interações nas aplicações desenvolvidas.

Este capítulo finaliza o trabalho, apresentando os aspectos gerais, as contribuições, as pesquisas futuras e os comentários finais sobre as interações em sistemas de RA.

8.1 Aspectos gerais do trabalho

Embora as técnicas de interação possibilitem a investigação de vários outros tipos de simulações, as avaliações dos protótipos mostraram novas alternativas para otimizar o processo de interação em sistemas de RA, bem como novos caminhos no sentido de viabilizar o uso desse tipo de recurso em aplicações presentes no cotidiano, utilizando-se de equipamentos comumente encontrados no mercado. Sabe-se que a tarefa de simular o comportamento dos usuários ainda não está concluída. Devido a sua enorme complexidade, o que se fez foram muitas simplificações nos protótipos de interação. Apesar disso, acredita-se que uma gama de aplicações poderia se beneficiar com os serviços do suporte implementado, que podem variar desde jogos a simulações da vida real.

A partir das observações levantadas nesta dissertação, é possível partir para pesquisas mais detalhadas, seguidas por experimentos mais profundos, para implementar os recursos

discutidos em simulações reais, utilizando-se das possibilidades oferecidas pela utilização da biblioteca ARToolKit.

8.2 Contribuições

Novas formas de interação em sistemas computacionais estão minimizando a distância existente entre o relacionamento do homem com o computador, de forma a facilitar a sua usabilidade. Consoante a esse contexto está a RA que possibilita a visualização e a manipulação de objetos virtuais modelados por computador no mundo real. Com isso, é possível desenvolver ferramentas que auxiliem o processo de construção do conhecimento do usuário.

A principal contribuição desta dissertação foram as exposições de paradigmas de interação que visam descrever modelos de forma a permitir a geração de sistemas de RA, cujas áreas de aplicabilidade vão desde simulações até a desafiadora área de jogos. Assim, o usuário poderá estar mais entretido no manuseio dessas aplicações, dado o aumento do grau da interação com o computador, o que é considerado um benefício aos usuários de sistemas computacionais.

8.3 Pesquisas futuras

As interações em sistemas de RA possuem grande potencial para exploração de suas funcionalidades, melhoramento das técnicas já existentes e proposições de novas aplicações. Este trabalho colabora para a expansão dos modelos de representação de objetos virtuais, propondo novas técnicas e funcionalidades e, também, criando protótipos que consolidem os trabalhos relacionados, existindo diversos pontos que podem ser melhorados ou explorados de forma a estender a capacidade das proposições apresentadas, tornando-as mais completas.

Algumas das extensões são dispostas a seguir, e servem como orientação para a continuação dos trabalhos para interação em sistemas de RA:

- Implementação de algoritmos de segmentação, como *Octrees*, sendo essencial que se possam extrair características precisas destes, com a intenção de potencializar os acertos do sistema, que não demandem muito esforço computacional que comprometam a qualidade da aplicação, como: a redução de quadros por segundo perdendo a percepção de realismo;
- Implementação de uma luva virtual com todos os ligamentos acoplados por marcadores ilustrados na Figura 8.1(a), assim o usuário é provido com a representação de uma mão virtual no AV ilustrado na Figura 8.1(b) proporcionando uma alternativa para este equipamento de interação;

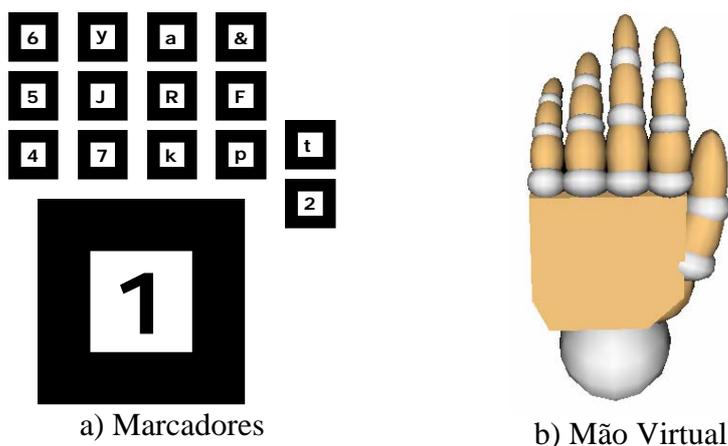


Figura 8.1: Futura implementação de uma Luva Virtual

- Desenvolvimento de técnicas que permitam implementar outros efeitos de iluminação, como reflexão, refração e sombras, para obtenção de maior realismo nas interações com os objetos virtuais;
- A geração de sensações de tato através de dispositivos eletrônicos;
- Adição de uma ou mais câmeras para minimizar o problema de oclusão dos marcadores, descrito no item 4.5; e

- Suporte a outras opções de formatos de imagens e vídeos, para proporcionar mais recursos visuais ao AV.

Permanecem ainda muitos problemas a serem resolvidos, como a necessidade de adequar as representações virtuais de forma que possam ser utilizadas mais ergonomicamente pelos usuários, como uma possível proposta de avaliação de técnicas de interação (item 5.1).

8.4 Comentários finais

As evoluções dos sistemas computacionais têm permitido estabelecer melhores relações de interação entre o homem e o computador, o que, conseqüentemente, possibilita o aumento no número de usuários, dado que interfaces mais intuitivas e interativas requerem cada vez menos pessoas altamente especializadas para sua manipulação, não restringindo, assim, o seu uso.

Dessa mesma maneira, nota-se o crescente número de trabalhos que incorporaram o computador para o seu desenvolvimento, proporcionando, dessa forma, um aumento no grau de dependência do homem em relação à máquina. Portanto, dado à crescente demanda pelo computador, é necessário criar ambientes mais interativos, a fim de que o homem minimize possíveis dificuldades na manipulação da máquina e, por conseguinte, execute seu trabalho de modo mais eficiente e satisfatório.

Durante a execução deste trabalho, diversas informações foram coletadas servindo como base de dados para nossa conclusão, indicando um caminho promissor a ser seguido dentro da pesquisa científica na área de RA: a geração de interfaces de interação para AV.

9. REFERÊNCIAS

AZUMA, Ronald T. *A Survey of Augmented Reality*. Teleoperators and Virtual Environments v. 355-385. Ago. 1997. Disponível em: <<http://www.cs.unc.edu/~azuma/ARpresence.pdf>>. Acesso em: 29 dez. 2004.

AZUMA, Ronald T. et al. *Recent Advances in Augmented Reality*. IEEE, Computer Graphics and Applications, Nov, 2001. Disponível em: <<http://www.cs.unc.edu/~azuma/cga2001.pdf>>. Acesso em: 29 dez. 2004.

BADIQUÉ, Eric; RACZYNSKI, Artur. *STAR - Service and Training through Augmented Reality*. Ago, 2002. Disponível em: <http://review.realviz.com/Documents/STAR_summary_01_08_02.pdf>. Acesso em: 29 dez. 2004.

BONSOR, Kevin. *How Augmented Reality Will Work*. Disponível em: <<http://computer.howstuffworks.com/augmented-reality2.htm>>. Acesso em: 29 dez. 2004.

BOWMAN, Doug A. et al. *Interaction Techniques for Common Tasks in Immersive Virtual Environments: design, evaluation, and application*. Georgia Institute of Technology, 1999.

BRUEGGE, Bernd; KLINKER, Gudrun. *DWARF Distributed Wearable Augmented Reality Framework*. 2001. Disponível em: <<http://www.bruegge.in.tum.de/pub//DWARF/WhitePaper/dwarf.pdf>>. Acesso em: 29 dez. 2004.

BUCHMANN, Volkert. et al. *FingARtips - Gesture Based Direct Manipulation in Augmented Reality*. ACM, Proceedings of the 2nd International Conference on Computer Graphics and Interactive Techniques, Singapore, p. 212-221, Jun 2004.

BURDEA, Grigore C.; COIFFET, Philippe. *Virtual Reality Technology*. Wiley-IEEE Press, 2. ed. 2003.

CeBIT. *Virtual Mosquito Hunt: An Augmented Reality Game*. 2003. Disponível em: <http://w4.siemens.de/en2/html/press/newsdesk_archive/2003/e_0311_d.html>. Acesso em: 29 dez. 2004.

COOPER, A. *About face: The essential of user interface design*. IDG Books, 1997.

COSTELLO, Patrick. *Health and safety issues associated with virtual reality - A review of current literature*. JISC Advisory Group on Computer Graphics, Technical Report No.37, 1997.

FJELD, Morten; VOEGTLI, Benedikt M. *Augmented Chemistry: An Interactive Educational Workbench*. ISMAR 2002, Darmstadt, Germany, Set. 30, 2002. Disponível em: <<http://csdl.computer.org/comp/proceedings/ismar/2002/1781/00/17810259.pdf>>. Acesso em: 29 dez. 2004.

GLEUE, Tim; DÄHNE, Patrick. *Design and Implementation of a Mobile Device for Outdoor Augmented Reality in the ARCHEOGUIDE Project*. Nov, 2001. Disponível em: <<http://archeoguide.intranet.gr/papers/publications/ARCHEOGUIDE-VAST01-3.pdf>>. Acesso em: 29 dez. 2004.

HÖLLERER, Tobias. et al. *ARWin - A Desktop Augmented Reality Window Manager*. ISMAR, Mai, 2003. Disponível em: <http://www.cs.ucsb.edu/~sdiverdi/arwin/pub/arwin_ismar03.pdf>. Acesso em: 29 dez. 2004.

ISDALE, Jerry. *What is virtual reality?: a web-based introduction*. 1998. Disponível em: <<http://www.isdale.com/jerry/VR/WhatIsVR/frames/WhatIsVR4.1.html>>. Acesso em: 29 dez. 2004.

KATO, Hirokazu. et al. *ARToolKit*. Disponível em: <<http://www.hitl.washington.edu/people/grof/SharedSpace/Download/ManualPC2.11.doc>>. Acesso em: 14 jan. 2004.

KATO, Hirokazu; BILLINGHURST, Mark. *Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System*. Proceedings of IWAR 99, San Francisco, California, 1999.

KIRNER, Cláudio. *Realidade virtual: dispositivos e aplicações*. Disponível em: <http://www.realidadevirtual.com.br/publicacoes/apostila_rv_disp_aplicacoes/apostila_rv.html>. Acesso em: 14 jan. 2004.

LEDERMANN, F. et al. *Dynamically Shared Optical Tracking*. Vienna University of Technology, 2002. Proceedings of the First IEEE International Workshop on ARToolKit (ART02).

LUZ, Rodolfo Pinto. *Proposta de especificação de uma plataforma de desenvolvimento de ambientes virtuais de baixo custo*. 1997. 128 f. Dissertação (Mestrado em Engenharia de Produção) - Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 1997.

MILGRAM, Paul; COLQUHOUN JR., Herman. *A Taxonomy of Real and Virtual World Display Integration*. ISMAR 1999, [S.l.], 1999. Disponível em: <http://vered.rose.utoronto.ca/publication/1999/Milgram_Colquhoun_ISMR1999.pdf>. Acesso em: 29 dez. 2004.

MILGRAM, Paul; DRASCIC, David. *Perceptual issues in augmented reality*. SPIE, San Jose, v. 2653, p. 123-134, Fev. 1996.

MINE, Mark R. *Virtual environment interaction techniques*. UNC Chapel Hill CS Dept.: Technical Report TR95-018. 1995.

OWEN, Charles. et al. *Imagetclar: a blended script and compiled code development system for augmented reality*. STARS 2003, Tokyo, Oct. 2003. Disponível em: <<http://stars2003.cs.tum.edu/finals/stars-owen-et-al-imagetclar.pdf>>. Acesso em: 29 dez. 2004.

PIEKARSKI, Wayne; THOMAS, Bruce H. *Tinmith-evo5 A Software Architecture for Supporting Research Into Outdoor Augmented Reality Environments*. South Australia, 2001. Disponível em: <<http://citeseer.ist.psu.edu/cache/papers/cs/26474/http://zSzzSzwww.tinmith.netzSzpaperszSzpiekarski-tr-arch-2001.pdf/tinmith-evo.pdf>>. Acesso em: 29 dez. 2004.

PINHO, Márcio Serolli. *Interação em Ambientes Tridimensionais*. WRV, 2000, Tutorial. Disponível em: <<http://www.inf.pucrs.br/~pinho/3DInteraction/>>. Acesso em: 29 dez. 2004.

PONDER, Michal. et al. *VHD++ Development Framework: Towards Extendible, Component Based VR/AR Simulation Engine Featuring Advanced Virtual Character Technologies*. CGI, Jul, 2003. Disponível em: <http://vrlab.epfl.ch/Publications/pdf/Ponder_and_al_CGI_2003.pdf>. Acesso em: 29 dez. 2004.

POUPYREV, Ivan. et al. *A Framework and Testbed for Studying Manipulation Techniques for Immersive VR*. ACM, 1997, p 21 - 28, Lausanne, Switzerland. Disponível em: <<http://www.mic.atr.co.jp/~poup/research/papers/vrst97.pdf>>. Acesso em: 29 dez. 2004.

RADEMACHER, Paul. *GLUI User Interface Library*. Versão 2.1. Disponível em: <<http://www.cs.unc.edu/~rademach/glui/>>. Acesso em: 24 mar. 2005.

RASKAR, Ramesh; BIMBER, Oliver. *Modern Approaches to Augmented Reality*. 25th Annual Conference of the European Association for Computer Graphics EUROGRAPHICS, Grenoble France, 2004.

SPLECHTNA, Rainer. et al. *ARAS - Augmented Reality Aided Surgery System Description*. 2002. Disponível em: <http://www.vrvis.at/TR/2002/TR_VRVis_2002_040_Full.pdf>. Acesso em: 29 dez. 2004.

STRADIOTTO, César Ramirez Kejelin. *Biblioteca para criação de jogos utilizando geração pseudorandômica paramétrica, realidade virtual, inteligência artificial e redes de computadores*. 2002. 186 f. Dissertação (Mestrado em Engenharia de Produção) - Engenharia de Produção, Universidade Federal de Santa Catarina, Florianópolis, 2002.

SZALAVÁRI, Zsolt. et al. *“Studierstube” An Environment for Collaboration in Augmented Reality*. Proceedings of Collaborative Virtual Environments, Nottingham, UK, p. 19- 20, Set. 1996. Disponível em: <<http://www.cg.tuwien.ac.at/research/vr/studierstube/jvrs-paper.pdf>>. Acesso em: 29 dez. 2004.

TISSIANI, Gabriela. et al. *Metodologia para criação de ambientes virtuais tridimensionais*. Simpósio Nacional de Geometria Descritiva e Desenho Técnico, 15. Nov, 2001, São Paulo. Disponível em: <<http://www.lrv.ufsc.br/recursos/artigos/MetodologiaAVs.PDF>>. Acesso em: 29 dez. 2004.

TISSIANI, Gabriela. et al. *Design de Interfaces Para Ambientes Virtuais: Como Obter Usabilidade em 3D*. Disponível em: <http://www.lrv.ufsc.br/recursos/artigos/siggradi_design_de_interface.pdf>. Acesso em: 29 dez. 2004.

UMLAUF, Eike J. et al. *ARLib: The Augmented Library*. Proceedings of the First IEEE International Workshop on ARToolKit (ART02). Vienna University of Technology, Austria, 2002.