



UNIVERSIDADE METODISTA DE PIRACICABA
FACULDADE DE CIÊNCIAS EXATAS E DA NATUREZA
MESTRADO EM CIÊNCIA DA COMPUTAÇÃO

**SISTEMA DE AUTORIA EM AMBIENTE COLABORATIVO COM
REALIDADE AUMENTADA**

RAFAEL SANTIN

ORIENTADOR: PROF. DR. CLAUDIO KIRNER

PIRACICABA, SP
2008



UNIVERSIDADE METODISTA DE PIRACICABA
FACULDADE DE CIÊNCIAS EXATAS E DA NATUREZA
MESTRADO EM CIÊNCIA DA COMPUTAÇÃO

**SISTEMA DE AUTORIA EM AMBIENTE COLABORATIVO COM
REALIDADE AUMENTADA**

RAFAEL SANTIN

ORIENTADOR: PROF. DR. CLAUDIO KIRNER

Dissertação apresentada ao Mestrado em Ciência da Computação, da Faculdade de Ciências Exatas e da Natureza, da Universidade Metodista de Piracicaba – UNIMEP, como requisito para obtenção do Título de Mestre em Ciência da Computação.

PIRACICABA, SP
2008

Santin, Rafael

Sistema De Autoria Em Ambiente Colaborativo com
Realidade Aumentada.

Orientador: Prof.Dr. Claudio Kirner

Dissertação (mestrado) - Universidade Metodista de
Piracicaba - Faculdade de Ciências Exatas e da Natureza.

1-Realidade Aumentada. 2-Realidade Virtual. 3-
Autoria. 4-Colaboração. 5-Interação. I. Kirner, Claudio. II.
Universidade Metodista de Piracicaba. Faculdade de
Ciências Exatas e da Natureza. III. Título

SISTEMA DE AUTORIA EM AMBIENTE COLABORATIVO COM REALIDADE AUMENTADA

AUTOR: RAFAEL SANTIN

ORIENTADOR: PROF. DR CLAUDIO KIRNER

Dissertação de Mestrado defendida e aprovada em 28 de
Fevereiro de 2008, pela Banca Examinadora constituída dos Professores:

Prof. Dr. Claudio Kirner (Orientador)
UNIMEP

Prof. Dr. Marcio Merino Fernandes
UNIMEP

Prof. Dr. Alexandre Cardoso
UFU

Aos

Meus pais, Wilson e Maria, pelo conforto e incentivo

À

Minha irmã, Aline, pelo companheirismo

Aos

Meus familiares, pelo apóio

À

Minha namorada, Carla, pela ajuda e compreensão

AGRADECIMENTOS

Ao professor Claudio Kirner, pela extrema competência e por ser um visionário detentor de brilhante sabedoria, que iluminou com primorosas idéias os caminhos desse trabalho. Sou muito grato pela amizade, incentivo, conselhos e o compartilhamento de suas experiências, que foram fundamentais para a minha formação nesse período de pesquisa.

À CAPES, pela bolsa de mestrado.

Aos professores do mestrado da UNIMEP, pelos ensinamentos.

À Rosa, por estar sempre disposta a ajudar.

Aos colegas de curso, pelas preciosas trocas de informações e experiências, além das agradáveis e divertidas conversas.

"Computação não se relaciona mais a computadores.
Relaciona-se a viver".

(Nicholas Negroponte)

RESUMO

Os avanços tecnológicos em ambientes computacionais mostram algumas tendências para recursos altamente personalizáveis, oferecendo a seus usuários ambientes configuráveis, distribuídos e extremamente interativos. Algumas das características dessas tecnologias são encontradas em ambientes de autoria, ambientes colaborativos e ambientes de realidade virtual e aumentada. Os ambientes de autoria oferecem aos usuários recursos computacionais facilmente configuráveis para a elaboração personalizada de seus próprios trabalhos. Já os ambientes colaborativos buscam agrupar diferentes pessoas para contribuírem na elaboração de um trabalho em comum. A realidade aumentada consiste na interface que atua, em tempo real, por meio de dados computacionais altamente interativos, no ambiente físico do usuário. Dessa maneira, o presente trabalho apresenta o desenvolvimento de um sistema de autoria colaborativa de realidade aumentada, que incorpora técnicas de autoria e colaboração à interface de realidade aumentada altamente interativa, oferecendo a seus usuários novas formas de interação para a construção de ambientes virtuais.

PALAVRAS-CHAVE: Realidade Aumentada, Realidade Virtual, Autoria, Colaboração e Interação.

AUTHORING SYSTEM IN COLLABORATIVE ENVIRONMENT WITH AUGMENTED REALITY

ABSTRACT

The technological advances of computers environments show some trends for highly personalized resources, offering configurable, distributed and extremely interactive environment to their users. Some features of these technologies are found in authoring, collaborative, virtual and augmented reality environments. The authoring environment offers easily configurable computer resources for the personalized elaboration of user's work. The collaborative environments allow different people to work together, contributing to build their common tasks. The augmented reality allows the interaction in physic's user environment with computer data in real time. This work presents the development of a collaborative authoring system with augmented reality that combine authoring, collaborative and the highly interactive augmented reality interface techniques, offering new forms of interaction for to the development of virtual reality environments.

KEYWORDS: Augmented Reality, Virtual Reality, Authoring, Collaboration, Interaction.

SUMÁRIO

RESUMO	VII
ABSTRACT	VIII
LISTA DE FIGURAS	XI
LISTA DE TABELAS	XV
1 INTRODUÇÃO.....	1
2 REALIDADE VIRTUAL E AUMENTADA.....	5
2.1 REALIDADE VIRTUAL.....	6
2.1.1 <i>Definição de Realidade Virtual</i>	7
2.1.2 <i>Realidade Virtual e Imersibilidade</i>	8
2.1.3 <i>Dispositivos de Realidade Virtual</i>	9
2.1.3.1 <i>Dispositivos Visuais</i>	9
2.1.3.2 <i>Dispositivos Acústicos</i>	11
2.1.3.3 <i>Dispositivos Táteis</i>	11
2.1.4 <i>Áreas de aplicação da Realidade Virtual</i>	12
2.2 REALIDADE AUMENTADA	13
2.2.1 <i>Imersão em realidade aumentada</i>	14
2.2.2 <i>Aplicações</i>	15
2.2.2.1 <i>Aplicações educacionais</i>	16
2.2.2.2 <i>Visualização de Informações</i>	16
2.2.2.3 <i>Aplicações em Treinamento</i>	17
2.2.3 <i>Sistemas de desenvolvimento de aplicações de realidade Aumentada</i>	18
2.2.3.1 <i>Studierstube</i>	18
2.2.3.2 <i>ARToolKit</i>	19
3 ARTOOLKIT	20
3.1 FUNDAMENTOS DO ARTOOLKIT	20
3.1.1 <i>Marcadores</i>	20
3.1.2 <i>Rastreamento</i>	22
3.1.3 <i>Funcionamento do ARToolKit</i>	24
3.2 INSTALAÇÃO E CONFIGURAÇÃO DO ARTOOLKIT	26
3.2.1 <i>Instalação</i>	26
3.2.2 <i>Configurações na plataforma Windows</i>	27
3.3 APLICAÇÕES INCLUSAS NO ARTOOLKIT	27
3.3.1 <i>Aplicações Utilitárias</i>	27
3.3.2 <i>Exemplo de aplicações de RA</i>	29
3.3.2.1 <i>Teste de Colisão</i>	30
3.3.2.2 <i>Uso da Pá</i>	31
3.3.2.3 <i>SimpleVRML</i>	32
3.4 SOFTWARE BASEADOS NO ARTOOLKIT.....	33
4 COLABORAÇÃO COM REALIDADE AUMENTADA	34
4.1 CONCEITOS DE COLABORAÇÃO.....	34
4.2 COLABORAÇÃO EM AMBIENTES DE REALIDADE AUMENTADA.....	38
4.2.1 <i>Colaboração face a face</i>	39
4.2.2 <i>Colaboração Remota</i>	40
4.3 FERRAMENTAS/EXEMPLOS DE COLABORAÇÃO E AMBIENTES DE REALIDADE AUMENTADA	41
4.4 ASPECTOS DE COMUNICAÇÃO EM REDE.	44
5 SISTEMAS DE AUTORIA COM REALIDADE AUMENTADA	52
5.1 SISTEMA DE AUTORIA DE MUNDOS VIRTUAIS	52

5.1.1 <i>Autoria usando Realidade Virtual</i>	53
5.1.2 <i>Autoria usando Realidade Aumentada</i>	56
5.2 FORMAS DE INTERAÇÃO	59
5.3 CLASSIFICAÇÃO DAS INTERAÇÕES DOS USUÁRIOS EM AMBIENTES DE REALIDADE AUMENTADA.	61
5.3.1 <i>Presença dos objetos</i>	62
5.3.2 <i>Mudanças de características</i>	63
5.3.3 <i>Manipulação</i>	63
5.3.4 <i>Comportamento</i>	64
6 SISTEMA DE AUTORIA COLABORATIVA COM REALIDADE AUMENTADA- SACRA.	65
6.1 CARACTERIZAÇÃO DO AMBIENTE.	65
6.2 DESENVOLVIMENTO.....	67
6.2.1 <i>NetARToolKit</i>	68
6.2.2 <i>Implementação do Sacra</i>	70
6.3 FUNCIONALIDADES DO SACRA.....	76
6.3.1 <i>Cadastramento de pontos</i>	76
6.3.2 <i>Técnicas de interação</i>	78
6.3.3 <i>Funcionalidades dos marcadores de ações</i>	79
6.3.3.1 <i>Inspector</i>	79
6.3.3.2 <i>Control</i>	80
6.3.3.3 <i>Copy</i>	80
6.3.3.4 <i>Transport</i>	80
6.3.3.5 <i>Eraser</i>	81
6.3.3.6 <i>Status</i>	81
6.3.3.7 <i>Path</i>	82
6.3.3.8 <i>Lock</i>	83
6.4 AMBIENTE DE AUTORIA COLABORATIVA.....	83
6.4.1 <i>Referência Local</i>	84
6.4.2 <i>Referência Remota</i>	84
6.5 TESTE DE FUNCIONALIDADE	87
6.5.1 <i>Trabalho Local</i>	87
6.5.2 <i>Trabalho remoto</i>	91
7 CONCLUSÃO	94
REFERÊNCIAS BIBLIOGRÁFICAS.....	97
ANEXOS.....	106
ANEXO 1- COMPILAÇÃO DO ARTOOLKIT NA PLATAFORMA WINDOWS	106
ANEXO 2- PUBLICAÇÕES	109

LISTA DE FIGURAS

FIGURA 2.1. ÓCULOS ESTEREOSCÓPICOS CRYSTALEYES DA REAL D (ABSOLUT, 2007).	10
FIGURA 2.2. HMD I-GLASSES DA I-O DISPLAY SYSTEMS (ABSOLUT, 2007).	10
FIGURA 2.3. LUVAS PINCH GLOVE DA FAKESPACE LABS (ABSOLUT, 2007).	12
FIGURA 2.4. EXEMPLO DE VISÃO INDIRETA.	14
FIGURA 2.5. EXEMPLO DE VISÃO DIRETA.	15
FIGURA 2.6. OBJETOS VIRTUAIS, CARRO E VASO, NO AMBIENTE REAL.	15
FIGURA 2.7. LIVRO INTERATIVO DE REALIDADE AUMENTADA (LIRA)	16
FIGURA 2.8. VISUALIZAÇÃO DE GRÁFICOS COM REALIDADE AUMENTADA.	17
FIGURA 2.9. SISTEMA DE TREINAMENTO PARA A MANUTENÇÃO INDUSTRIAL (SCHWALD, 2003).	17
FIGURA 2.10. FUNCIONAMENTO DO STUDIERSTUBE (SCHMALSTIEG, 2002).	18
FIGURA 3.1. EXEMPLO DE MARCADOR.	21
FIGURA 3.2 UTILIDADE DA MOLDURA BRANCA DO MARCADOR	22
FIGURA 3.3. RELACIONAMENTO ENTRE OS SISTEMAS DE COORDENADAS DO MARCADOR E DA CÂMERA.	23
FIGURA 3.4. MULTIPLICAÇÃO DE MATRIZES	23
FIGURA 3.5 ETAPAS ENVOLVIDAS NO FUNCIONAMENTO DE UMA APLICAÇÃO DE RA : A) IMAGEM DA CENA COM O MARCADOR, B) A IMAGEM LIMARIZADA E C) OBJETO VIRTUAL SOBREPONDO O MARCADOR.	26
FIGURA 3.6. FUNCIONAMENTO DE UMA APLICAÇÃO DE RA USANDO ARTOOLKIT.....	26
FIGURA 3.7. ESTRUTURA DO DIRETÓRIO DA VERSÃO 2.72.1 DO ARTOOLKIT APÓS A INSTALAÇÃO.	27
FIGURA 3.8. INÍCIO DA EXECUÇÃO DO MKPATT	28
FIGURA 3.9. MARCADOR IDENTIFICADO PELO “MKPATT”	29
FIGURA 3.10. EXECUÇÃO DA APLICAÇÃO “COLLIDETEST”: A) CUBOS DESENHADOS SOBRE OS MARCADORES DISTANTES E B) ESFERAS DESENHADAS SOBRE MARCADORES PRÓXIMOS.	30

FIGURA 3.11. EXECUÇÃO DO EXEMPLO “PADDLEDEMO”, MOSTRANDO A ESFERA SOBRE A PÁ, A INCLINAÇÃO PARA DESPEJAR A ESFERA NA BASE E A ESFERA FIXADA NA BASE.....	31
FIGURA 3.12. RESULTADO DA EXECUÇÃO DO <i>SIMPLEVRML</i>	32
FIGURA 3.13. REFERÊNCIA DE MARCADORES E OBJETOS VIRTUAIS VRML	32
FIGURA 3.14. RESULTADO DA EXECUÇÃO DE EXEMPLOS DO OSGART: A) EXEMPLO OSGARTSIMPLENPR E B) EXEMPLO OSGARTVIDEOPLANE.	33
FIGURA 4.1. INTERAÇÃO NUM AMBIENTE COLABORATIVO FACE A FACE (AGRAWALA, 1997).....	37
FIGURA 4.2. EXEMPLO DE AMBIENTE COLABORATIVO REMOTO (GOEBBELS, 2003).	37
FIGURA 4.3. EXEMPLO DE AMBIENTE COLABORATIVO DE RA FACE A FACE (BILLINGHURST, 2002).	39
FIGURA 4.4. VÍDEO-CONFERÊNCIA EM RA (BILLINGHURST, 2002)	40
FIGURA 4.5. EXEMPLO DE COLABORAÇÃO REMOTA, DOIS USUÁRIOS GEOGRAFICAMENTE SEPARADOS ATUANDO NO AMBIENTE COMPARTILHADO (SCHMALSTIEG, 2000).....	41
FIGURA 4.6. MAGICMEETING (REGENBRECHT, 2002).....	42
FIGURA 4.7. EXECUÇÃO DO DWARF (BRUEGGE E KLINKER, 2003).	42
FIGURA 4.8. EXEMPLO DE COLABORAÇÃO FACE A FACE E REMOTA (SCHMALSTEIG, 2000).	43
FIGURA 4.9. PAINEL PESSOAL DE INTERAÇÃO (SCHMALSTEIG, 2000).	44
FIGURA 4.10. MODELO CLIENTE-SERVIDOR.	47
FIGURA 4.11. MODELO PEER-TO-PEER.	48
FIGURA 5.1. EXECUÇÃO DO ALICE.....	54
FIGURA 5.2. INTERFACE DO FLUX STUDIO.....	55
FIGURA 5.4. EXECUÇÃO DO VIZARD.....	55
FIGURA 5.5. EXECUÇÃO DO AMIRE: A) MODO DE EDIÇÃO, B) MODO DE VISUALIZAÇÃO NO AMBIENTE.	56
FIGURA 5.6. DEFINIÇÃO DAS OPERAÇÕES REALIZADAS NO TILES (POUPYREV, 2001).	58

FIGURA 5.7. REDUÇÃO DO VALOR DA ESCALA DO OBJETO VIRTUAL: A) ESCALA 2.0 E B) ESCALA 1.90 (POSTECH, 2007).	59
FIGURA 5.8. EXEMPLOS DE TANGIBLE AR: A) SISTEMAS DE AUTORIA TILES (POUPYREV, 2002) E B) EXECUÇÃO DO MAGIC BOOK (BILLINGHURST, 2001)	61
FIGURA 6.1. INTERAÇÃO ENTRE AS CAMADAS DO SISTEMA.....	68
FIGURA 6.2. ESTRUTURA DE FUNCIONAMENTO DO AMBIENTE.....	69
FIGURA 6.3. ESTRUTURA DO SACRA.....	70
FIGURA 6.4. DEFINIÇÃO DAS FUNCIONALIDADES DOS MARCADORES.....	72
FIGURA 6.5. ESTRUTURA PRINCIPAL DA AUTORIA	73
FIGURA 6.6. ESTRUTURA DO MARCADOR REF NO SACRA.	74
FIGURA 6.7. RELACIONAMENTO ENTRE AS ESTRUTURAS DE DADOS NO SACRA.	75
FIGURA 6.8. ARQUIVO DE ASSOCIAÇÃO DE OBJETOS VIRTUAIS AO MARCADOR.....	76
FIGURA 6.9. EXEMPLO DOS MARCADORES DE AÇÕES E REFERÊNCIA.	78
FIGURA 6.10. DISTÂNCIA PARA A ATUAÇÃO DO MARCADOR DE AÇÃO.	79
FIGURA 6.11. FUNÇÃO UTILIZADA PARA A MODIFICAÇÃO DA STRING NO NÓ TEXT DO PAINEL DE STATUS.	82
FIGURA 6.12. ESQUEMATIZAÇÃO DO USO DA REFERÊNCIA REMOTA PARA COLABORAÇÃO SÍNCRONA.....	85
FIGURA 6.13. EXEMPLO DO ARQUIVO DE HOSTS DO SACRA.	85
FIGURA 6.14. USO DO RASTRO PARA INDICAR A TRAJETÓRIA.	86
FIGURA 6.15. USO DO MARCADOR DE INSPEÇÃO: A) APROXIMAÇÃO DA ESFERA DO MARCADOR DE INSPEÇÃO DA ESFERA REPRESENTANTE DO PONTO, B) ATIVAÇÃO DA VISUALIZAÇÃO DO OBJETO E C) VISUALIZAÇÃO DOS OBJETOS VIRTUAIS REPRESENTADOS PELOS PONTOS.	88
FIGURA 6.16. USO DO MARCADOR DE CONTROLE PARA A TROCA DE OBJETOS VIRTUAIS: A) TELEVISÃO E B) TAPETE	89
FIGURA 6.17. SELEÇÃO DA REFERÊNCIA DE DESTINO: A) APROXIMAÇÃO DA ESFERA DO MARCADOR DE CÓPIA DA REFERÊNCIA E B) REFERÊNCIA SELECIONADA.	89

FIGURA 6.18. CÓPIA DE OBJETO VIRTUAL: A) APROXIMAÇÃO DA ESFERA DO MARCADOR DE CÓPIA DO OBJETO, B) OBJETO COPIADO E C) FIXAÇÃO DO OBJETO COPIADO NO MARCADOR DE DESTINO.	90
FIGURA 6.19. TRANSPORTE DO OBJETO VIRTUAL: A) APROXIMAÇÃO DA ESFERA DO MARCADOR DE TRANSPORTE DO OBJETO A SER COPIADO E B) OBJETO REPOSICIONADO E REORIENTADO ATRAVÉS DO MARCADOR DE TRANSPORTE.	90
FIGURA 6.20. REMOÇÃO DO OBJETO VIRTUAL: A) APROXIMAÇÃO DO MARCADOR DE REMOÇÃO DO OBJETO VIRTUAL A SER REMOVIDO E B) OBJETO VIRTUAL REMOVIDO.	91
FIGURA 6.21. MONTAGEM DE UMA SALA VIRTUAL.....	91
FIGURA 6.22. VISUALIZAÇÃO DA REFERÊNCIA REMOTA COM AS PEÇAS DO HELICÓPTERO: A) USUÁRIO1 E B) USUÁRIO2.....	92
FIGURA 6.23. TRAJETÓRIA INDICADA PELA SEQÜÊNCIA DE ESFERAS: A) TRAJETÓRIA VISUALIZADA PELO USUÁRIO1 E B) TRAJETÓRIA VISUALIZADA PELO USUÁRIO2.....	92
FIGURA 6.24. MOVIMENTAÇÃO DA PEÇA DO HELICÓPTERO PELO USUÁRIO2: A) VISUALIZAÇÃO DO MOVIMENTO PELO USUÁRIO1 E B) PONTO DE VISTA DO USUÁRIO2.	93
FIGURA 6.25. MONTAGEM COMPLETA DO HELICÓPTERO.....	93

LISTA DE TABELAS

TABELA 3.1 PASSOS E AS FUNÇÕES IMPLEMENTADAS NUM EXEMPLO DE APLICAÇÃO DISTRIBUÍDA COM O ARTTOOLKIT	25
TABELA 4.1. CLASSIFICAÇÃO DA INTERAÇÃO DO GROUPWARE EM RELAÇÃO AO TEMPO E ESPAÇO.....	35
TABELA 6.1. COMANDOS IMPLEMENTADOS.....	69
TABELA 6.2. POSSÍVEIS COMANDOS ENVIADOS AOS USUÁRIOS REMOTOS.....	86

1 INTRODUÇÃO

As interfaces computacionais vêm ao longo do tempo se aprimorando de forma a facilitar o uso do computador. Assim, tornou-se possível aumentar a quantidade de usuários, visto que, interfaces mais interativas necessitam de treinamentos menos rigorosos para sua manipulação. Dessa maneira, o uso de certos sistemas, ou até mesmo do computador, passou a deixar de ser restrito às pessoas especializadas, possibilitando estender a usabilidade computacional ao cotidiano dos cidadãos.

A popularidade dos computadores pode ser notada através do crescente número de trabalhos que abarcaram o computador como ferramenta principal para o seu desenvolvimento. Dessa forma, foi promovido um forte laço de dependência do trabalho com os sistemas computacionais, que passaram a ser um dos principais meios de produção da sociedade atual. Isso é tão verídico na atualidade, que problemas em computadores podem provocar a ociosidade do trabalhador, por não ter o meio para realizar grande parte de suas tarefas.

Por esse motivo, o desenvolvimento de ambientes computacionais, com interfaces mais intuitivas e com alto grau de interação, possibilita que os usuários exerçam suas atividades sem dificuldade, beneficiando, assim, o seu trabalho. Para isso, existe uma área multidisciplinar de estudo da relação homem-computador, denominada Interação Humano-Computador (IHC). Essa área procura estabelecer critérios de ajustes aos ambientes computacionais, a fim de proporcionar aos usuários melhores formas de interação com o computador. O termo interação humano-computador refere-se à forma com que o usuário se comunica com o computador (SILVA FILHO, 2003). Essa comunicação é, comumente, realizada através de dispositivos de entrada padrões, como o mouse e o teclado.

Embora a maioria dos usuários tenha se adaptado ao uso desses dispositivos, existem os que encontram dificuldades para manipulá-los, como no caso de boa parte de usuários idosos (XAVIER, 2004). Isso acontece

devido ao fato dos dispositivos, normalmente utilizados, não corresponderem com as ações intuitivas do usuário, fazendo a necessidade de um treinamento para a aprendizagem da operação desses dispositivos. Assim, para facilitar o uso do computador, torna-se necessário buscar maneiras mais próximas da sua realidade, ou seja, formas mais naturais de interação humano-computador.

Consoante a essa tendência para as futuras interfaces de ambientes computacionais, está a realidade aumentada (RA), por permitir uma interação humano-computador mais intuitiva, ou seja, permitir que o usuário manipule o computador de forma mais amigável e natural (KIRNER, 2004a). Por exemplo, as operações tangíveis de RA possibilitam que o usuário manipule objetos virtuais com as próprias mãos, ou por intermédio de uma placa, de forma semelhante a interação com objetos reais.

A interface tangível de RA estimula a percepção do usuário, através da interação com elementos computacionais visíveis no seu próprio ambiente. Essa interação visual é marcada pela associação de eventos computacionais a símbolos gráficos, facilmente acessíveis aos usuários.

O sentido visual é tão importante para viabilizar operações mais intuitivas no computador, que NAJORK (1994) desenvolveu o Cube uma linguagem de programação visual em 3D. Nesse contexto, abstrações em níveis mais elevados são empregadas em ferramentas de autoria, para permitir que usuário, sem o conhecimento específico de programação, possa desenvolver as suas próprias aplicações. Assim, as ferramentas de autoria através de determinados recursos computacionais visuais possibilitam facilitar e agilizar o trabalho dos usuários.

O desenvolvimento de técnicas computacionais para auxiliar e aprimorar o trabalho das pessoas não é objetivo, apenas, das ferramentas de autoria, mas também de outros tipos de sistemas computacionais, como os ambientes colaborativos.

Os ambientes colaborativos face a face ou remotos viabilizam o trabalho em grupo, através da disposição de formas de cooperação,

coordenação e comunicação entre os seus participantes. Dessa maneira, a troca de experiência entre os usuários do grupo permite contribuir para o desenvolvimento de trabalhos em comum.

O trabalho compartilhado por vários usuários (típicos dos ambientes de autoria) facilita e agiliza o desenvolvimento de aplicações por não programadores e os benefícios das interfaces de RA são conceitos interessantes para serem reunidos e oferecidos em uma única ferramenta de autoria.

Assim, esse trabalho apresenta o desenvolvimento de um sistema de autoria colaborativo com Realidade Aumentada, que possibilita unir o trabalho colaborativo com técnicas de autoria em Realidade Aumentada para a construção de ambientes virtuais.

O objetivo do sistema de autoria colaborativa proposto é fornecer a seus usuários a possibilidade de desenvolverem, por meio de recursos altamente interativos de Realidade aumentada, as suas próprias aplicações, favorecendo o trabalho de não programadores. A utilização desses novos recursos de interação, tanto no ambiente local, quanto no remoto, permitem a colaboração entre diferentes usuários, que visualizam e participam na produção de uma determinada aplicação.

O presente trabalho está dividido em sete capítulos. O primeiro capítulo apresenta a introdução do assunto do trabalho. O segundo capítulo aborda os conceitos e tecnologias envolvidas nas interfaces de realidade virtual e realidade aumentada. O terceiro capítulo apresenta uma descrição detalhada das principais características do ARToolKit, mostrando aspectos relacionados ao funcionamento, a configuração e a utilização dos exemplos disponíveis. O quarto capítulo descreve os conceitos de ambientes colaborativos, focando os ambientes colaborativos de RA, os sistemas disponíveis e os conceitos de sistemas distribuídos envolvidos para a construção do sistema de colaboração remoto. O quinto capítulo aborda os sistemas de autoria em realidade virtual e realidade aumentada, descrevendo algumas ferramentas existentes e as formas de interação existentes em realidade aumentada. O sexto capítulo

descreve a implementação do sistema de autoria em ambiente colaborativo de realidade aumentada e relata um teste de funcionalidade do sistema desenvolvido. Por fim, o sétimo capítulo apresenta a conclusão e aponta novos desdobramentos para trabalhos futuros.

2 REALIDADE VIRTUAL E AUMENTADA

O ser humano, segundo PINKER (1998), é capaz de formalizar o seu pensamento através da linguagem. A linguagem é descrita por ARANHA (1993) como um sistema de signos capazes de substituir, através da representação, objetos existentes no mundo natural. Dessa maneira, a necessidade do homem em expressar seus pensamentos promoveu a criação de diversos tipos de linguagens, como as linguagens verbais, matemáticas, de computador, artísticas (musical, teatral, cinematográfica etc.), gestuais, espaciais e etc.

TORI e KIRNER (2006), afirmam que representações da realidade ou da imaginação permitiram expressar a cultura de uma sociedade ao longo do tempo, começando pelos desenhos primitivos, símbolos, figuras, pinturas, passando pelos jogos e teatros até o cinema. Os sistemas computacionais convergiram e aprimoraram essas formas de expressões, por meio do desenvolvimento de interfaces gráficas, que podem ser enriquecidas com um conjunto de mídia (texto, imagem, áudio, vídeo e animação), gráficos 3D e interação multisensorial com o usuário (SUTCLIFFE, 2002).

A realidade virtual representa esse tipo de interface, através da geração em tempo real de ambientes tridimensionais interativos. Os ambientes de realidade virtual superam as limitações inerentes à interação das interfaces 2D e oferece aos usuários a interação intuitiva no espaço 3D. O uso de alguns dispositivos tecnológicos de realidade virtual promove o dinamismo no estímulo sensorial do usuário, possibilitando-o atuar na aplicação e receber o *feedback* de suas ações.

O usuário da interface de realidade virtual pode estar imerso num ambiente virtual totalmente imaginário, como na cabeça de uma barata voadora sobrevoando o “planeta” Krypton, ou explorar um cenário modelado sinteticamente fiel ao real, como a cabine de comando de uma aeronave, laboratórios de universidades, museus, estádios de futebol entre outros. Assim, a abrangência dessa interface estende a um grande número de aplicações,

como na educação, medicina, entretenimento etc. Um exemplo de uso da interface de realidade virtual está no mundo virtual on-line, que simula certos aspectos da vida real, conhecido como Second Life. Com mais de 5 milhões de residentes, o Second Life mostra a potencialidade da realidade virtual em possibilitar uma forma de interação social virtual entre as pessoas, que são representadas na simulação por avatares (Second Life, 2007).

As próximas seções abordarão os conceitos da realidade virtual e de sua vertente a realidade aumentada.

2.1 REALIDADE VIRTUAL

A Realidade Virtual (RV) tem se tornado cada vez mais popular na atualidade, devido à queda dos custos de seus equipamentos (CLIBURN, 2004). Desde a sua criação por volta década de 50, até meados da década de 80, a RV era restrita a algumas instituições altamente equipadas, visto os proibitivos custos dos hardwares utilizados para a sua execução. No final da década de 80, a RV passou a ser viável às empresas e universidades, devido aos avanços na arquitetura dos PCs e ao aumento na demanda de dispositivos de RV, oferecendo, no mercado, sistemas capazes de desenvolver e executar aplicações de realidade virtual a custos mais acessíveis (PINHO, 1997).

A popularização da RV ressaltou o uso dessa interface em diversos projetos de pesquisa, o que possibilitou sua propagação a diversas áreas do conhecimento. Devido à sua vasta possibilidade de aplicações, a RV é utilizada no auxílio à venda de produtos, no tratamento de alguns distúrbios psicológicos, no benefício ao aprendizado, seja de funcionários de alguma empresa, alunos das instituições de ensino, ou, até mesmo, de pessoas deficientes, etc. (PINHO, 1997; MACREDIE, 1996; CLIBURN, 2004).

Assim, a realidade virtual demonstra ser versátil, para atender as mais variadas necessidades exigidas na interação homem-máquina, abrindo um canal sensorial para expressão do usuário e o computador.

2.1.1 DEFINIÇÃO DE REALIDADE VIRTUAL

No final do século XIX, os cientistas acreditavam na existência de um meio hipotético que preenchia todo o espaço, denominado de 'éter'. Esse 'éter' seria o meio necessário para a propagação dos raios eletromagnéticos (HAWKING, 2002). Um conceito, de certa forma, similar ao 'éter', foi utilizado por SHERMAN (2003) para definir o *medium*, que corresponde ao meio responsável pela comunicação. Consideram a realidade virtual como sendo o *medium*, ou seja, o meio, ou ponte que propaga a comunicação de idéias. Nesse sentido, segundo a sua definição, a realidade virtual consiste num *medium* composto por simulações computacionais interativas, que detectam as posições e ações dos usuários, a fim de substituir ou potencializar a resposta a um ou mais sentidos, fornecendo a sensação de se estar mentalmente imerso ou presente na simulação de um mundo virtual.

De acordo com TORI (2006): *“A realidade virtual é uma interface avançada para aplicações computacionais, que permite ao usuário a movimentação (navegação) e interação em tempo real, em um ambiente tridimensional, podendo fazer uso de dispositivos multisensoriais, para atuação ou feedback”*.

VINCE (2004) define a realidade virtual, como uma forma de navegação e manipulação de ambientes 3D elaborados por computador. O usuário pode navegar andando, correndo, ou até mesmo sobrevoando o ambiente virtual e explorar pontos de vistas que seriam impossíveis no mundo real. Mas, o benefício real da RV consiste na habilidade de tocar, animar, pegar e reposicionar o objeto virtual, além de criar novas configurações.

A partir dessas definições, pode-se dizer que a realidade virtual consiste em uma interface avançada de computador, que permite a navegação, a interação e a imersão em ambientes tridimensionais, em tempo real e de forma dinâmica ao envolver os canais multisensoriais do usuário.

Os canais sensoriais são os meios que possibilitam ao homem captar dados do ambiente para que sejam interpretados pelo cérebro, a fim de

orientar as suas ações frente a um determinado estímulo. De maneira análoga a esse comportamento humano, encontra-se o funcionamento básico de computadores, os quais recebem as informações dos dispositivos convencionais de entrada, como o mouse e o teclado, as processam e exibem os resultados ao usuário pelos sistemas de saídas, como monitores e impressoras (NETTO, 2002).

Existem dispositivos não convencionais, para sistemas de realidade virtual, capazes de lograr os sentidos dos usuários, pois tais dispositivos, por exemplo, possibilitam que dados complexos do movimento do usuário sejam captados pelo computador, processados e expostos pelos dispositivos de saída de maneira altamente realista. Os HMDs e rastreadores são exemplos desses dispositivos, que oferecem ao usuário a sensação de naturalidade na manipulação do ambiente tridimensional, proporcionando a interação de maneira intuitiva (PINHO, 1997).

A sensação de o usuário estar situado no interior do ambiente é denominada de imersão. A imersão depende de determinados tipos de dispositivos para estimular os sentidos do corpo (SHERMAN, 2003).

2.1.2 REALIDADE VIRTUAL E IMERSIBILIDADE

Na realidade virtual os dispositivos utilizados para a interação caracterizam o ambiente em imersivo ou não imersivo. O ambiente é denominado imersivo, quando o usuário tem a sensação de estar dentro do ambiente. A sensação de imersão é propiciada por dispositivos que envolvem os sentidos do usuário, como a visão, audição e tato. Embora a audição e tato auxiliem na imersão, é a visão o sentido crucial a ser estimulado, de modo que o usuário sinta-se dentro do ambiente virtual. Alguns exemplos de dispositivos que viabilizam a imersão são HMDs, CAVES, rastreadores de posição, luvas, dispositivos sonoros espaciais, etc,

A utilização de dispositivos convencionais, para interagir no ambiente, caracteriza a RV como não imersiva. A visualização do ambiente virtual através do monitor é uma forma não imersiva. Nesse caso, mesmo

utilizando óculos estereoscópicos para ressaltar os elementos tridimensionais, o usuário não desfrutará de uma ambiente imersivo, devido à facilidade de “sair” do ambiente virtual, por exemplo, com um desviar de olhos (KIRNER, 2004b).

2.1.3 DISPOSITIVOS DE REALIDADE VIRTUAL.

Existem vários tipos de dispositivos de realidade virtual. Alguns exemplos de dispositivos visuais, acústicos e táteis serão mostrados nas próximas seções.

2.1.3.1 DISPOSITIVOS VISUAIS.

A visão é o principal sentido humano. Portanto, os dispositivos que operam sobre esse sentido podem influenciar significativamente na sensação de imersividade (NETTO, 2002). Dentre alguns dos dispositivos visuais estão: os óculos estereoscópicos, *head mounted displays* (HMDs) e sistemas baseados em projeções.

A estereoscopia, visão binocular ou visão em profundidade resulta da captação de imagens por cada olho, com um pequeno diferencial angular (STEREO VISION, 2007). Nos óculos estereoscópicos, Figura 2.1, as imagens da direita e esquerda são ligeiramente alternadas na tela do monitor. Quando o observador olha para a tela com os óculos, cada lente de cristal líquido está sincronizada para omitir as imagens quando necessário, para que cada olho veja apenas suas respectivas perspectivas de visão. O olho esquerdo vê apenas o campo de visão esquerdo e o olho esquerdo o campo de visão direito. Esse dispositivo é útil em aplicações como visualização científica ou cirurgias nas quais várias pessoas precisam observar a imagem estereoscópica (ENGDAHL, 1999).



FIGURA 2.1. ÓCULOS ESTEREOSCÓPICOS CRYSTALEYES DA REAL D (ABSOLUT, 2007).

O capacete de visualização, conhecido como HMD (*Head Mounted Display*), Figura 2.2, é constituído de duas pequenas telas de TV, ou seja, possui dois dispositivos de visualização e lentes especiais onde são exibidas as imagens virtuais. É adaptável à cabeça como um capacete, mas desconfortável, não permitindo total liberdade de movimentos. Pode oferecer ao usuário a sensação de imersão no mundo virtual, além de poder produzir tridimensionalmente imagens realísticas. O usuário poderá sentir que está se locomovendo, a partir do uso de sensores de rastreamento dos movimentos da cabeça. Com dados do rastreamento, o computador consegue gerar os objetos virtuais na posição da área de observação do usuário. Os HMDs são construídos, normalmente, usando um dentre dois tipos de monitores: os CRTs monitores de TV ou os monitores de cristal líquido, os LCDs. Os CRTs possuem alta resolução, mas desconforto ergonômico. Já os LCDs são leves, mas não tem boa resolução, pois esta tecnologia é deficiente para monitores pequenos (ENGDAHL, 1999).



FIGURA 2.2. HMD I-GLASSES DA I-O DISPLAY SYSTEMS (ABSOLUT, 2007).

Existem vários tipos de dispositivos baseados na projeção. A *cave*, a *virtual table* e tela panorâmica são alguns exemplos desses dispositivos que permitem a imersão parcial ou total em ambientes virtuais.

A *cave* consiste em um ambiente que permite imersão em tamanho real e visualização em grupo. Os usuários são envolvidos por imagens estereoscópicas, projetadas no chão, nas paredes e no teto. Dessa maneira, um usuário responsável pela navegação utiliza rastreadores e óculos estereoscópicos, enquanto os outros usam apenas os óculos estereoscópicos (KIRNER, 2004).

A *virtual table* consiste num dispositivo que projeta imagens estereoscópica numa base horizontal ou inclinada. Os usuários desse sistema necessitam utilizar óculos estereoscópicos para visualizar e manipular os objetos em três dimensões.

A tela panorâmica refere-se a uma tela com uma curvatura que pode atingir 180 graus, onde são projetadas cenas seqüenciais, através de vários projetores alimentados por supercomputadores. Assim, torna-se possível à navegação panorâmica nos ambientes virtuais.

2.1.3.2 DISPOSITIVOS ACÚSTICOS.

A audição permite, através do som emitido por um objeto, localizá-lo no espaço. Com isso, a inserção de som em ambientes virtuais possibilita aumentar o seu grau de realismo. Esse som pode ser emitido por alto falantes ou fones de ouvido, constituindo ou não o capacete de visualização.

2.1.3.3 DISPOSITIVOS TÁTEIS.

As sensações de tato no ambiente virtual são registradas por sensores, que detectam a posição do usuário em relação ao objeto virtual e enviam informações para dispositivos que podem travar os movimentos do

usuário, emitir vibrações ou podem estar conectados a dispositivos geradores de som que emitem sons para indicar colisão.

Um exemplo desse dispositivo é a luva, Figura 2.3, que permite manipular objetos virtuais de maneira bastante intuitiva, dada a possibilidade de o usuário usar os gestos para empurrar, puxar, girar ou agarrar esses objetos.



FIGURA 2.3. LUVAS PINCH GLOVE DA FAKESPACE LABS (ABSOLUT, 2007).

2.1.4 ÁREAS DE APLICAÇÃO DA REALIDADE VIRTUAL

A realidade virtual tem sido aplicada em diversas áreas de conhecimento. Essas aplicações colocam o ser humano frente a abstrações da sua realidade, através da interpretação de dados complexos, articulando os seus sentidos. Portanto, é possível cada vez mais a integração homem-máquina, em virtude dos avanços no desenvolvimento de computadores e dispositivos.

Existem diversas aplicações de realidade virtual, envolvendo as mais variadas áreas. Por exemplo, aplicações: industriais, médicas, arquitetônicas, científicas, artísticas, educacionais entretenimento e etc.

2.2 REALIDADE AUMENTADA

Segundo SHERMAN (2003), a Realidade Aumentada (RA) consiste num tipo de realidade virtual, na qual estímulos sintéticos são realizados sobre objetos do mundo real. Geralmente, a RA é utilizada para fazer com que a informação imperceptível seja perceptível ao sentido humano.

AZUMA (1997) afirma que a realidade aumentada é uma variação da realidade virtual. Enquanto que, na realidade virtual, os usuários são imersos em um ambiente totalmente sintético, a realidade aumentada, por outro lado, permite que o usuário veja objetos virtuais sobrepondo o mundo real. O ideal, em realidade aumentada, seria que os objetos virtuais aparecessem ao usuário coexistindo no mesmo espaço.

De acordo com INSLEY (2003) a realidade aumentada é uma melhoria do mundo real por meio de um mundo virtual criado por computador. O mundo virtual pode variar na complexidade, desde objetos simples como textos, sons e imagens, até objetos virtuais complexos com aparência de objetos reais.

KIRNER e KIRNER (2007) afirmam que “realidade aumentada é a inserção de objetos virtuais no ambiente físico, mostrada ao usuário, em tempo real, com o apoio de algum dispositivo tecnológico, usando a interface do ambiente real, adaptada para visualizar e manipular os objetos reais e virtuais”.

Uma definição considerando as definições apresentadas é:

Realidade aumentada é uma interface avançada de computador, que promove em tempo real a exibição de elementos virtuais sobre a visualização de determinadas cenas do mundo real, oferecendo um forte potencial a aplicações, devido ao alto grau de interatividade.

Os elementos virtuais, em ambientes de realidade aumentada (RA), contribuem para a expansão do ambiente real, através da adição de informações computacionais. Dessa forma, a imersão na visualização do

mundo real conjugado com objetos computacionais virtuais, depende da utilização de alguns dispositivos tecnológicos.

2.2.1 IMERSÃO EM REALIDADE AUMENTADA

O uso de dispositivos para a visualização em realidade aumentada pode oferecer ou não a possibilidade do sentido de imersão ao usuário. Os dispositivos ópticos que possibilitam a imersão são classificados de dispositivos de visão direta, já os dispositivos de visão indireta não exercem a sensação de imersão. A diferença de imersão entre as duas formas de visão está relacionada à autenticidade impelida ao sentido visual do usuário.

Os dispositivos de visão indireta, Figura 2.4, como os monitores não imersivos, promovem a visualização conjugada das cenas virtual e real fora do espaço alvo de atuação.

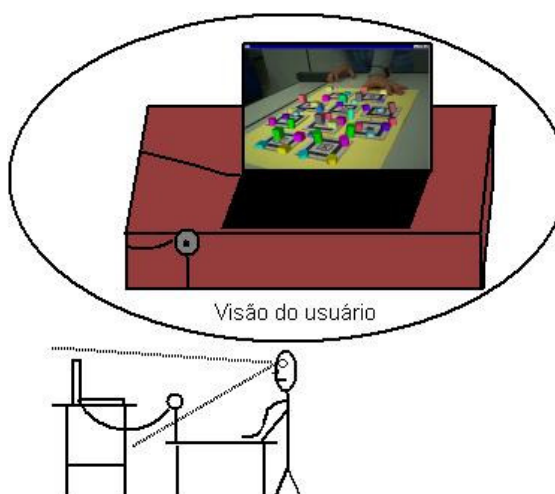


FIGURA 2.4. EXEMPLO DE VISÃO INDIRETA.

Já a visualização direta, imersiva, Figura 2.5, pode ser obtida por meio dos dispositivos como os HMDs, que fornecem a visualização combinada das cenas diretamente no espaço alvo de atuação da aplicação (KIRNER, 2006).

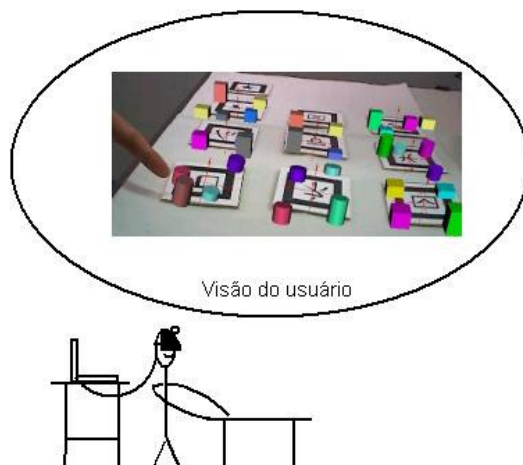


FIGURA 2.5. EXEMPLO DE VISÃO DIRETA.

Mesmo possuindo a variação na imersão, ambos os dispositivos (visão direta e indireta) possibilitam não só visualizar objetos virtuais junto ao mundo real, de maneira realista, conforme Figura 2.6, mas também, interagir intuitivamente, incrementando a percepção do usuário (KIRNER; TORI, 2006).



FIGURA 2.6. OBJETOS VIRTUAIS, CARRO E VASO, NO AMBIENTE REAL.

2.2.2 APLICAÇÕES

A aplicabilidade da interface de RA abrange diversas áreas do conhecimento. Assim, existem diversas aplicações que empregam a RA para incrementar suas funcionalidades. Nas próximas seções, serão descritos três projetos em diferentes áreas, que utilizam essa interface.

2.2.2.1 APLICAÇÕES EDUCACIONAIS

A realidade aumentada pode ser utilizada para auxiliar o ensino. Um exemplo é o projeto LIRA (Livro Interativo de Realidade Aumentada), que busca potencializar o uso do livro com o complemento do computador. Uma das aplicações desse projeto é facilitar o estudo de ciências. A Figura 2.7 mostra a execução do LIRA, durante a visualização de sólidos geométricos (KIRNER, 2004a).

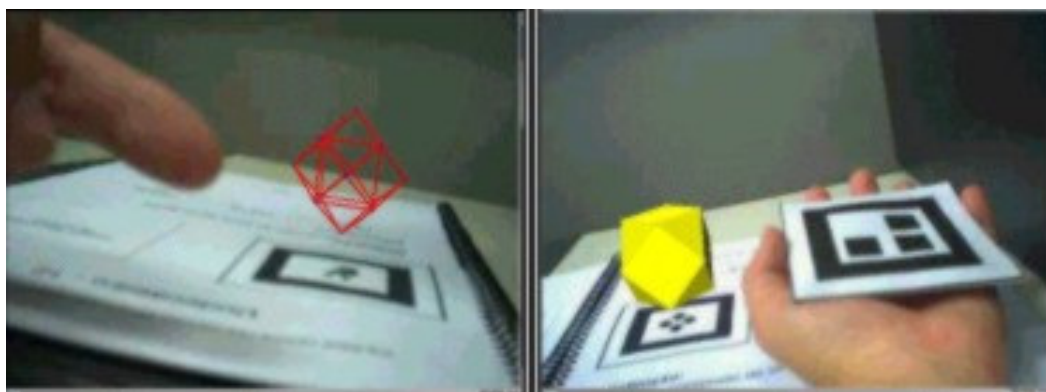


FIGURA 2.7. LIVRO INTERATIVO DE REALIDADE AUMENTADA (LIRA)

2.2.2.2 VISUALIZAÇÃO DE INFORMAÇÕES

A RA na área informacional trabalha com a visualização e gerenciamento da informação, possibilitando facilitar as tomadas de decisões em determinadas situações. Nesse caso, é realizado o realce das características intrínsecas nos elementos reais, durante a sua visualização. Isso é obtido por intermédio de informações que simplificam as formas de entendimento das entidades em questão, ou seja, dos objetos visualizados. Nessa linha está a ferramenta DATAVIS-AR, que objetiva visualizar dados de maneira clara e intuitiva, por meio de atributos gráficos, para os usuários poderem exercer ações, como a de controle. A Figura 2.8 mostra a execução desse exemplo (KIRNER, 2004a).

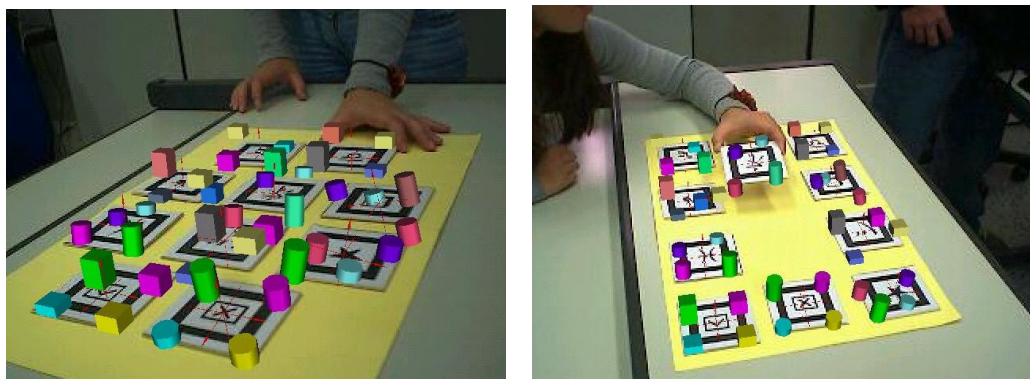


FIGURA 2.8. VISUALIZAÇÃO DE GRÁFICOS COM REALIDADE AUMENTADA.

2.2.2.3 APLICAÇÕES EM TREINAMENTO

O treinamento é uma das grandes possibilidades de aplicações em RA, visto a redução de riscos durante a operação no mundo real e a eficiência do aprendizado com simulações realísticas. A Figura 2.9 demonstra um sistema de que auxilia a manutenção de equipamentos industriais. Nesse sistema, o trabalho do usuário é orientado pelo computador através de textos e setas [SCHWALD, 2003].



FIGURA 2.9. SISTEMA DE TREINAMENTO PARA A MANUTENÇÃO INDUSTRIAL (SCHWALD, 2003).

2.2.3 SISTEMAS DE DESENVOLVIMENTO DE APLICAÇÕES DE REALIDADE AUMENTADA.

Existem diversos sistemas para o desenvolvimento de aplicações de realidade aumentada. Nas seções a seguir, serão apresentados dois exemplos desses sistemas, o Studierstube e o ARToolKit.

2.2.3.1 STUDIERSTUBE

O sistema Studierstube foi desenvolvido pela Universidade de Tecnologia de Viena, a fim de viabilizar aos usuários uma interface de aplicações que permitisse a manipulação de dados complexos em três dimensões. O principal objetivo de seus desenvolvedores é fornecer uma metáfora para interface de usuário 3D tão poderosa quanto a metáfora do desktop para 2D (SCHMALSTIEG, 2002).

No núcleo do sistema do Studierstube, a realidade aumentada colaborativa é utilizada para embutir a imagem gerada por computador no ambiente do mundo real. Esse sistema possibilita tanto a colaboração face a face, quanto remota, entre os usuários, além de lhes oferecer diferentes pontos de vista. A Figura 2.10 mostra dois usuários trabalhando face a face no Studierstube.

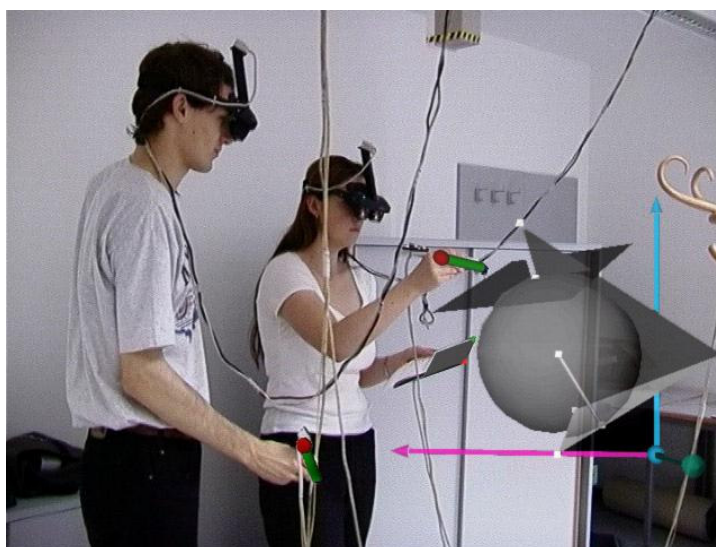


FIGURA 2.10. FUNCIONAMENTO DO STUDIERSTUBE (SCHMALSTIEG, 2002).

2.2.3.2 ARToolKIT

O ARToolKit (ARToolKit, 2007) é uma biblioteca de desenvolvimento de aplicações de Realidade Aumentada, bastante popular na comunidade de RA. Isto acontece pelo fato da biblioteca fornecer soluções de rastreamento 3D, em tempo real, com baixo custo computacional (LEPETIT, 2005). Além disso, o ARToolKit é amplamente utilizado por ser distribuído livremente para fins não comerciais, incentivando a liberdade para os usuários executarem, estudarem e modificarem os códigos disponíveis na biblioteca de acordo com as suas necessidades

O rastreamento óptico oferecido pelo ARToolkit possibilita extrair de forma rápida a posição e orientação de padrões marcadores, apenas com o uso de um computador e uma web câmera simples. Nesse caso, a interação não sofre restrições de cabos utilizados nos diversos tipos de dispositivos de rastreamento (HORNECKER, 2005). Por esses motivos e o baixo custo do hardware necessário para a sua implementação, o ARToolKit foi utilizado para o desenvolvimento desse trabalho. O capítulo 3 apresentará detalhes de funcionamento e configurações necessárias para a compilação, além de mostrar alguns exemplos contidos nas distribuições, que podem auxiliar os programadores no desenvolvimento de suas próprias aplicações.

3 ARTOOLKIT

O ARToolKit é uma biblioteca de programação multi-plataforma, considerada um kit de ferramenta de RA, sendo bastante utilizada e discutida por desenvolvedores e pesquisadores da comunidade de Realidade Aumentada. O ARToolKit possui o seu código livre para modificações e uso no desenvolvimento de aplicações não comerciais sob licença GPL (GNU, 2007), enquanto que a versão proprietária para a comercialização é oferecida pela incorporação ARToolworks (ARToolworks 2007).

3.1 FUNDAMENTOS DO ARTOOLKIT

A biblioteca ARToolKit implementada em C e C++ oferece suporte a programadores para o desenvolvimento de aplicações de RA. Essa biblioteca utiliza o rastreamento óptico, que implementa técnicas de visão computacional para identificar e estimar em tempo real a posição e a orientação de um marcador (moldura quadrada desenhada em papel) em relação ao dispositivo de captura de vídeo. Assim, o cálculo da correlação entre os dados estimados do marcador real e a sua imagem, possibilita posicionar objetos virtuais alinhados à imagem do marcador (KATO, 2000).

A saída de aplicações desenvolvidas com ARToolKit pode ser visualizada através de dispositivos de visão indireta, não imersivos, ou visão direta, imersivos.

Os objetos virtuais visualizados em aplicações desenvolvidas com o ARToolKit podem ser modelados utilizando OPENGL, VRML e X3D. A visualização dos objetos virtuais é realizada no momento da inserção de seus respectivos marcadores no campo de captura da câmera de vídeo.

3.1.1 MARCADORES

O rastreamento implementado no ARToolKit estima a posição de marcadores, tornando possível desenvolver aplicações que necessitem conhecer a posição e orientação de elementos ou ações reais, que são representados na cena por marcadores. Por exemplo, as aplicações de RA,

que utilizam o marcador para posicionar e orientar elementos virtuais na cena do mundo real, tornando um meio de interação do usuário com essas aplicações.

Os marcadores reconhecidos pelo ARToolKit consistem em figuras geométricas quadradas, que contém, no seu interior, símbolos para identificá-los. A Figura 3.1 mostra um exemplo de marcador com símbolos para a sua identificação.

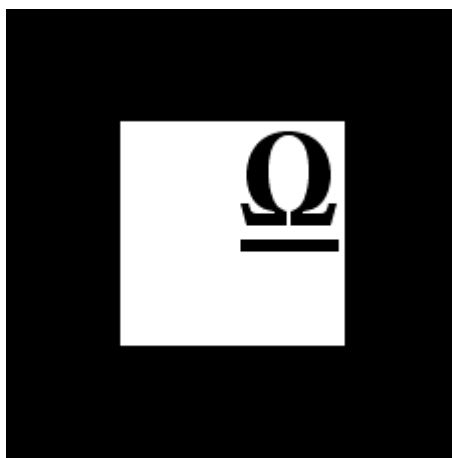


FIGURA 3.1. EXEMPLO DE MARCADOR.

Como o ARToolKit extrai, da imagem de vídeo limiarizada (em preto e branco), as bordas do quadrado em preto, utiliza-se uma moldura em branco envolvendo esse quadrado para promover o contraste no próprio marcador, viabilizando o seu reconhecimento sobre superfícies de cores escuras. A Figura 3.2a demonstra dois marcadores dispostos sobre uma superfície escura. A diferença entre os marcadores consiste no fato do marcador com o símbolo RA não possuir a moldura em branco e outro conter essa moldura. O marcador com o símbolo RA não pode ser identificado sobre a superfície escura, pois não é possível extrair as suas bordas na imagem limiarizada (Figura 3.2b).

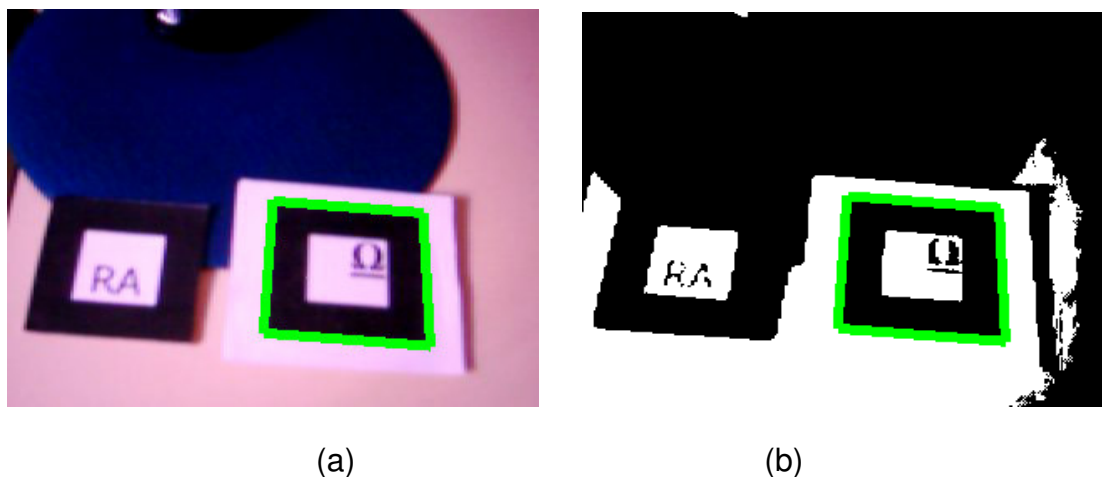


FIGURA 3.2 UTILIDADE DA MOLDURA BRANCA DO MARCADOR

O reconhecimento de padrões identifica os quatro vértices de regiões quadradas, contidas na imagem de vídeo, e compara os símbolos do seu interior com os gabaritos dos marcadores cadastrados pelo usuário (CLAUS, 2005). Caso o retângulo extraído seja semelhante com algum marcador cadastrado, o sistema passa a calcular a sua orientação e posição.

3.1.2 RASTREAMENTO.

O rastreamento no ARToolKit é responsável pelo processamento da imagem, que extrai algumas informações com relação a detecção, e pela identificação de características dos marcadores, além de estimar sua posição e orientação. Nesse caso, a obtenção da posição e orientação do marcador é realizada através da análise da imagem de vídeo, que estabelece o relacionamento entre as coordenadas do marcador e as coordenadas da câmera, como demonstrado na figura 3.3 (KATO, 1999).

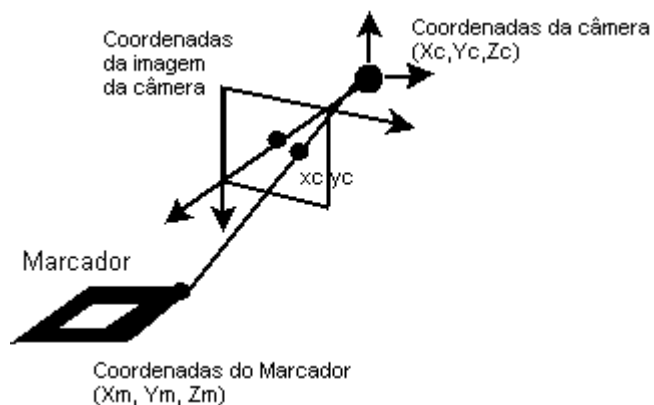


FIGURA 3.3. RELACIONAMENTO ENTRE OS SISTEMAS DE COORDENADAS DO MARCADOR E DA CÂMERA.

O relacionamento entre as coordenadas do marcador e as coordenadas da câmera é realizado por intermédio de uma matriz 3x4, denominada “matriz transformação”. A Figura 3.4 mostra a multiplicação de uma matriz transformação "T" por um ponto 3D no marcador (X_m, Y_m, Z_m), obtendo o ponto correspondente no sistema de coordenadas da câmera (X_c, Y_c, Z_c)

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_1 \\ R_{21} & R_{22} & R_{23} & T_2 \\ R_{31} & R_{32} & R_{33} & T_3 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} = T \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix}$$

FIGURA 3.4. MULTIPLICAÇÃO DE MATRIZES

Para estimar a posição e orientação do marcador, através da análise da imagem de câmera, torna-se necessário utilizar os parâmetros de câmera, a fim de corrigir as distorções inerentes a câmera (LEPETIT, 2005). Assim, é possível estimar, com alguma precisão, o relacionamento entre as coordenadas 3D do mundo e as coordenadas 2D da imagem (ABDULLAH, 2002). O ARToolKit utiliza esses parâmetros de câmera no cálculo da matriz transformação (KATO, 1999).

As funções responsáveis pelo cálculo da matriz transformação no ARToolKit, são “arGetTransMat” e “arGetTransMatCont”. A função “arGetTransMat” é utilizada no momento em que o marcador é detectado, enquanto a “arGetTransMatCont” deverá ser chamada posteriormente, caso o marcador permaneça visível nos quadros de vídeos subseqüentes, o que possibilita o uso de informações obtidas anteriormente para agilizar o cálculo da matriz (KATO, 2002).

Estimada a matriz transformação, a API “OPENGL” é utilizada para ajustar a câmera virtual, posicionar e desenhar o objeto virtual alinhado na visualização do marcador real (PIEKARSKI, 2002).

3.1.3 FUNCIONAMENTO DO ARTOOLKIT

A biblioteca de programação ARToolKit disponibiliza um conjunto de funções que oferecem suporte ao desenvolvimento de aplicações de Realidade Aumentada. Para implementar uma aplicação simples de RA, o programador necessita conhecer as funcionalidades de algumas funções dessa biblioteca e seguir os seguintes passos:

1. Iniciar a configuração do vídeo; ler o arquivo de cadastramento dos marcadores; ler os parâmetros da câmera.
2. Capturar um quadro do vídeo.
3. Detectar e identificar os marcadores
4. Calcular a transformação do marcador relativa à câmera.
5. Desenhar o objeto virtual referente ao marcador.
6. Encerrar a captura de vídeo.

Os passos de número 2 a 5 são repetidos continuamente até a aplicação ser finalizada, enquanto os passos 1 e 6 fazem respectivamente a inicialização e o término da aplicação (KATO, 2000).

O ARToolKit é distribuído com aplicações exemplos, que implementam os passos citados anteriormente, servindo de modelo aos programadores, tanto para o conhecimento de funções da biblioteca, quanto para o auxílio no desenvolvimento de novas aplicações de RA. A tabela 3.1 mostra os passos e as respectivas funções executadas na aplicação "simple" disponibilizada nas distribuições do ARToolKit.

TABELA 3.1 PASSOS E AS FUNÇÕES IMPLEMENTADAS NUM EXEMPLO DE APLICAÇÃO DISTRIBUÍDA COM O ARTOOLKIT .

Passos	Função
1. Inicializa a aplicação	Init
2. Captura do quadro de vídeo.	ArVideoGetImage
3. Detecta os marcadores	ArDetectMarker
4. Calcula a matriz transformação	ArGetTransMat
5. Desenha o objeto virtual.	Draw
6. Fecha a captura de vídeo	Cleanup

O funcionamento de aplicações de RA, como alguns dos exemplos incluídos na distribuição do ARToolKit, executa várias etapas relacionadas às ilustrações da Figura 3.5. Na primeira etapa, a imagem de vídeo capturada, conforme a Figura 3.5a, é convertida em uma imagem binária (em preto e branco), de acordo com o valor do limiar ou (*threshold*), resultando numa imagem como a da Figura 3.5b. Por conseguinte, os quadriláteros, nessa imagem binária, são detectados e comparados com gabaritos de marcadores cadastrados no sistema pelo usuário. Caso haja a identidade entre supostos marcadores e os marcadores conhecidos pelo sistema, a aplicação considera que encontrou um marcador na imagem. A próxima etapa, então, consiste na obtenção da posição e orientação de marcadores (KATO, 2000). Assim, é possível desenhar o objeto virtual sobreposto a seu respectivo marcador, como mostra a figura 3.5c.

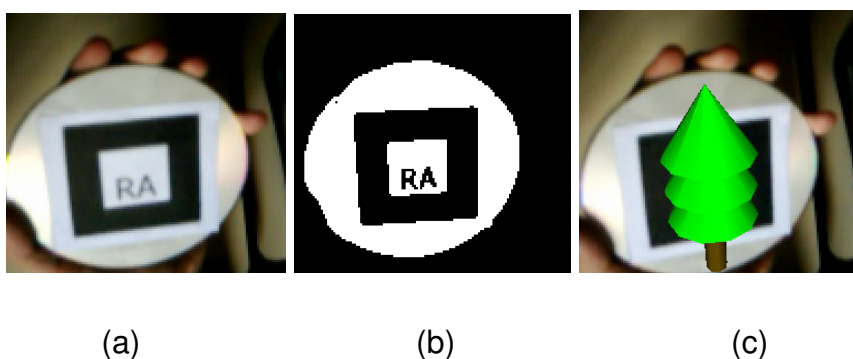


FIGURA 3.5 ETAPAS ENVOLVIDAS NO FUNCIONAMENTO DE UMA APLICAÇÃO DE RA : A) IMAGEM DA CENA COM O MARCADOR, B) A IMAGEM LIMARIZADA E C) OBJETO VIRTUAL SOBREPONDO O MARCADOR.

A Figura 3.6 mostra um diagrama, detalhando as principais etapas realizadas no funcionamento da aplicação.

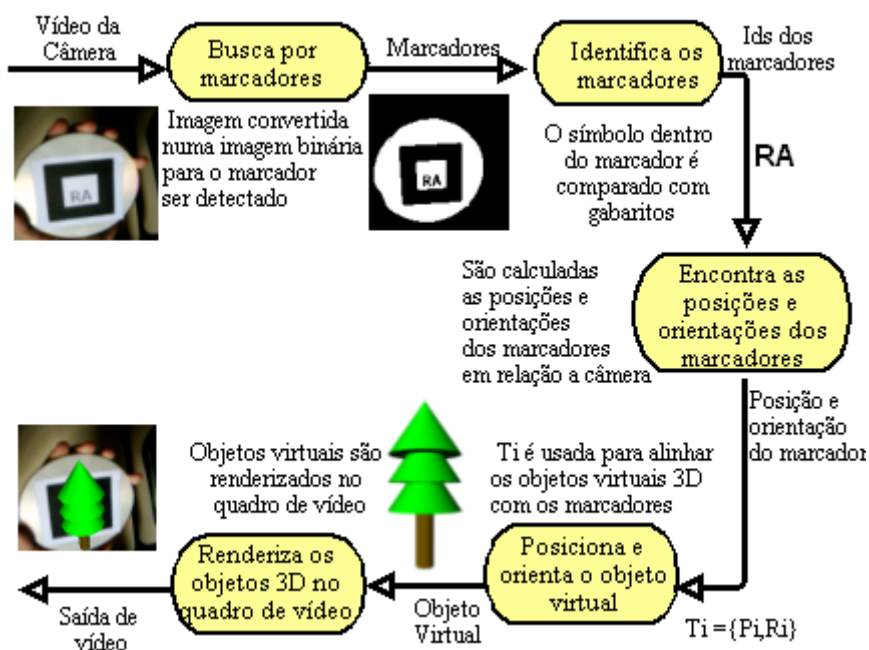


FIGURA 3.6. FUNCIONAMENTO DE UMA APLICAÇÃO DE RA USANDO ARTOOLKIT.

3.2 INSTALAÇÃO E CONFIGURAÇÃO DO ARTOOLKIT

O site do ARToolKit, no laboratório HITL da universidade de Washington (ARToolKit, 2007), disponibiliza o ARToolKit, a sua documentação, projetos e artigos relacionados à biblioteca. A área *download* desse site, além de manter os links de versões mais antigas e algumas outras contribuições, também disponibiliza um link para o site, onde estão disponíveis as versões mais recentes.

3.2.1 INSTALAÇÃO

Para a instalação do ARToolKit deve-se, primeiramente, baixar a versão desejada. O próximo passo é descompactar o arquivo no local de conveniência. Esse local será referenciado abaixo como {ARToolKit} A Figura

3.7 mostra a estrutura de diretórios da versão 2.72.1 do ARToolKit, após descompactação.

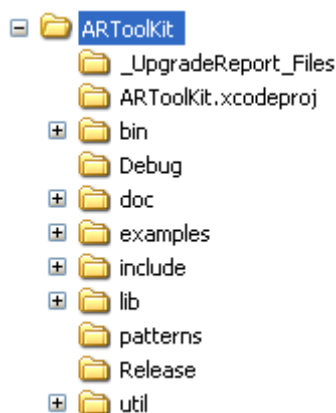


FIGURA 3.7. ESTRUTURA DO DIRETÓRIO DA VERSÃO 2.72.1 DO ARTOOLKIT APÓS A INSTALAÇÃO.

3.2.2 CONFIGURAÇÕES NA PLATAFORMA WINDOWS

Realizada a descompactação, o próximo passo consiste em configurar o sistema, instalando os pré-requisitos exigidos para a compilação do ARToolKit. Os pré-requisitos necessários para a compilação no Windows, Linux/SGI Irix e Mac OS X são descritos no arquivo *README*, contido na pasta "ARToolKit". Os pré-requisitos e passos necessários para a compilação são demonstrados no anexo 1.

3.3 APLICAÇÕES INCLUSAS NO ARTOOLKIT

O ARToolKit é distribuído com diversas aplicações. A versão 2.72.1, por exemplo, após a compilação, disponibiliza várias aplicações executáveis na pasta "bin". Essas aplicações possuem diferentes funcionalidades, permitindo que os usuários as utilizem tanto na configuração, quanto no auxílio no desenvolvimento de suas próprias aplicações. No total são vinte e duas aplicações, das quais seis são denominadas aplicações utilitárias e as demais são exemplos de aplicações de RA.

3.3.1 APLICAÇÕES UTILITÁRIAS

As aplicações utilitárias são responsáveis pela configuração e teste do sistema. Os utilitários de configuração são o “mkpatt, calib_camera2, calib_cparam e calib_distortion”. Já os utilitários de teste do sistema são o “graphicTest” e “videoTest”. Os códigos desses programas estão em “{ARToolKit}\util”.

O “mkpatt” é um programa usado na geração dos arquivos bitmaps (mapa de bits), que relacionam os marcadores aos objetos virtuais. Cada arquivo bitmap contém um conjunto de exemplos de imagens de marcador. Esse conjunto é conhecido como treinamento do marcador. Uma vez executado o “mkpatt”, será pedido para entrar com o arquivo contendo os parâmetros de câmera. Nesse momento, deve-se escrever o caminho para o arquivo, caso esse seja diferente da configuração *default* (“Data/câmera_para”), senão tecla-se o “enter”. A Figura 3.8 demonstra o início do programa.

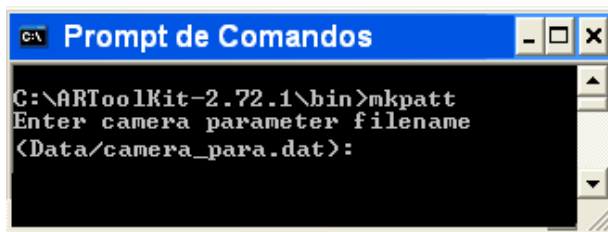


FIGURA 3.8. INÍCIO DA EXECUÇÃO DO MKPATT

O programa exibe uma janela com a imagem do vídeo. O marcador deverá ser enquadrado nessa imagem, de modo a aparecer um retângulo com lados vermelhos, à esquerda e acima, e verdes, à direita e abaixo, nas bordas do marcador, como mostra Figura 3.9.

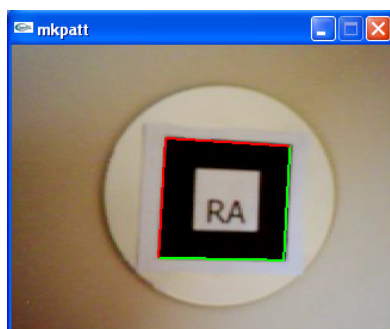


FIGURA 3.9. MARCADOR IDENTIFICADO PELO “MKPATT”

O botão esquerdo no mouse deverá ser apertado e posteriormente será pedida a entrada de um nome para o arquivo. Realizado esse passo, um arquivo será criado na pasta “bin”. Como os arquivos, com dados de configurações das aplicações, geralmente estão localizados na pasta “Data”, deve-se transferir o arquivo gerado para esta pasta.

Os programas “calib_camera2, calib_cparam e calib_distortion” são utilizados para a calibração de câmera. As etapas relacionadas à calibração de câmera são detalhadas em (CONSULARO, 2004).

O utilitário “graphicTest” exibe uma janela com um objeto 3D desenhado em seu interior, enquanto o “videoTest” mostra a uma janela com a imagem capturada pela câmera. Os resultados dos testes que esses programas realizam são as próprias execuções. Caso ocorra algum erro durante a execução de um desses utilitários, provavelmente ocorrerá o mesmo erro na execução dos outros programas fornecidos pela biblioteca.

3.3.2 EXEMPLO DE APLICAÇÕES DE RA

As aplicações de RA, fornecidas junto ao ARToolKit, são exemplos que não só viabilizam o entendimento no funcionamento das funções do ARToolKit, mas também servem como modelo para a produção de novas aplicações.

Os exemplos executáveis, contidos na pasta “bin”, são: “collide, exview, loadmultiple, modetest, multi, optical, paddle, paddledemo, paddleinteraction, range, relation, simple, simple2, simplelite, simplevrml e twoview”. Cada exemplo disponibiliza diferentes funções relacionadas à interação da aplicação, servindo como base aos usuários que desejem implementar novas funcionalidades em suas aplicações. Os códigos desses programas estão localizados em “{ARToolKit}\examples”. A próxima seção demonstrará algumas funcionalidades desses exemplos.

3.3.2.1 TESTE DE COLISÃO

A aplicação “collideTest” possui, como característica principal, a verificação de colisão entre dois marcadores. A função responsável em verificar a colisão é denominada “checkCollisions”. Essa função recebe as estruturas relacionadas às informações de cada um dos dois marcadores, além de um número representante do fator de colisão. O retorno dessa função é um número inteiro “1”, caso os marcadores estejam em colisão e “0”, caso contrário. A Figura 3.10 mostra a aplicação em execução, quando a distância entre os marcadores supera o limite do fator de colisão, sendo desenhado um cubo sobre os marcadores, conforme a Figura 3.10a. Quando a distância entre os marcadores for inferior ao limite, será desenhada uma esfera sobre os marcadores, conforme a Figura 3.10b.

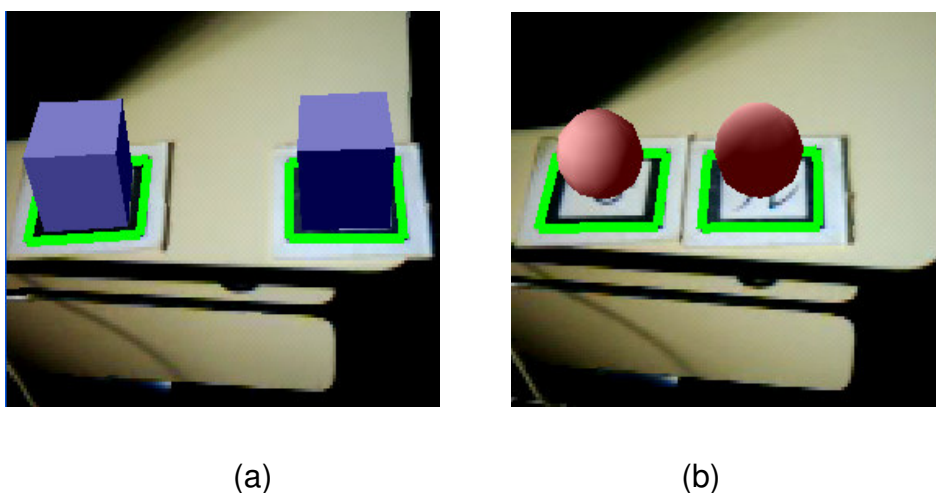


FIGURA 3.10. EXECUÇÃO DA APLICAÇÃO “COLLIDETEST”: A) CUBOS DESENHADOS SOBRE OS MARCADORES DISTANTES E B) ESFERAS DESENHADAS SOBRE MARCADORES PRÓXIMOS.

O exemplo “collide” utiliza os marcadores cadastrados no arquivo “object_data2” contido em “{ARToolKit}\bin\Data”. Dessa forma, o usuário pode associar os seus próprios marcadores, utilizando para isso o “mk_patt” na geração do arquivo bitmaps (mapa de bits) dos marcadores, os quais deverão substituir os marcadores configurados no “object_data2”.

3.3.2.2 USO DA PÁ

O “paddleDemo” (uso da pá) é um dos exemplos que insere um marcador com funcionalidades especializadas a técnicas tangíveis. Esse marcador será denominado pá. A pá é utilizada para interagir com objetos virtuais atrelados a um cartão, contendo vários marcadores, denominado de cartão base. Essa aplicação disponibiliza técnicas que permitem identificar a inclinação da pá, em relação ao cartão base e, assim, atribuir algumas ações, como no caso de despejar uma esfera no cartão base ou pegá-la. As funções que implementam essas técnicas são a "check_incline" e a "check_pickup". Essas funções encontram-se no arquivo "command_sub.c" em “{ARToolKit}\exemplo\paddleDemo”. A Figura 3.11 mostra a sequência de execução do “paddleDemo”.

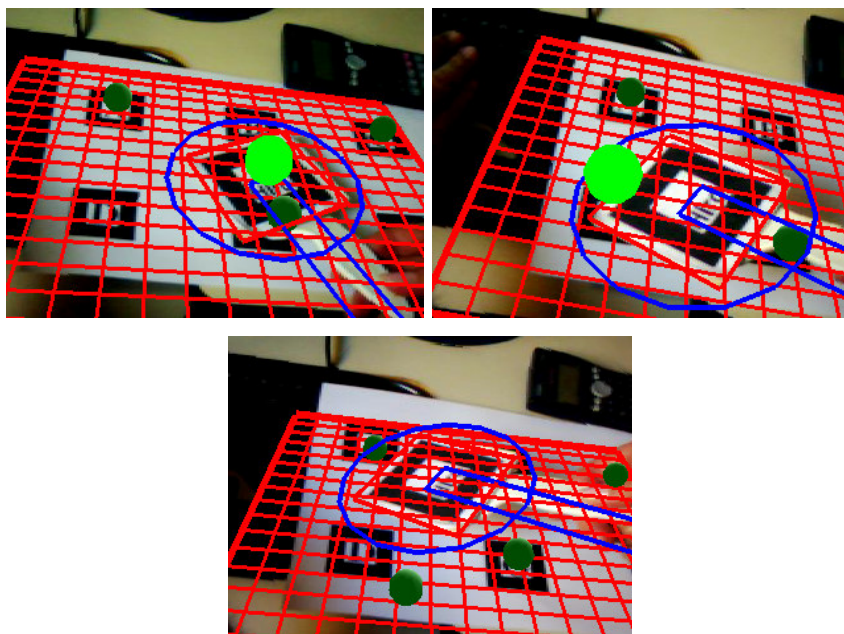


FIGURA 3.11. EXECUÇÃO DO EXEMPLO “PADDLEDEMO”, MOSTRANDO A ESFERA SOBRE A PÁ, A INCLINAÇÃO PARA DESPEJAR A ESFERA NA BASE E A ESFERA FIXADA NA BASE.

O arquivo que relaciona a pá na aplicação é o “paddle_data”, contido na pasta data em “{ARToolKit}\bin\Data”. Já o cartão base, configurado pelo arquivo “marker.dat”, está contido na pasta “multi” localizada em “{ARToolKit}\bin\Data\multi”.

3.3.2.3 SIMPLEVRML

O “simpleVRML” é um programa que possibilita visualizar objetos virtuais, escritos na linguagem VRML, sobrepostos aos marcadores. Esse exemplo utiliza a biblioteca OpenVRML para a renderização dos objetos VRML. A Figura 3.12 exibe o resultado da execução do “simpleVRML”.



FIGURA 3.12. RESULTADO DA EXECUÇÃO DO *SIMPLEVRML*

Essa aplicação utiliza o arquivo de configuração “object_data_vrml” para atrelar um marcador a um objeto virtual. Localizado em “{ARToolKit}\bin\Data”, esse arquivo armazena o relacionamento entre o arquivo bitmap do marcador, disposto no diretório “Data”, e o arquivo de referência ao objeto VRML, localizado em “{ARToolKit}\bin\wrl”. A Figura 3.13 mostra o arquivo “object_data_vrml”, que contém dois marcadores cadastrados: o “patt.hiro” e o “patt.kanji”. O marcador “patt.hiro” está associado a um arquivo de referência VRML, denominado “bud_B.dat”, enquanto o marcador “patt.kanji” encontra-se associado ao arquivo “snoman.dat”.

```
#the number of patterns to be recognized
2

#pattern 1
VRML      Wrl/bud_B.dat
Data/patt.hiro
80.0
0.0 0.0

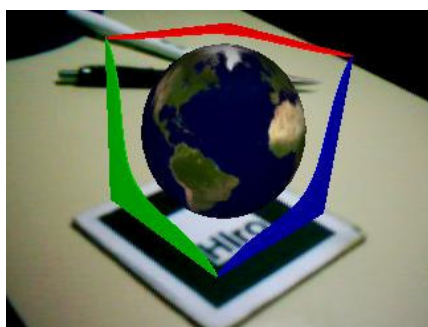
#pattern 2
VRML      Wrl/snoman.dat
Data/patt.kanji
80.0
0.0 0.0
```

FIGURA 3.13. REFERÊNCIA DE MARCADORES E OBJETOS VIRTUAIS VRML

Em virtude dessa associação de arquivos, é possível adicionar objetos virtuais de maneira muito simples, bastando editar o arquivo “object_data_vrml”. Os demais exemplos disponibilizados no ARToolkit possuem os objetos virtuais implementados diretamente em seu código, de forma que, para adicionar novos elementos virtuais, é necessário incluir o código do objeto virtual em OpenGL na aplicação e a compilar novamente.

3.4 SOFTWARE BASEADOS NO ARTOOLKIT.

Existem Software de Realidade Aumentada que estendem o rastreamento do ARToolkit a diferentes abordagens funcionais e estruturais. Alguns exemplos desses software são: O ARToolkitPlus, considerado uma biblioteca de RA otimizada para uso em dispositivos portáteis como PDAs e alguns telefones celulares (WAGNER, 2003); o jARToolkit, que possibilita escrever aplicações de RA em java, acessando funções da biblioteca ARToolkit através da interface JNI (GEIGER, 2002); o ARToolkit Python, um “bind” Python que encapsula as funções do ARToolkit, permitindo a exploração das vantagens do Python nas implementações, que podem ser feitas sem a compilação de código (KIRNER, 2007); osgART framework baseado no ARToolkit, que implementa a biblioteca gráfica OpenSceneGraph, apresentando alta qualidade na renderização de objetos virtuais e possibilidade de importar e exportar arquivos 3D gerados pelo 3D Studio Max e Maya (LOOSER, 2006). A Figura 3.14 mostra a execução de exemplos do osgART.



(a)



(b)

FIGURA 3.14. RESULTADO DA EXECUÇÃO DE EXEMPLOS DO OSGART: A) EXEMPLO OSGARTSIMPLENPR E B) EXEMPLO OSGARTVIDEOPLANE.

4 COLABORAÇÃO COM REALIDADE AUMENTADA

O trabalho realizado por um grupo de pessoas é caracterizado pela interação da diversidade entre os integrantes, para o desenvolvimento de um objetivo comum.

A contribuição de diferentes habilidades organizadas para a elaboração de tarefas é realizada desde o começo da humanidade. Na sociedade coletora de alimentos, por exemplo, os homens trabalhavam juntos, caçavam em grupos e partilhavam entre si as tarefas e os alimentos (GOMES, 2002). Já na sociedade moderna, a atuação de um grupo, para execução de um trabalho em comum, foi expandida com o surgimento de ferramentas que promoveram suporte a colaboração, como o computador, que possibilita as pessoas uma atuação não só face a face, mas também, geograficamente separadas.

A importância dos ambientes colaborativos, na atualidade, é notada sensivelmente, devido à globalização, possibilitando várias oportunidades aos seus usuários, desde prover o compartilhamento de informações locais até o favorecimento da interação assíncrona entre pessoas situadas geograficamente distantes.

A realidade aumentada, sendo uma interface computacional, pode ser amplamente utilizada como interface desses ambientes colaborativos, além de possibilitar a criação de novas aplicações, devido às inovações das técnicas de interações promovidas por essa interface.

4.1 CONCEITOS DE COLABORAÇÃO

O computador, segundo FOUSS (2000), é uma ferramenta que fornece suporte ao trabalho, permitindo agilizar e facilitar o desenvolvimento de tarefas dos usuários, como simulações físicas e matemáticas, modelação de fenômenos ou simplesmente uma ferramenta para editar texto. Embora o computador seja amplamente utilizado para auxiliar a elaboração de tarefas,

grande parte de seu foco é a produtividade individual, ignorando, dessa forma, o fato de grande parte do trabalho ser realizado em grupo.

Os problemas relacionados ao trabalho em grupo e as formas de uso de tecnologia para auxiliar esse processo são abordados pela área de pesquisa multidisciplinar conhecida como *Computer Suported Cooperative Work* (CSCW) (FOUSS, 2000).

O foco da CSCW é estudar as técnicas e ferramentas dos *groupware* a fim de relatar os impactos sociais, psicológicos e organizacionais promovidos pelo uso dessas tecnologias (RAMA, 2006). O termo *groupware* é utilizado pela comunidade CSCW para se referir à tecnologia (hardware e software) utilizada para o trabalho em grupo (GRUDIN, 1994)

A classificação do *groupware*, geralmente, é realizada relacionando o tempo/espço, como mostra a Tabela 4.1. O sistema pode auxiliar a interação de um grupo, onde os integrantes encontram-se reunidos face a face, no mesmo local atuando no mesmo tempo, como aplicações de salas de conferências que promovem suporte as reuniões. Caso os integrantes do grupo encontrem-se dispersos geograficamente, a interação poderá ocorrer ao mesmo tempo, de forma síncrona, como em vídeo conferências, ou assíncrona, como os e-mails. Outra forma que o sistema pode abranger consiste na presença do grupo no mesmo local, porém interagindo em tempo distinto, de forma assíncrona. Esse tipo de *groupware* pode ser encontrado em salas de projetos para auxiliar o desenvolvimento do trabalho (FOUSS, 2000; Ellis, 1991).

TABELA 4.1. CLASSIFICAÇÃO DA INTERAÇÃO DO GROUPWARE EM RELAÇÃO AO TEMPO E ESPAÇO.

	Mesmo Tempo	Tempo Diferente
Mesmo Local	Interação face a face	Interação assíncrona
Local Diferente	Interação síncrona e distribuída	Interação assíncrona e distribuída

ELLIS (1991) define *groupware* como sistema baseado em computador que viabiliza o trabalho em grupo para o desenvolvimento de tarefas em comum, oferecendo uma interface para um ambiente compartilhado. O objetivo do *groupware* é auxiliar a colaboração, de forma que a interação entre os integrantes do grupo abranja aspectos relacionados à comunicação, cooperação e coordenação (GRUDIN, 1996).

As técnicas que envolvem a interação entre os integrantes do grupo são fundamentais ao sistema colaborativo, visto que a colaboração é realizada entre as pessoas, através da troca de informações (comunicação), atuação organizada (coordenação) e trabalho em conjunto num espaço compartilhado (cooperação). Dessa forma, o sistema computacional colaborativo deve ressaltar a percepção do papel de cada integrante na elaboração de determinada tarefa, disponibilizando ferramentas que auxiliem a troca de informações e forneçam mecanismos de gerenciamento da atuação entre esses integrantes (FUKS, 2003).

Um grande número de sistemas colaborativos tem explorado formas de compartilhamento espaciais, a fim de oferecer recursos para potencializar a percepção de seus usuários. A interface de realidade virtual, por oferecer técnicas avançadas de interação, passou a ser utilizada em alguns sistemas colaborativos, recebendo a denominação de *CVEs (Colaborative Virtual Enviroments)*. Os CVEs passaram a ter um papel central em CSCW, visto a rápida ascensão de inúmeras pesquisas e desenvolvimentos desses ambientes (BÜSHER, 1999).

Os CVEs compartilham um espaço virtual em comum, onde é possível interagir tanto entre os usuários, quanto com o próprio ambiente (GOEBBELS, 2003). Os usuários podem realizar a interação face a face ou remotamente. Na colaboração face a face, os usuários compartilham o mesmo local físico e operam no ambiente virtual por meio da comunicação interpessoal natural, usando gestos e expressões faciais, que facilitam a compreensão sobre a fala do locutor (AGRAWALA, 1997). A Figura 4.1, demonstra usuários interagindo num ambiente colaborativo face a face.



FIGURA 4.1. INTERAÇÃO NUM AMBIENTE COLABORATIVO FACE A FACE (AGRAWALA, 1997).

Nos ambientes colaborativos remotos, os usuários geograficamente dispersos podem interagir com elementos sintéticos distribuídos, como objetos virtuais e representações dos outros participantes (OTTO, 2006). A comunicação, nesse caso, pode ser realizada por vários meios de transmissão de informação, como, texto, áudio e vídeo. A Figura 4.2 demonstra a execução de um ambiente colaborativo remoto, no qual os usuários estabelecem comunicação através de áudio e vídeo em tempo real.



FIGURA 4.2. EXEMPLO DE AMBIENTE COLABORATIVO REMOTO (GOEBBELS, 2003).

Os CVEs, tanto remoto, quanto face a face, viabilizam a colaboração em ambientes 3D, nos quais, geralmente, os participantes são totalmente imersos, para colaborarem com maior grau de naturalidade. Porém,

existem limitações, como: a atuação permanente no ambiente virtual, fora do ambiente real intuitivo do usuário com suas tradicionais ferramentas; o alto custo de dispositivos de realidade virtual, utilizados para rastreamento, visualização e interação; além de certo desconforto sentido na utilização desses recursos, que são impostos para realizar o trabalho colaborativo. Dessa forma, acarreta-se a restrição do número de usuários, visto a tendência de rejeição de ferramentas colaborativas, que forcem os usuários a mudarem a sua maneira usual de trabalho e os custos proibitivos desses dispositivos tecnológicos (KATO, 1999).

A realidade aumentada, por outro lado, possibilita que a colaboração seja realizada, a partir da visualização de elementos sintéticos no ambiente real, interagindo com os usuários de forma altamente intuitiva, através de dispositivos simples e acessíveis a grande maioria de usuários de computador.

4.2 COLABORAÇÃO EM AMBIENTES DE REALIDADE AUMENTADA

A tecnologia de realidade aumentada incrementa os ambientes colaborativos, oferecendo interfaces que ressaltam formas intuitivas de interação atreladas à visualização dos elementos sintéticos gerados por computador, no ambiente real. Dessa maneira, são superados eventuais problemas relacionados à interação com dados 3D, ou entre os participantes do ambiente colaborativo, promovendo, de forma natural, a atuação do trabalho compartilhado (BILLINGHURST, 2002).

Os ambientes colaborativos de RA podem oferecer áudio espacial e visualização da comunicação corporal dos participantes, como gestos e expressões faciais, que são compartilhados no mundo real dos usuários, não separados da realidade, como em CVEs.

RENGENBRECHTD (2002) afirma que os sistemas colaborativos de RA oferecem aos múltiplos usuários a possibilidade de compartilhar pelo menos um espaço em comum no ambiente. A colaboração em RA pode ser realizada de duas formas, face a face ou remotamente.

4.2.1 COLABORAÇÃO FACE A FACE

O uso de interfaces de RA nos *groupwares* aprimora o compartilhamento do espaço físico de trabalho, além de oferecer técnicas avançadas de interação, próximas da forma natural de colaboração face a face (BILLINGHURST, 2002). A Figura 4.3 demonstra um exemplo de sistema de colaborativo de RA face a face, no qual os usuários utilizam HMDs com câmeras acopladas, que possibilitam a visualização de elementos virtuais sobrepostos a uma mesa, enquanto, no mesmo momento, são capazes de visualizar uns aos outros.



FIGURA 4.3. EXEMPLO DE AMBIENTE COLABORATIVO DE RA FACE A FACE (BILLINGHURST, 2002).

A colaboração face a face em RA, ao contrário de alguns *groupwares*, não separa o espaço de comunicação do espaço de atuação, possibilitando aos usuários utilizarem a percepção tradicional dos objetos reais, durante a manipulação dos objetos virtuais.

Segundo BILLINGHURST (2002), estudos compararam a interface colaborativa de RA com outras tecnologias, como ambientes imersivos de RV. Os resultados obtidos mostraram que os usuários consideram a interface de RA colaborativa mais natural e significativamente mais fácil de interagir, em relação ao sistema de RV baseado em projeção.

4.2.2 COLABORAÇÃO REMOTA

A colaboração remota possibilita que usuários, localizados em posições geograficamente diferentes, possam colaborar no desenvolvimento de algum trabalho em comum.

Os ambientes colaborativos de RA remotos podem oferecer suporte à comunicação entre os usuários, no desenvolvimento do trabalho colaborativo por meio de vídeo-conferência de RA, permitindo adicionar, no ambiente real do usuário, imagens capturadas em tempo real dos participantes remotos, como mostra a Figura 4.4 (KATO,1999; BILLINGHURST, 2002; BILLINGHURST, 1999). O trabalho colaborativo não necessariamente é realizado pelas técnicas de interação oferecidas pelas aplicações RA, mas sim apoiado por uma visualização compartilhada dos usuários remotos. Por exemplo, pode-se utilizar uma vídeo-conferência de RA para auxiliar a elaboração de uma carta em grupo.



FIGURA 4.4. VÍDEO-CONFERÊNCIA EM RA (BILLINGHURST, 2002)

A outra possibilidade de colaboração remota em RA ressalta o desenvolvimento de trabalhos num ambiente compartilhado, através de sistemas de RA multi-usuários, que oferecem técnicas mais intuitivas para a coordenação, cooperação e comunicação.

Nesse caso, os sistemas utilizam formas de gerenciamento (coordenação) e interação compartilhada (cooperação), oferecidas pelas aplicações de RA, para viabilizar a contribuição dos usuários remotos na elaboração e visualização do trabalho colaborativo, no cenário real. A comunicação entre os usuários, nesse caso, pode ser realizada através da

vídeo-conferência de RA, ou utilizando trocas de mensagens textuais ou sonoras. A Figura 4.5, demonstra um exemplo de ambiente colaborativo remoto de RA, onde os usuários podem atuar remotamente, num ambiente compartilhado, para a modelagem de cenários, por exemplo.

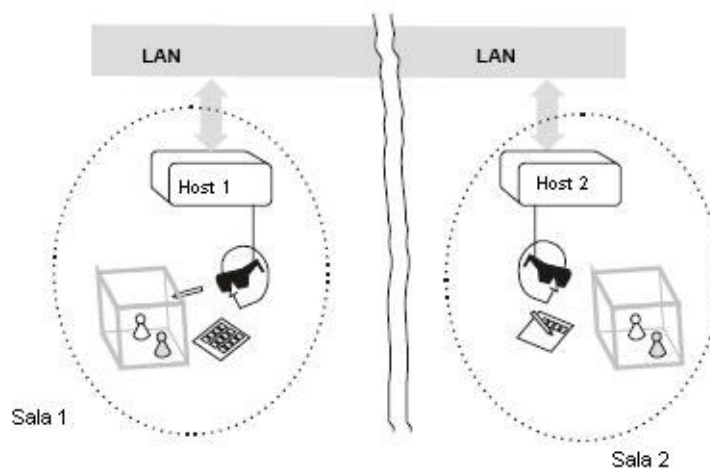


FIGURA 4.5. EXEMPLO DE COLABORAÇÃO REMOTA, DOIS USUÁRIOS GEOGRAFICAMENTE SEPARADOS ATUANDO NO AMBIENTE COMPARTILHADO (SCHMALSTIEG, 2000).

4.3 FERRAMENTAS/EXEMPLOS DE COLABORAÇÃO E AMBIENTES DE REALIDADE AUMENTADA

Existem vários sistemas de RA que auxiliam o desenvolvimento do trabalho colaborativo. A seguir, serão apresentados três desses sistemas colaborativos.

MagicMeeting

O MagicMeeting, Figura 4.6, combina a tecnologia colaborativa de RA com técnicas de interação tangíveis. Esse sistema permite realizar a colaboração face a face e a interação dos usuários com dados 2D e 3D, no ambiente compartilhado. O rastreamento utilizado no MagicMeeting é baseado no rastreamento do ARToolKit (REGENBRECHT, 2002).



FIGURA 4.6. MAGICMEETING (REGENBRECHT, 2002).

DWARF

O Dwarf consiste num framework que possibilita o desenvolvimento de aplicações colaborativas de RA. Esse framework é baseado no conceito de serviços distribuídos colaborativos. O sistema possui arquitetura modular, possibilitando que componente seja reusados entre aplicações, além de possibilitar a execução em ambientes heterogêneos. Sua infra-estrutura utiliza CORBA, que funciona em diversos hardwares e sistemas operacionais, possibilitando utilizar uma vasta gama de linguagem de programação para o desenvolvimento dos serviços (BRUEGGE, 2003; BAUER, 2001). A Figura 4.7 demonstra a execução do DWARF, usando vários dispositivo, como LapTops, PDA e no sistema que está projetando o ambiente.



FIGURA 4.7. EXECUÇÃO DO DWARF (BRUEGGE E KLINKER, 2003).

Studierstube

O Studierstube é um framework de RA que possibilita elaborar tanto o trabalho colaborativo face a face, quanto remotamente, ou até uma mistura das duas formas de colaboração, como demonstra a figura 4.8.

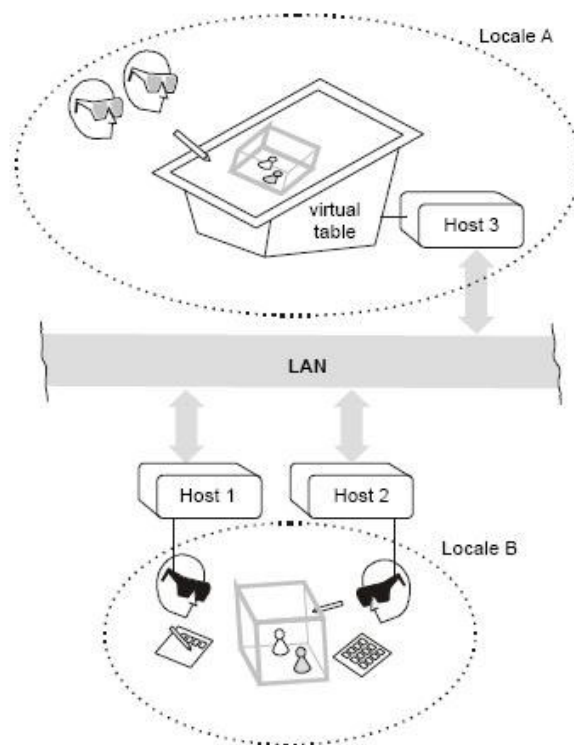


FIGURA 4.8. EXEMPLO DE COLABORAÇÃO FACE A FACE E REMOTA (SCHMALSTEIG, 2000).

Os usuários desse sistema utilizam HMDs que possibilitam a cada usuário escolher pontos de vista individuais no ambiente compartilhado. A interação no ambiente pode ser realizada, através de vários dispositivos de entrada que são gerenciados pelo OpenTracker. Geralmente, as aplicações elaboradas com o Studierstube utilizam o painel pessoal de interação (*personal interaction panel*), para controlar as aplicações. Esse dispositivo consiste numa interface composta de caneta e uma base, ambas acopladas a mecanismos de rastreamento eletromagnético. A Figura 4.9. mostra um exemplo do painel pessoal de interação (SCHMALSTEIG, 2000; PSIK, 2004).

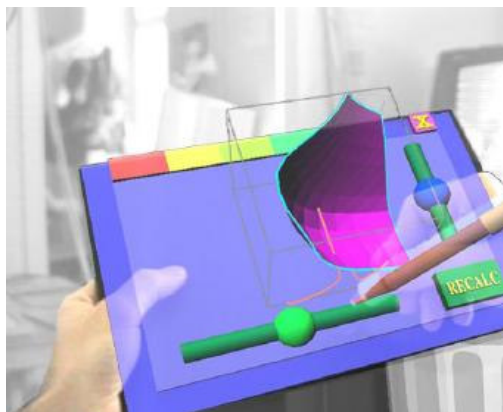


FIGURA 4.9. PAINEL PESSOAL DE INTERAÇÃO (SCHMALSTEIG, 2000).

A colaboração entre diferentes usuários requer a distribuição de diversos estados da aplicação entre as outras aplicações, que podem estar executando em ambientes heterogêneos. Dessa maneira, o Studierstube implementa o renderizador Open Inventor (OIV), distribuído, obtendo um poderoso grafo de cena compartilhado, distribuído e orientado a objetos (PSIK, 2004).

4.4 ASPECTOS DE COMUNICAÇÃO EM REDE.

Pessoas situadas em diferentes localidades, ao redor do globo, podem realizar tarefas em conjunto, por intermédio de sistemas computacionais que são baseados na integração de redes de comunicações e técnicas de sistemas distribuídos. Nessa linha, encontram-se os sistemas colaborativos de realidade aumentada, que transmitem dados através da rede, a fim de promover formas distribuídas de interação entre usuários remotos.

A colaboração entre os usuários, no ambiente de realidade aumentada, é realizada junto a objetos virtuais compartilhados, sobrepostos no mundo real. Embora o ambiente real de cada usuário seja diferente dos demais, a colaboração em RA fornece a sensação de compartilhamento do espaço de atuação entre os participantes. Os usuários se sentem atuando no espaço comum entre outros usuários, em virtude da interação com os objetos virtuais compartilhados, que são os elementos de enfoque no trabalho nesse ambiente. Além da sensação de compartilharem um lugar em comum, os

usuários desenvolvem a sensação de interagirem em tempo real, extraindo dessa forma a idéia de tempo compartilhado. Essas características permitem classificar os ambientes colaborativos de RA como um caso específico dos sistemas distribuídos.

Segundo Tanenbaum (1995), os sistemas distribuídos são um conjunto de computadores independentes que aparecem para os usuários do sistema como um único computador. Para COULOURIS (2001), um sistema distribuído é um sistema, no qual os componentes de hardware ou software, localizados em uma rede de computadores, se comunicam e coordenam suas ações, através da passagem de mensagens. Uma análise sobre as duas definições possibilita identificar diferenças nas abordagens entre os dois autores. A definição de TANENBAUM aborda, de maneira mais próxima, os aspectos relacionados ao hardware e seus sistemas operacionais, que devem permitir a atuação autônoma do sistema. Já a definição de Coulouris ressalta a importância tanto de hardware, quanto de software, para realizar a comunicação e coordenação do sistema. Dessa forma, a definição de Coulouris é mais próxima à proposta desse trabalho, que consiste em desenvolver um ambiente colaborativo de realidade aumentada.

A construção de sistemas distribuídos empreende certos desafios que caracterizam o desenvolvimento desses sistemas. A seguir, serão apresentadas as propriedades dos desafios que devem ser considerados na implementação de um sistema distribuído:

- **Heterogeneidade:** consiste na capacidade do sistema operar nas diferentes variedades de redes, sistemas operacionais, hardware e linguagens de programação.
- **Abertura:** é a característica que determina se o sistema pode ser estendido e re-implementado em várias formas, devido aos grandes desafios impostos na integração de diferentes componentes, existentes nos diversos tipos de hardware e software.

- **Segurança:** consiste em proteger de forma adequada os recursos compartilhados e manter em segredo as informações que são transmitidas em mensagens pela rede.
- **Escalabilidade:** característica que permite a expansão dos recursos e números de usuários, sem acarretar danos ao funcionamento do sistema.
- **Tratamento de falhas:** possibilita que o sistema tenha capacidade de operar mesmo na presença de falhas. Para isso, cada componente do sistema precisa reconhecer possíveis falhas e suas conseqüências, para atuarem de forma apropriada no tratamento de falhas.
- **Concorrência:** consiste na capacidade do sistema atender a requisições de diversos usuários, procurando acessar recursos compartilhados ao mesmo tempo. Para isso, cada recurso deve ser projetado para operar de modo seguro em um ambiente concorrente.
- **Transparência:** consiste no fator que possibilita obter a imagem de um único sistema, escondendo toda a distribuição do usuário, até mesmo de alguns programas. O conceito de transparência pode ser aplicado para vários aspectos de um sistema distribuído.
 - ✓ **Localização:** o usuário desconhece a localização dos recursos.
 - ✓ **Migração:** os recursos podem mover-se sem modificar seus nomes.
 - ✓ **Replicação:** os usuários desconhecem a quantidade de cópias existentes.
 - ✓ **Concorrência:** múltiplos usuários podem compartilhar recursos automaticamente.
 - ✓ **Paralelismo:** atividades que podem ocorrer em paralelo, sem o conhecimento do usuário.

O conhecimento destas propriedades é de extrema importância na construção de sistemas distribuídos, permitindo que os desenvolvedores sejam cuidadosos e relevem esses aspectos, durante o desenvolvimento do projeto. Dessa forma, é possível construir sistemas para serem utilizados no ambiente real, onde serão expostos as mais variadas circunstâncias, em face a possíveis dificuldades e problemas. Um aspecto relevante no projeto do sistema distribuído é o modelo arquitetural a ser adotado, pois é a estrutura desse modelo, que poderá garantir o seu funcionamento perante as exigências requisitadas tanto no presente, quanto as que serão demandadas no futuro.

O modelo arquitetural define a maneira que os componentes dos sistemas interagem entre si, além da forma como são mapeados em uma rede de computadores. As maiores preocupações da estrutura da arquitetura de um sistema consistem em torná-lo confiável, gerenciável, adaptável e a custos efetivos. A seguir, serão apresentados dois modelos arquiteturais de SD (COULOURIS, 2001): os modelos cliente-servidor e o peer-to-peer.

Modelo cliente-servidor

O modelo cliente-servidor possui grande importância no cenário dos sistemas distribuídos. Basicamente, essa estrutura consiste no servidor atendendo as solicitações de clientes. Existem vários exemplos desse modelo, conforme a Web e serviços de Internet como FTP, notícias e e-mails (COULOURIS, 2001). A Figura 4.10 mostra um exemplo do modelo cliente-servidor.

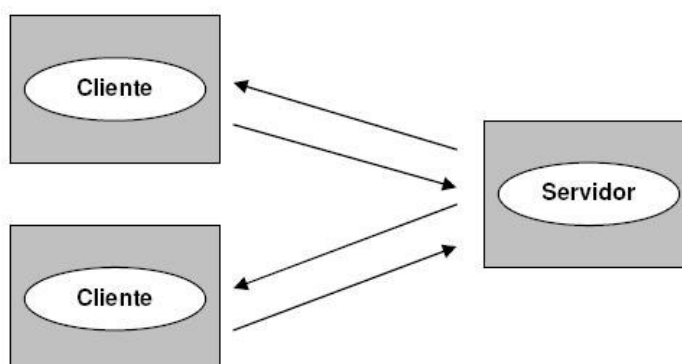


FIGURA 4.10. MODELO CLIENTE-SERVIDOR.

Nessa arquitetura centralizada, o número elevado de clientes pode transformar o servidor em um gargalo, que eventualmente não poderá lidar no processamento de todas as requisições recebidas (PIRCHHEIM, 2006).

Peer-to-Peer

No modelo peer-to-peer, todos os processos desempenham papel semelhante, interagindo de forma cooperativa, como *peers* para executar tarefas distribuídas, sem qualquer distinção entre clientes e servidores. A eliminação do processo servidor reduz o atraso da comunicação inter-processo para acesso aos objetos locais. Outro benefício desse modelo consiste na redução do número de mensagens enviadas, em relação ao modelo cliente-servidor. Isso acontece, devido à redução do número de conexões e conseqüentemente a quantidade de mensagens enviadas entre as aplicações. Por exemplo, no modelo cliente-servidor, um servidor com um conjunto j de clientes realiza j conexões; por outro lado, no modelo per-to-per, um conjunto de n peers realizam $n-1$ conexões. Assim, caso um peer realize m interações por segundo, serão enviados $m(n-1)$ mensagens por segundo, reduzindo m mensagens por segundo, em comparação ao modelo cliente-servidor (COULOURIS, 2001; PIRCHHEIM, 2006). A Figura 4.11 demonstra uma aplicação distribuída baseada no modelo peer-to-peer.

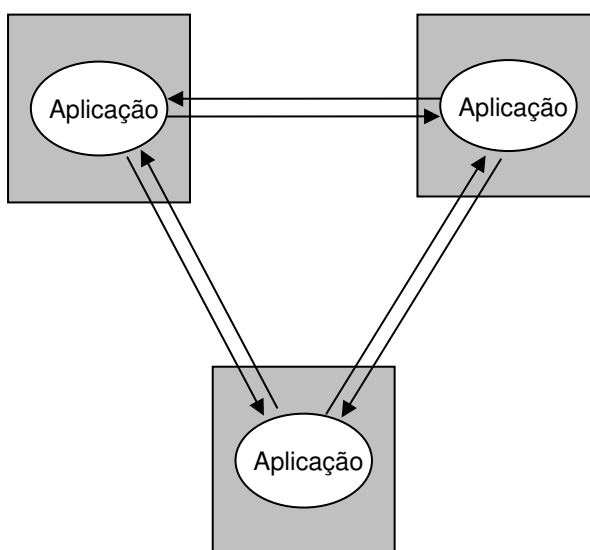


FIGURA 4.11. MODELO PEER-TO-PEER.

A utilização de modelos de comunicação como *broadcast* ou *multicast*, nesse caso, possibilitaria reduzir o número de envios de mensagens de $m(n-1)$, para somente m . As mensagens broadcast são recebidas por todos *hosts* da rede local, enquanto as mensagens multicast são recebidas apenas por um grupo definido na rede.

Protocolos TCP e UDP

A troca de mensagens entre processos de sistemas distribuídos é gerenciada por protocolos de comunicação, oferecendo um conjunto de regras e formatos para a comunicação entre diferentes processos.

Existem diversos protocolos, sendo os mais conhecidos o *Transmission Control Protocol* (TCP) e o *User Datagram Protocol* (UDP), que são protocolos do nível de transporte. O TCP consiste num protocolo confiável, pois oferece garantia de entrega das mensagens. Já o UDP consiste num protocolo não confiável, visto que não garante a entrega de mensagens, não ordena mensagens de entrada e não fornece informações para o controle da velocidade com que as informações são transmitidas entre as máquinas (COULOURIS, 2001, COMMER, 1998).

Um ponto importante, na comparação desses protocolos, consiste na possibilidade de aplicações, que utilizam o protocolo confiável, o TCP, executarem dez vezes mais lentas que aplicações, utilizando o UDP (GOSSWEILER, 1994).

Os protocolos TCP e UDP suportam comunicação inter-processos, podendo ser realizada através da passagem de mensagens via *sockets*.

Sockets

Segundo SILBERSCHATZ (2000), um *socket* é definido como a extremidade da comunicação entre dois processos. Um par de processos se comunica em uma rede, através de um par de sockets. Para um processo receber uma mensagem, o seu socket deve estar ligado a uma porta e um

endereço Internet do computador, onde está sendo executado. Dessa forma, para um processo enviar uma mensagem, o seu socket deve referenciar a porta e o endereço Internet do computador de destino (COULOURIS, 2001).

Os *sockets* viabilizam uma forma de comunicação comum, robusta e eficiente entre processos, mas considerada de baixo nível. Um dos motivos dessa classificação consiste no fato da troca de mensagens, via *sockets*, ser um fluxo não estruturado de bytes entre processos. Dessa maneira, a estrutura de dados é definida pela aplicação. Assim, um dos cuidados a serem tomados na sua implementação consiste em possíveis problemas relacionados à heterogeneidade dos sistemas computacionais (SILBERSCHATZ, 2000; MATTHIESEN, 2006).

Obstantes aos possíveis percalços encontrados em técnicas de comunicação de baixo nível, encontram-se técnicas que viabilizam a transmissão de representações das estruturas de dados ou valores primitivos. Essas técnicas são utilizadas para realizar invocações de métodos remotos (*Remote Method Invocation*, RMI), ou chamadas de procedimentos remotos (*Remote Procedure Call*, RPC) (COULOURIS, 2005).

RMI e RPC

O RMI permite a um objeto invocar um método contido em um objeto em um processo remoto. Dois exemplos de sistemas que possibilitam a invocação de métodos remotos são o *Common Object Broker Architecture* (CORBA) e Java RMI. Esses sistemas permitem instanciar e acessar objetos, através de uma interface orientada a objetos bem definida. O CORBA, por exemplo, fornece a Interface Definition Language (IDL), que possibilita implementar interfaces em independentes linguagens de programação, permitindo a comunicação de aplicações escritas em diferentes linguagens. A invocação de métodos remotos em CORBA é realizada, através do *Object Request Broker* (ORB), responsável em localizar o objeto remoto no sistema distribuído (COULOURIS, 2001; SILBERSCHATZ, 2000; YANG, 1996).

O RPC possui características muito similares ao RMI, visto que um programa cliente chama um procedimento em outro programa, em execução no servidor. Porém, ao contrário do RMI, o RPC não realiza a comunicação entre objetos distribuídos. Oferece a programação procedural, possibilitando apenas as chamadas remotas de funções ou procedimentos (COULOURIS, 2001; SILBERSCHATZ, 2000). Um exemplo de sistema que possibilita chamadas de procedimentos é o Sun RPC.

O RPC e RMI permitem estabelecer a comunicação de processos distribuídos em alto nível. Dessa maneira, aplicações que implementam essas tecnologias podem oferecer serviços avançados, como a interoperabilidade entre aplicações escritas em diferentes linguagens.

Os benefícios avançados obtidos com as abstrações dessas tecnologias acabam, entretanto, acarretando custos na performance. Um estudo realizado por TILLEY (1999) comparou o desempenho na transmissão de *callbacks* entre uma *GUI* cliente e uma aplicação servidora, utilizando *socket* em C++, *socket* em Java, RMI do Java e RMI do CORBA. Os resultados dos testes comprovaram que a transmissão de mensagens utilizando *socket* em C++ foi cerca de quatro vezes mais rápida, em relação ao RMI do Java e do CORBA, e cerca de duas vezes mais rápida do que o *socket* em Java.

5 SISTEMAS DE AUTORIA COM REALIDADE AUMENTADA

Os sistemas de autoria oferecem ferramentas para a manipulação de técnicas e recursos computacionais complexos, sem exigir de seus usuários conhecimentos específicos em programação. O objetivo desses sistemas de autoria é possibilitar que usuários possam criar e desenvolver as suas próprias tarefas, não dependendo de programadores para realizá-las.

Na atualidade, existem diversos tipos de sistemas de autoria para as mais variadas necessidades. Algumas soluções de autoria são encontradas para a criação de aplicações Web, permitindo o desenvolvimento de *websites* complexos, sem a exigência de conhecimentos em programação. Por exemplo, existem os sistemas que promovem suporte a ensino a distância, como o Moodle (*Modular Object Oriented Dynamic Learning Enviroment*), que possibilita professores e alunos desenvolverem cursos webs capazes de gerenciar a aprendizagem (MOODLE, 2007).

A autoria aliada às interfaces de realidade virtual ou realidade aumentada permitem aos usuários elaborarem e caracterizarem os seus próprios objetos virtuais. Em realidade virtual, ferramentas de autoria permitem desenvolver aplicações gráficas altamente interativas em 3D. Da mesma maneira, a autoria em realidade aumentada deve possibilitar que usuários elaborem as suas próprias aplicações (LEE, 2005).

5.1 SISTEMA DE AUTORIA DE MUNDOS VIRTUAIS

O desenvolvimento de mundos virtuais envolve uma grande quantidade de conceitos e técnicas de interação que precisam ser programadas. Dessa forma, geram-se restrições aos usuários desenvolvedores de mundos virtuais, visto que usuários, sem conhecimentos específicos em programação, acabam encontrando dificuldades no desenvolvimento de suas aplicações (HENDRICKS, 2003).

A solução para as dificuldades dos não programadores consiste na construção de ferramentas de autoria, que empregam técnicas de

programação visual para o desenvolvimento de aplicações gráficas interativas em 3D (NAJORK, 1994). HENDRICKS (2003) afirma que a criação de uma ferramenta de autoria genérica, para todas as aplicações seria uma tarefa impossível e aponta como uma solução mais realística, para esse problema, a criação de ferramentas de autoria específicas para o contexto de cada tipo de ambiente.

A seguir serão apresentadas aplicações de sistemas de autoria em realidade virtual e realidade aumentada.

5.1.1 AUTORIA USANDO REALIDADE VIRTUAL

A autoria em realidade virtual pode ser aplicada para a criação de mundos virtuais. A criação de mundos virtuais possui, geralmente, dois passos, a modelagem do ambiente e o desenvolvimento de interação e comportamento de objetos virtuais.

A tarefa de modelagem consiste em criar os ambientes virtuais e os objetos que os preenchem. Já o desenvolvimento de interação e especificação do comportamento refere-se à criação da interação entre o ambiente, objetos virtuais e os usuários, além de especificar e elaborar os seus comportamentos frente a determinadas características existentes no ambiente virtual, como a presença de gravidade. A fase de elaboração das formas de interação pode requerer o auxílio de formas de programação, a fim de ampliar as possibilidades das técnicas de interação necessárias à aplicação. Assim, usuários, que não possuem conhecimentos específicos em programação, podem encontrar dificuldades nessa etapa (HENDRICKS, 2003).

Existem diversas ferramentas para a construção de ambientes virtuais, que oferecem formas gráficas de programação e minimizam exigências de conhecimentos específicos em práticas de programação aos seus usuários desenvolvedores de objetos virtuais (LEE, 2005). Alice, Flux Studio são alguns exemplos, entre as inúmeras ferramentas existentes para a construção de aplicações de realidade virtual. WorldViz, Eon e Virtools oferecem opções comerciais avançadas para a modelagem e animação de objetos virtuais. A

seguir, serão apresentadas descrições de algumas ferramentas de construção e animação de objetos virtuais.

Alice

O Alice é um sistema de prototipagem rápida utilizado para a construção de cenas animadas em 3D. Não consiste em uma ferramenta CAD (Computer Aided Design) para a construção de objetos geométricos, mas é utilizado para descrever o comportamento e a interação de objetos virtuais. O Alice utiliza Python como linguagem script, permitindo a implementação de interações mais elaboradas (CONWAY, 2000; PIERCE, 1997). A Figura 5.1 demonstra o Alice em execução.

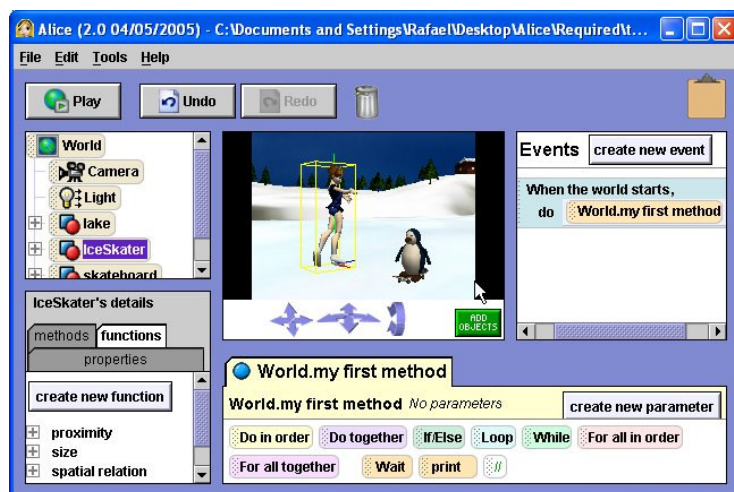


FIGURA 5.1. EXECUÇÃO DO ALICE.

Flux Studio

O Flux Studio, Figura 5.2, consiste em uma ferramenta de autoria para o desenvolvimento de conteúdo de realidade virtual. Utiliza polígonos para a modelagem de objetos, que podem ser importados e exportados para diversos formatos, como X3D, VRML, 3D Studio entre outros. Para a produção de animações e interações mais avançadas, o Flux Studio permite implementar scripts em Javascript (Media Machine, 2007).

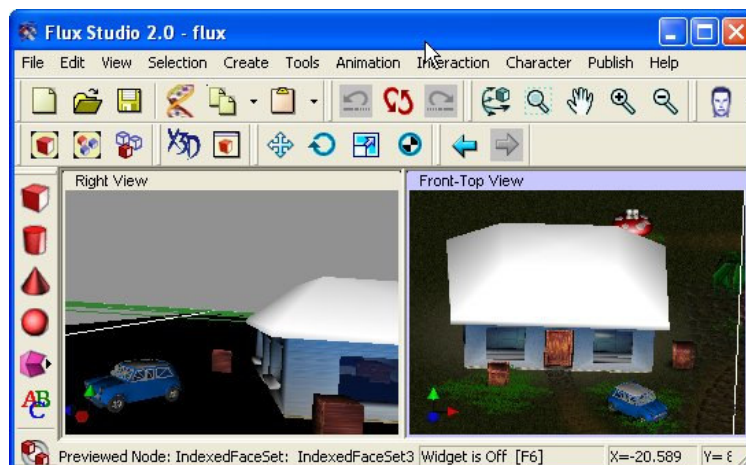


FIGURA 5.2. INTERFACE DO FLUX STUDIO.

Vizard

O Vizard da WorldViz, Figura 5.4, consiste em uma ferramenta avançada que possibilita construir conteúdos 3D interativos. Desenvolvido para a prototipagem rápida, o Vizard fornece recurso para o a implementação das mais desafiadoras aplicações. Embora o principal recurso dessa ferramenta seja o uso da linguagem de programação Python para a animação de objetos virtuais, o alto grau de abstração do Vizard permite que usuários não programadores possam desenvolver as suas aplicações (Worldviz, 2007).

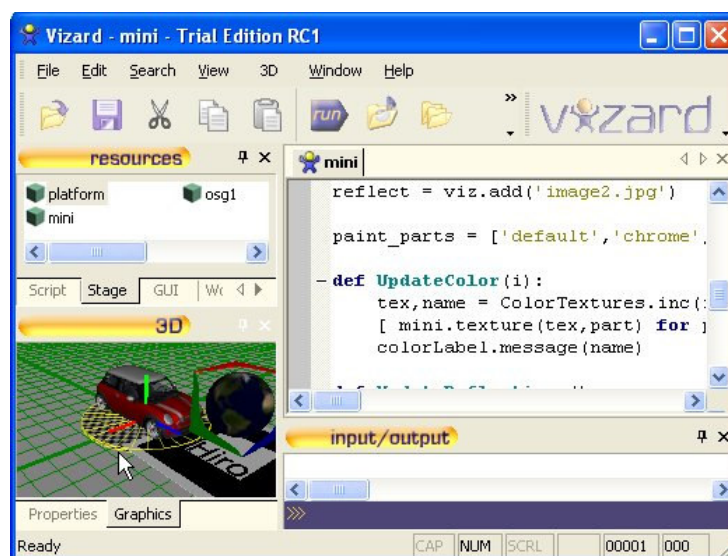


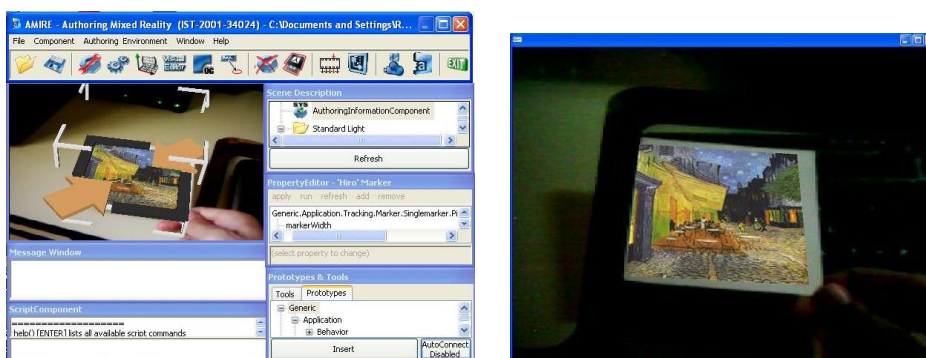
FIGURA 5.4. EXECUÇÃO DO VIZARD

5.1.2 AUTORIA USANDO REALIDADE AUMENTADA

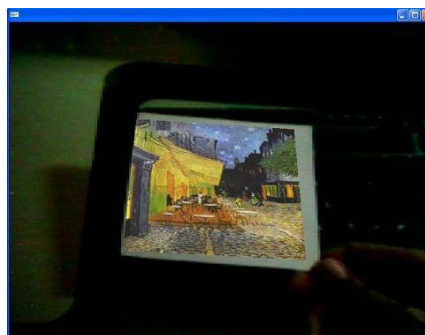
Os sistemas de autoria em realidade aumentada, similarmente, a autoria em realidade virtual, também, buscam oferecer, aos seus usuários, ferramentas não focadas na programação para a construção de suas próprias aplicações (LEDERMAN, 2004).

A construção de aplicações interativas em RA envolve dois passos principais, a sintetização de elementos gráficos e sonoros e a elaboração de formas de interação entre os usuários e o conteúdo do ambiente. Como existem avançadas ferramentas para a construção dos elementos gráficos e sonoros, os sistemas de autoria em RA assumem a estruturação da interação entre usuário e o conteúdo sintético do ambiente de RA (LEE, 2005).

O AMIRE (*Authoring Mixed Reality*) consiste em um *framework* que disponibiliza a programação visual aliada à tecnologia de orientação a componentes para a construção das aplicações de RA. Para a sobreposição dos elementos virtuais no ambiente real, o AMIRE implementa o rastreamento óptico baseado no ARToolKit. Os usuários do AMIRE desenvolvem os projetos de suas aplicações em uma interface 2D, utilizando mouse e teclado. Nessa fase de projeto, é realizada a inserção e a conexão entre os componentes e atribuições de propriedades aos objetos, como mostra a Figura 5.5a. Após a elaboração do projeto pode-se realizar o teste no ambiente de RA, de acordo com a Figura 5.5b (GRIMM, 2002; LEE, 2005).



(a)



(b)

FIGURA 5.5. EXECUÇÃO DO AMIRE: A) MODO DE EDIÇÃO, B) MODO DE VISUALIZAÇÃO NO AMBIENTE.

O *Designers Augmented Reality Toolkit* (DART) é um sistema de autoria de RA baseado no software Director da Macromedia, considerado um ambiente de autoria multimídia. O Director utiliza uma interface 2D, que oferece recursos de *timeline*, suporte a animação e uso de *scripts* para atender as necessidades de processos mais complexos. O DART, também, utiliza o rastreamento óptico baseado no ARToolKit (MACINTYRE, 2003; LEDERMAN, 2004)

Ao contrário do DART e AMIRE, existem sistemas de autoria que possibilitam a seus usuários realizarem o desenvolvimento do trabalho, sem a necessidade de alternar entre os ambientes 2D de edição e o 3D de exibição da aplicação de RA. Dessa maneira, com a atenção do usuário voltada apenas a um tipo de interface, aumenta-se a presença (WITMER, 1998; YANG 2001). O TILES e o *immersive authoring for Tangible Augmented Reality* (iaTAR) são exemplos desses sistemas e ambos utilizam o rastreamento óptico oferecido pelo ARToolKit.

O TILES consiste em um ambiente simples de RA que fornece uma interface de uso fácil e ágil para a autoria de ambientes de RA. A interação no ambiente é realizada por um conjunto de ferramentas, que permite os usuários realizarem interações dinâmicas sobre objetos virtuais, em qualquer posição do ambiente de trabalho 3D. As interações que os usuários podem realizar são: adição, remoção, duplicação, copia e visualização comentários de objetos virtuais. Para realizar as interações, os usuários devem aproximar o marcador de operação do marcador de dados a uma distância inferior a 15% de seus tamanhos, obtendo-se assim o resultado da combinação. A Figura 5.6 mostra a definição das operações realizadas no sistema de autoria TILES (POUPYREV, 2001).





























Operation	Result
Menu operations	
 +  = 	
Clipboard operations	
 +  = 	
 +  = 	
 +  = 	
Trashcan operations	
 +  = 	
 +  = Not defined	
 +  = 	
Help operations	
 +  = 	
 +  = 	
 +  = Not defined	

FIGURA 5.6. DEFINIÇÃO DAS OPERAÇÕES REALIZADAS NO TILES (POUPYREV, 2001).

O iaTAR(Immersive Authoring for Tangible Augmented Reality Applications) é um sistema baseado em componentes que permite estruturar a interação entre três principais componentes: objetos virtuais, objetos reais e a caixa de atributos (lógicos/comportamentais). Cada componente possui um conjunto de características que podem ser visualizadas, selecionadas, modificadas e conectadas a outros componentes, para a construção da aplicação de RA (LEE, 2005). A Figura 5.7 mostra a forma de interação utilizada no iaTAR para reduzir a escala do objeto virtual.

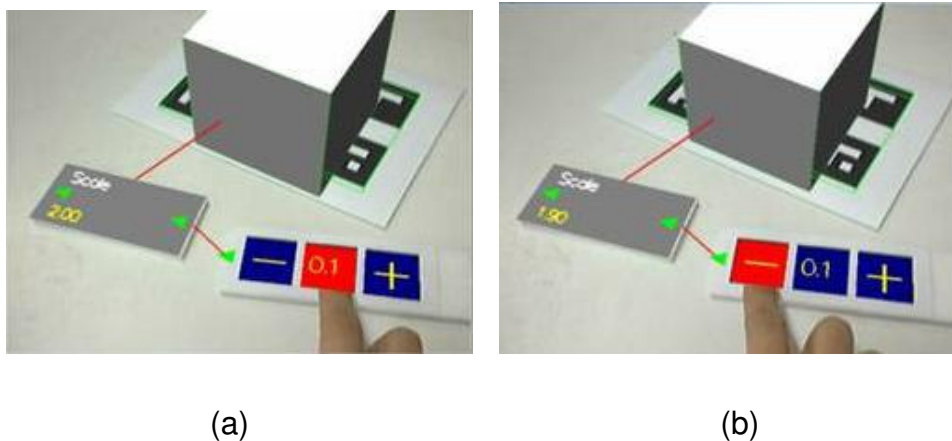


FIGURA 5.7. REDUÇÃO DO VALOR DA ESCALA DO OBJETO VIRTUAL: A) ESCALA 2.0 E B) ESCALA 1.90 (POSTECH, 2007).

5.2 FORMAS DE INTERAÇÃO .

O mouse, teclado e monitor são os dispositivos mais amplamente utilizados para realizar a interação homem-computador. Esses são dispositivos padrões das interfaces tradicionais WIMP (*Windows, Icons, Menus, Pointers*), que consistem no mais sucedido paradigma de interface de usuário já produzido (BOWMAN, 2001; BILLINGHURST, 2003).

As técnicas WIMP são implementadas na maioria das aplicações computacionais desenvolvidas, inclusive em aplicações de realidade virtual ou em algumas fases de sistemas de realidade aumentada, como o modo de edição das ferramentas de autoria DART e AMIRE.

A interface 3D de realidade aumentada permite que os usuários, utilizando HMD, possam interagir com objetos virtuais 2D e 3D em qualquer posição do ambiente de atuação. Como em RA as imagens de objetos virtuais são sobrepostas no ambiente real, as formas de interação do usuário sobre esses elementos podem requerer o uso de dispositivos que não são comumente utilizados no mundo real. Dessa maneira, a presença do ambiente de RA acaba sendo afetada, visto que a interação descontínua força os usuários a alternarem entre os modos de interação real e virtual (POUPYREV, 2002; WITMER, 1998).

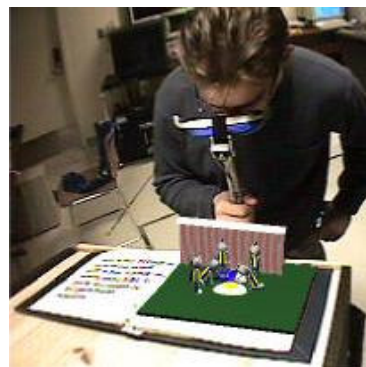
Em contrapartida, as *Tangible User Interfaces (TUI)* são extremamente intuitivas, pois utilizam objetos reais para a interação com os objetos virtuais, os quais são projetados no ambiente real. Segundo ULBRICHT (2003), as TUIs utilizam objetos reais como interfaces para o computador. Entretanto, existem alguns desafios relacionados às informações exibidas por essas interfaces. Por exemplo, a dificuldade de alterar dinamicamente as características de objetos reais, pois, geralmente, as informações visualizadas são projeções 2D sobre objetos reais ou superfícies aumentadas. Dessa maneira, o ambiente de atuação da interface é limitado à superfície de projeção, gerando a descontinuidade do espaço. Assim, fora desse espaço, os usuários não podem manipular os objetos virtuais no ambiente real, como se fossem objetos reais (BILLINGHURST, 2005; POUPYREV, 2002).

A solução para as limitações presentes nas interfaces de RA e TUI é oferecida pela *Tangible Augmented Reality (RA tangível)*. A RA tangível consiste na combinação entre as técnicas de visualização de realidade aumentada e formas de interação físicas, oferecidas pelas TUIs. Nesse caso, os usuários podem interagir intuitivamente com as técnicas 3D, visto que a manipulação dos objetos virtuais é realizada da mesma maneira como é promovida a interação com os objetos reais, não necessitando o aprendizado de técnicas de interação 3D. A RA tangível, desse modo, potencializa a visualização em conjunto com a interação, permitindo que os usuários manipulem os objetos virtuais de maneira altamente intuitiva (BILLINGHURST, 2002; ULBRICHT, 2002).

O ARToolKit e o Studierstube são exemplos de sistemas que oferecem suporte ao desenvolvimento de interfaces tangíveis de RA (ULBRICHT, 2002). Aplicações desenvolvidas com o ARToolKit, por exemplo, permitem que os usuários utilizem as próprias mãos para interagir com as referências de objetos virtuais, que são os marcadores. A Figura 5.8 mostra dois exemplos de aplicações da RA tangível, a ferramenta de autoria TILES e o sistema que possibilita visualizar e interagir com objetos virtuais sobrepostos às páginas de um livro real, denominado Magic Book (BILLINGHURST, 2001).



(a)



(b)

FIGURA 5.8. EXEMPLOS DE TANGIBLE AR: A) SISTEMAS DE AUTORIA TILES (POUPYREV, 2002) E B) EXECUÇÃO DO MAGIC BOOK (BILLINGHURST, 2001)

A interface tangível de RA permite não só alterar as formas habituais de interação com o computador, mas, também, oferecer nativamente a possibilidade de colaboração face a face. Os usuários podem compartilhar a visualização e manipular objetos virtuais de maneira natural, semelhante aos objetos físicos, devido à integração em tempo real da imagem dos elementos computacionais a imagem do ambiente real.

Assim, a RA tangível pode ser aplicada como ferramenta de apoio ao ensino, aprendizagem e treinamento (KIRNER, 2004a).

5.3 CLASSIFICAÇÃO DAS INTERAÇÕES DOS USUÁRIOS EM AMBIENTES DE REALIDADE AUMENTADA.

Os marcadores são um meio de interação utilizado em alguns ambientes de RA, como os que são desenvolvidos com o ARToolKit. O usuário manipula o marcador com a finalidade de promover a execução de ações sobre os objetos virtuais exibidos na cena. Por exemplo, na aplicação simpleVRML, o marcador tem a funcionalidade de alterar determinados atributos dos objetos virtuais que o sobrepõe, como a sua visibilidade, translação, rotação e escala. As funcionalidades do marcador podem ser alteradas, de modo a possibilitar a execução de outras formas de atuação do usuário sobre os objetos virtuais.

O esclarecimento dessas outras formas de atuação no ambiente de RA aponta para a elaboração de uma definição geral sobre os tipos de interação existentes entre o usuário e os objetos virtuais. Nesse sentido, a experiência de alguns anos em trabalhos com realidade aumentada, juntamente com os conceitos de Bowman (BOWMAN, 2001) sobre a interação em ambientes de realidade virtual, possibilitaram uma adequação dos conceitos para o ambiente de RA, culminando numa definição que envolve de maneira geral esses tipos de interações.

A definição das possíveis formas de interação do usuário no ambiente de realidade aumentada envolve os aspectos relacionados à presença dos objetos, mudança de características, manipulação e comportamento.

5.3.1 PRESENÇA DOS OBJETOS

A presença do objeto virtual é o mecanismo de interação que permite ao usuário controlar a existência do objeto na cena. Por exemplo, o objeto pode ser inserido, copiado, trocado, retirado e até mesmo apagado da cena. A seguir, serão apresentados os conceitos dos mecanismos de controle da presença dos objetos.

Inserção – A inserção representa as possibilidades do usuário introduzir um objeto virtual na cena. Dessa maneira, a inserção abrange as formas de interações que tenham a função de realizar a exibição dos objetos na imagem do ambiente.

Cópia – A cópia consiste no recurso que oferece a possibilidade do usuário replicar os objetos existentes na cena. Esse mecanismo permite a possibilidade de estruturar um mundo virtual, a partir da disponibilização de alguns elementos virtuais.

Troca – A troca permite ao usuário alterar um objeto visualizado por outro. Esse mecanismo possibilita minimizar a quantidade dos dispositivos de rastreamento, como os marcadores, uma vez que a realização

da troca busca promover o agrupamento das referências entre os objetos virtuais sobre o mesmo marcador.

Retirada – O mecanismo de retirada possibilita controlar a quantidade de objetos existentes na cena, através da exibição ou não de determinados objetos. Assim, é possível adotar esse mecanismo para impor determinados limites durante a visualização do objeto, de modo que a cena não se torne saturada com a exibição de uma grande quantidade de objetos.

Apagamento – O apagamento permite ao usuário desalocar determinados objetos carregados no ambiente, promovendo sua eliminação. Esse mecanismo libera os recursos computacionais, possibilitando que novos objetos possam ser carregados e exibidos.

5.3.2 MUDANÇAS DE CARACTERÍSTICAS

A mudança de características dos objetos virtuais consiste na modificação dos atributos relacionados à aparência desses objetos, como tamanho, cor, luminosidade, transparência, etc. Nesse caso, além de ser possível oferecer a possibilidade do usuário interferir diretamente na aparência dos objetos virtuais, torna-se viável utilizar esse recurso como um recurso de *feedback* à aplicação. Por exemplo, o objeto pode alterar suas propriedades para chamar a atenção do usuário.

A flexibilidade da modificação, nesse tipo de interação, permite explorar operações dinâmicas capazes de oferecer mecanismos potencialmente atrativos à aplicação.

5.3.3 MANIPULAÇÃO

A manipulação consiste no recurso de interação, que permite ao usuário transladar ou rotacionar partes ou até mesmo a cena toda, utilizando os dispositivos de interação do ambiente.

A manipulação possibilita ao usuário explorar outros pontos de vista e articular a distribuição dos objetos na cena, reorganizando-os. A

facilidade e objetividade no uso desse recurso são aspectos fundamentais para a motivação da interação do usuário com o ambiente de RA.

5.3.4 COMPORTAMENTO

O comportamento consiste numa forma do usuário atuar na dinâmica do objeto virtual. Nesse caso, os objetos podem oferecer respostas diferenciadas, tendo em vista os as ações executadas pelo usuário.

6 SISTEMA DE AUTORIA COLABORATIVA COM REALIDADE AUMENTADA-SACRA.

As ferramentas de autoria são desenvolvidas para facilitar e agilizar o trabalho de usuários de computadores, permitindo que usuários, sem conhecimentos específicos em programação, encontrem a liberdade para o desenvolvimento das suas próprias aplicações. Essas ferramentas, geralmente, utilizam interfaces gráficas para associar operações computacionais a elementos visuais, visto que os apelos visuais são estimulantes e reforçam a percepção do usuário.

O uso da interface tangível de RA em autoria permite potencializar o incremento da percepção dos usuários, visto que utilizam a intuição natural para a manipulação de objetos virtuais, que povoam próprio ambiente real onde estão acostumados a trabalhar.

Além das ferramentas de autoria que facilitam e agilizam os trabalhos de seus usuários, existem as ferramentas colaborativas que viabilizam o compartilhamento do trabalho entre seus usuários. As ferramentas colaborativas oferecem técnicas de cooperação, coordenação e comunicação para que participantes possam contribuir de maneira responsável no desenvolvimento de um trabalho em comum.

Esse capítulo aborda a elaboração de um sistema de autoria colaborativa com Realidade Aumentada, que será referenciado no transcorrer do capítulo como SACRA. O sistema desenvolvido oferece autoria colaborativa para a construção de mundos virtuais, utilizando a interface tangível de Realidade Aumentada.

6.1 CARACTERIZAÇÃO DO AMBIENTE.

O SACRA oferece um ambiente de RA tangível, para a autoria de mundos virtuais, possibilitando a colaboração face a face e remota.

O seu desenvolvimento foi realizado com auxílio da biblioteca ARToolKit, devido a características como facilidade de programação e utilização de dispositivos de baixo custo, usando basicamente uma webcam e um computador. Embora o Studierstube forneça suporte a colaboração remota, recurso que o ARToolKit não disponibiliza, o rastreamento oferecido com o custo mais acessível, dentre os vários suportados pelo OpenTracker, consiste no próprio rastreamento óptico do ARToolKit. Assim, devido à facilidade de programação, o ARToolKit foi a biblioteca adotada para o desenvolvimento do trabalho.

A interação do usuário no ambiente do SACRA é realizada através do uso de marcadores, caracterizando a interface tangível de RA. Por esse motivo, um ponto relevante adotado pelo sistema é a manutenção da presença do usuário no ambiente de RA. Assim, buscou-se minimizar a descontinuidade da sensação de presença no ambiente, ocasionada pelas mudanças entre tipos de interfaces e dispositivos utilizados.

O SACRA disponibiliza a seus usuários técnicas de interação, a partir de operações envolvendo os propriedades dos marcadores como visibilidade, posição e orientação. Essas operações permitem aos usuários interagirem no ambiente, através do controle da presença, mudança de características e manipulação dos objetos virtuais.

A autoria no SACRA oferece suporte a colaboração, permitindo que outros usuários possam contribuir na construção de mundos virtuais. A visualização do ambiente em modo de projeção, ou através de monitores, possibilita explorar a colaboração face a face, enquanto a autoria remota necessitou ser implementada.

O trabalho colaborativo remoto é viável no ambiente, através do emprego de técnicas de cooperação, coordenação e comunicação.

As técnicas de cooperação permitem que múltiplos usuários trabalhem na elaboração de mundos virtuais em comum. Dessa maneira, é disponível um espaço compartilhado entre os usuários, para a elaboração da

autoria colaborativa. O espaço compartilhado emprega técnicas de coordenação, que restringem as ações dos usuários sobre determinados objetos virtuais, possibilitando controlar o desenvolvimento das tarefas.

A troca de informações entre os participantes é fundamental para a coordenação do trabalho colaborativo. Dessa maneira, é necessário utilizar as tecnologias de comunicação existentes na atualidade, como os chats ou sistemas vídeo-conferência.

Com esses requisitos, o SACRA possibilita que os usuários realizem as seguintes formas de interação: face a face, assíncrona, síncrona distribuída e assíncrona distribuída.

A interação face a face ocorre no mesmo local e ao mesmo tempo; é caso da colaboração face a face. A interação assíncrona é realizada no mesmo local, mas em tempos diferentes; nesse caso técnicas para salvar e recuperar os trabalhos foram desenvolvidas. Na interação síncrona distribuída, usuários remotos realizam o trabalho colaborativo em tempo real. Para ocorrer a interação assíncrona distribuída, o usuário pode enviar ou receber por email o trabalho de autoria desenvolvido, independentemente de tempo e local para a elaboração do trabalho.

Assim, o SACRA possui variadas e concisas formas de interação a seus usuários, a fim de oferecer suporte a diversas aplicações educacionais, possibilitando a exploração do potencial da interface de RA aliada aos benefícios do trabalho colaborativo.

6.2 DESENVOLVIMENTO

O Sacra foi desenvolvido com base na biblioteca ARTToolKit, nas versões 2.65 e 2.72.1 ambas com suporte a VRML, e nos módulos do NetARTToolKit, para a interação remota entre os usuários.

6.2.1 NETARTOOLKIT

O NetARToolKit consiste numa modificação do ARToolKit 2.65 com VRML, que disponibiliza serviços remotos de interação com o renderizador, possibilitando a implementação de aplicações que promovam modificações nas características das, pré-carregadas, cenas escritas na linguagem VRML (OLIVEIRA, 2006).

Com o uso de funcionalidades da biblioteca LibVRML97, essa aplicação permite alterar, características de determinada parte do objeto virtual, sem a necessidade de se aplicar uma nova leitura dos objetos VRML, como mostrado em (SANTIN, 2005). Isto é possível com a atuação direta no renderizador, através da execução de comandos específicos, que são recebidos remotamente. A Figura 6.1 demonstra a interação entre as camadas do sistema.

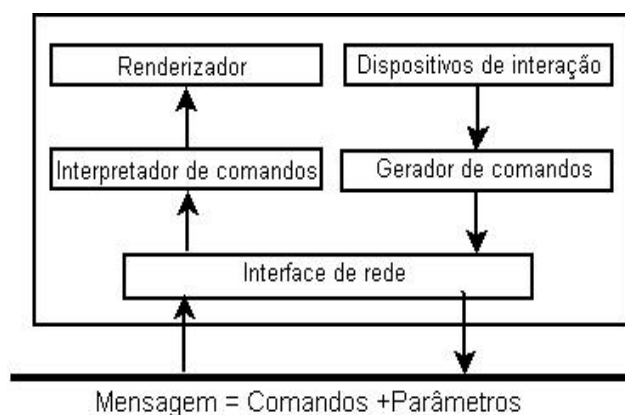


FIGURA 6.1. INTERAÇÃO ENTRE AS CAMADAS DO SISTEMA.

A arquitetura de rede implementada no NetARToolKit consiste no modelo peer-to-peer, sendo, nesse caso, cada aplicação um cliente e servidor ao mesmo tempo. Dessa maneira, os recursos oferecidos pela aplicação geram as mensagens que são enviadas para todas as outras aplicações interconectadas.

As mensagens, nesse caso, podem ser geradas tanto pela aplicação de RA, como também por uma outra aplicação externa. A criação da mensagem é realizada pela classe chamada ARNetEAI, ou por outra aplicação

que implemente a funcionalidade para gerar os comandos e parâmetros necessários, interagindo, dessa maneira, com a aplicação de realidade aumentada.

A classe ARNetEAI, além de gerar os comandos e passá-los para a interface de rede, que no caso é representada pela classe ARServer, também possui o papel fundamental de interpretar os comandos provindos da rede e executá-los. A execução do comando atua no grafo de cena, a partir das funções oferecidas pela biblioteca OpenVRML. A seguir, a Tabela 6.1 mostra alguns comandos implementados pelo sistema.

TABELA 6.1. COMANDOS IMPLEMENTADOS.

Comando	Função sobre o “nodo”
set_translation	Transladar
set_rotation	Rotacionar
set_diffuseColor	Alterar a cor

A Figura 6.2 mostra um exemplo de funcionamento do ambiente, onde uma aplicação de realidade aumentada recebe mensagens de um software desenvolvido em Java, além de enviar mensagens a outras aplicações de RA.

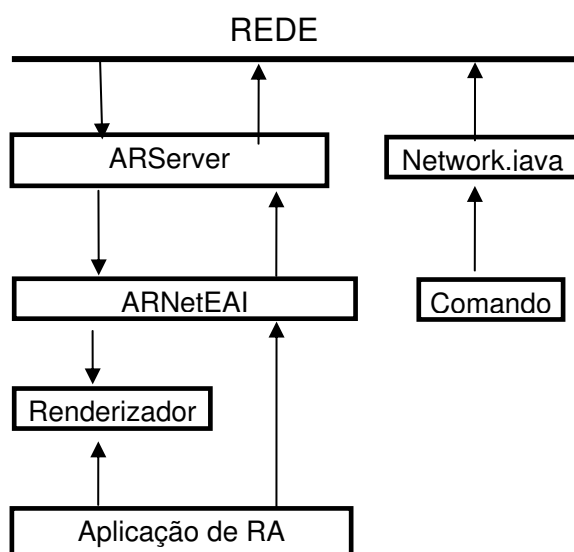


FIGURA 6.2. ESTRUTURA DE FUNCIONAMENTO DO AMBIENTE.

A utilização do NetARToolkit facilitou o desenvolvimento do sistema, além de expandir a interação com os objetos virtuais, dada a possibilidade de modificação diretamente no seu grafo de cena.

6.2.2 IMPLEMENTAÇÃO DO SACRA

O desenvolvimento do sistema de autoria colaborativa com Realidade Aumentada utilizou a linguagem C, envolvendo modificações no projeto simpleVRML do ARToolKit2.65vrm1, através da inserção de funções para a autoria, acoplamento de módulos de rede e serviços do netARToolkit, troca dos módulos de vídeo da versão 2.65 pelos módulos da versão 2.72.1 e alterações nas bibliotecas libarvrm1.lib e libvrm197core.lib. A Figura 6.3 demonstra a estrutura do sistema desenvolvido.

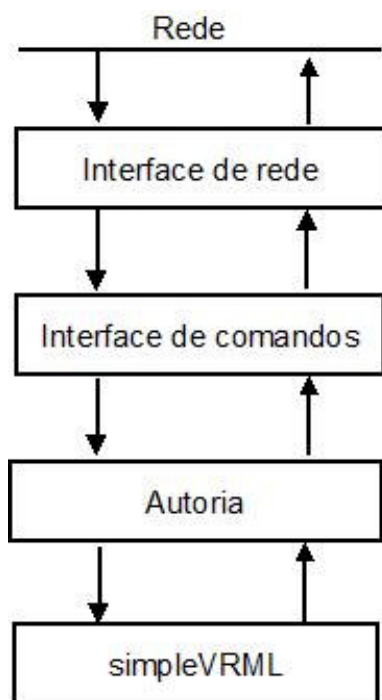


FIGURA 6.3. ESTRUTURA DO SACRA.

O módulo de autoria é responsável pela inicialização do interpretador de comandos, além de adicionar as funcionalidades aos marcadores no Sacra. No caso desse trabalho, o comportamento de determinados marcadores é utilizado para realizar a interação com os objetos virtuais.

O comportamento dos marcadores está associado aos possíveis estados que o sistema de rastreamento permite identificar. Por exemplo, é possível identificar: a presença do marcador na cena; a distância do marcador, em relação a outros marcadores ou objetos virtuais; e a orientação do marcador e o seu ângulo de rotação. Esses comportamentos do marcador podem ser utilizados em conjunto para expandir o tipo de interação, como aliar a detecção da distância e rotação.

As variações e, possivelmente, a combinação dos estados referentes ao comportamento dos marcadores possibilitaram a criação de uma nova categoria de marcadores, que foram denominados marcadores de ações.

Os marcadores de ações viabilizam o acesso a formas de interação dos objetos virtuais, pois são capazes de exercer meios fundamentais para a execução das operações sobre os elementos virtuais. Essas operações são conhecidas como seleção, ação e liberação.

Um marcador pode ter a funcionalidade elaborada para explorar as três operações em conjunto, ou essas operações podem ser executadas por diferentes tipos de marcadores de ações, dependendo da necessidade da sua atuação. Por exemplo, a cópia de um objeto exibido na cena poderia ser realizada por um marcador que selecione, copie e libere o objeto na cena. Outra forma seria a utilização de um marcador para cada uma das três etapas, ou seja, um marcador seria responsável pela seleção, outro pela cópia e o terceiro teria a função de liberar o objeto.

- seleção: Para realizar alguma interação com o objeto virtual, primeiramente, o usuário necessita selecionar qual será o objeto alvo da ação. Uma forma para selecionar o objeto virtual consiste na verificação da proximidade entre o marcador de ação e os objetos disponíveis na cena.
- ação: A ação a ser efetuada sobre o objeto virtual é desempenhada através dos marcadores de ação. A função desses marcadores consiste

em oferecer os recursos necessários para o usuário realizar a tarefa desejada.

- liberação: A liberação realiza as atividades inversas a seleção. Esse mecanismo finaliza uma determinada atuação do usuário sobre o objeto virtual, possibilitando que o usuário execute novas tarefas no ambiente.

As funcionalidades dos marcadores de ações são definidas por constantes de identificação implementadas no código fonte do módulo de autoria. Os números de identificação estão relacionados à ordenação do cadastramento dos marcadores no arquivo “Data/vrml_data.dat” do ARToolKit. A Figura 6.4 demonstra a definição das funcionalidades dos marcadores.

#define	INSPECTOR	0
#define	CONTROL	1
#define	COPY	2
#define	TRANSPORT	3
#define	ERASER	4
#define	STATUS	5
#define	PATH	6
#define	LOCK	7

FIGURA 6.4. DEFINIÇÃO DAS FUNCIONALIDADES DOS MARCADORES

A cada iteração são realizadas comparações a fim de verificar se o marcador, que está na cena, é de ação e caso seja executá-la. A Figura 6.5 exemplifica a estrutura principal do módulo de autoria.


```

Enquanto Sacra estiver ativado
  Teste múltiplo:
    se object[INSPECTOR].visible==true então
      se isNear(INSPECTOR)==true então
        changePersistentStatus()

    se object[CONTROL].visible==true então
      se isNear(CONTROL)==true então
        changeObjectList()

    se object[COPY].visible==true então
      se isNear(COPY)==true
        refwasSelect==true então
          copy()

    se object[TRANSPORT].visible==true então
      se isNear(TRANSPORT)==true então
        transport()

    se object[ERASER].visible==true então
      se isNear(ERASER)==true então
        freeObject()

    se object[STATUS].visible==true então
      showStatus()

    se object[PATH].visible==true então
      se createPath==true então
        path()

    se object[Lock].visible==true então
      se isNear(Lock)==true então
        changeRemoteLockStatus()
  Fim-Teste múltiplo
Fim-enquanto

```

FIGURA 6.5. ESTRUTURA PRINCIPAL DA AUTORIA

No Sacra, os marcadores que não possuem ações definidas, ou seja, os que foram cadastrados a partir do oitavo marcador podem assumir o papel de marcador de referência, que será denominado no texto como REF

O REF permite que o usuário interaja com pontos no espaço cadastrados no sistema. Cada REF pode ter diversos pontos, que referenciam

um ou mais objetos virtuais. A Figura 6.6 mostra a estrutura das referências no SACRA.

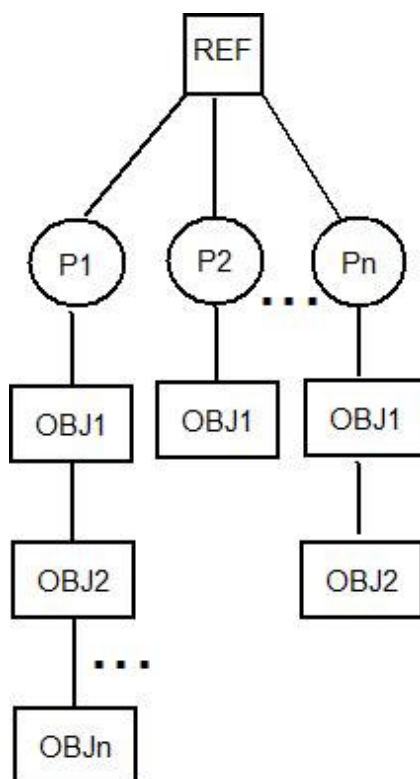


FIGURA 6.6. ESTRUTURA DO MARCADOR REF NO SACRA.

O relacionamento entre a estrutura de dados do SACRA é demonstrado na Figura 6.7. A estrutura *Object* é associada a todos os marcadores cadastrados no sistema. Caso o marcador seja REF a estrutura *reference* é atribuída a esse marcador. Os pontos associados a cada REF são representados pela estrutura *position*. A classe *arVrml97Viewer* possibilita o acesso dos atributos dos objetos WRLs carregados no sistema, que no caso são referenciados por marcadores ou pontos.

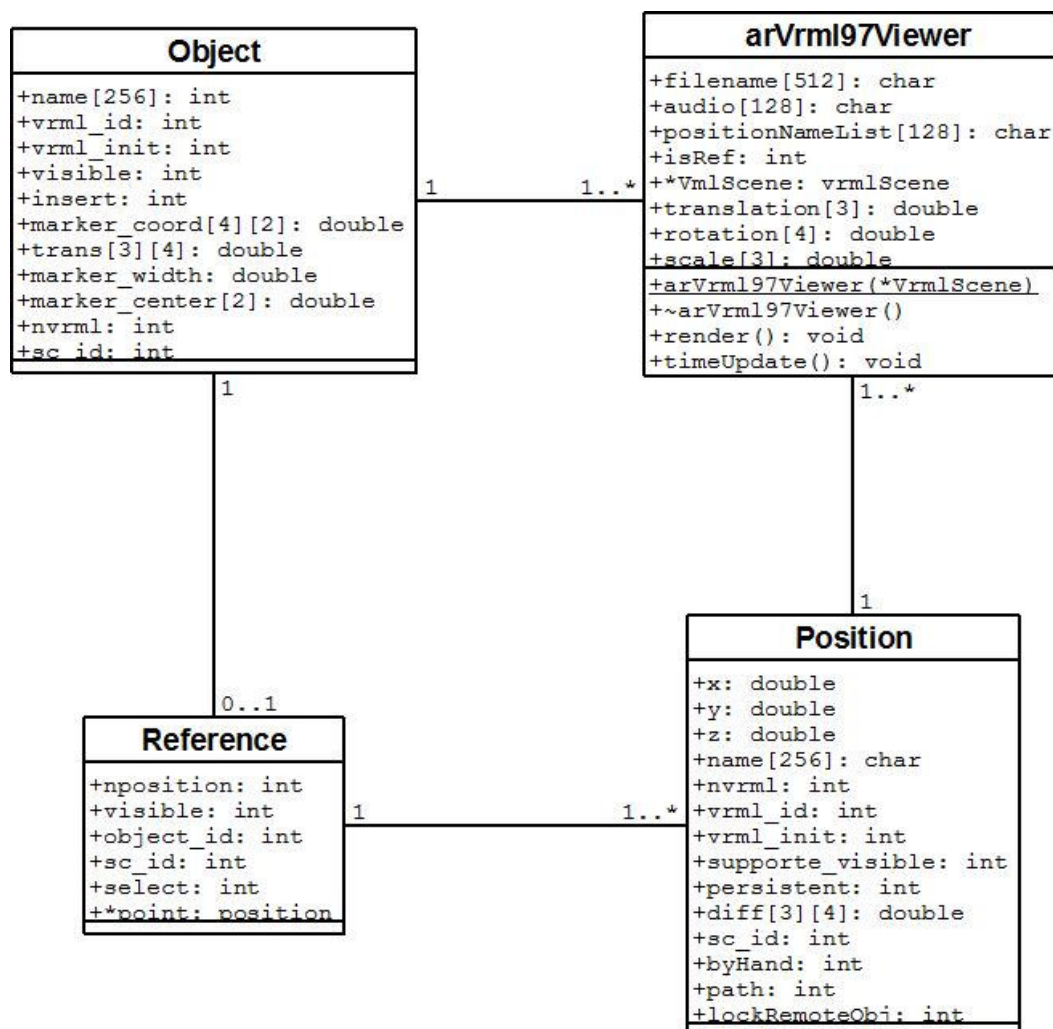
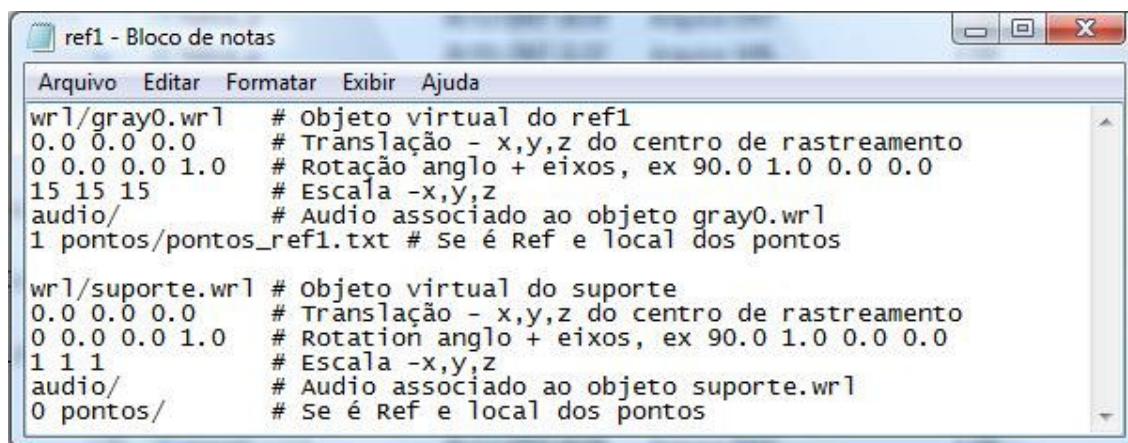


FIGURA 6.7. RELACIONAMENTO ENTRE AS ESTRUTURAS DE DADOS NO SACRA.

Os marcadores são atribuídos como REF através de dois parâmetros presentes nos arquivos de extensão dat contidos na pasta WRL para a associação de objetos virtuais. Esses parâmetros indicam se o marcador é REF e, caso seja, indicam o nome do arquivo que contém os pontos associados a esse marcador. A Figura 6.8 mostra o exemplo do arquivo responsável pela associação de objetos virtuais ao marcador, contendo os atributos relacionados a esses objetos, como a translação, rotação, escala e o áudio. A última linha indica se o marcador referente a determinado objeto virtual é ou não REF, primeiro parâmetro, e o nome do arquivo que contém os pontos atrelados a essa referência, segundo parâmetro.



```

Arquivo  Editar  Formatar  Exibir  Ajuda
wrl/gray0.wrl # Objeto virtual do ref1
0.0 0.0 0.0 # Translação - x,y,z do centro de rastreamento
0 0.0 0.0 1.0 # Rotação angulo + eixos, ex 90.0 1.0 0.0 0.0
15 15 15 # Escala -x,y,z
audio/ # Audio associado ao objeto gray0.wrl
1 pontos/pontos_ref1.txt # Se é Ref e local dos pontos

wrl/suporte.wrl # Objeto virtual do suporte
0.0 0.0 0.0 # Translação - x,y,z do centro de rastreamento
0 0.0 0.0 1.0 # Rotation angulo + eixos, ex 90.0 1.0 0.0 0.0
1 1 1 # Escala -x,y,z
audio/ # Audio associado ao objeto suporte.wrl
0 pontos/ # Se é Ref e local dos pontos

```

FIGURA 6.8. ARQUIVO DE ASSOCIAÇÃO DE OBJETOS VIRTUAIS AO MARCADOR.

No SACRA, o primeiro REF cadastrado obtém a funcionalidade de referência remota. Esse marcador consiste na referência compartilhada entre os usuários remotos. Toda a ação realizada nessa REF será enviada para as REFs colaborativas dos usuários remotos. Assim, os usuários passam a usufruir de uma área compartilhada em comum entre os integrantes do ambiente.

A implementação do REF permitiu estender a capacidade de interação do sistema, visto que o referencial de objetos virtuais no mundo real deixou de ser exclusivamente o marcador, passando a mesclar com pontos cadastrados.

6.3 FUNCIONALIDADES DO SACRA.

A seguir, serão demonstradas as principais funcionalidades implementadas no SACRA.

6.3.1 CADASTRAMENTO DE PONTOS

Os pontos cadastrados no Sacra são, basicamente, posições extraídas das transformações relativas entre um determinado REF, visível na cena, e o marcador de inspeção, denominado “INSPECTOR”.

A transformação relativa entre os marcadores é calculada pela mudança de base da matriz transformação do marcador INSPECTOR no

sistema de coordenadas da câmera, para o sistema de coordenadas do marcador REF. A equação 6.1 demonstra como é realizada a mudança do sistema de coordenadas da câmera para o sistema de coordenadas do marcador. Para isso, é realizada a multiplicação da matriz inversa da transformação de REF pela matriz transformação do marcador INSPECTOR, obtendo-se a matriz $M_{3 \times 4}$ que representa a transformação do marcador INSPECTOR no sistema de coordenadas do marcador REF.

$$M = object[id_ref]trans^{-1} \cdot object[INSPECTOR]trans \quad (6.1)$$

O cadastramento é responsável por armazenar em arquivo o conteúdo na matriz M, mais especificamente a posição e orientação do marcador INSPECTOR, em relação ao REF. Existem duas maneiras para realizar o cadastro de pontos no SACRA. A primeira possibilidade é posicionar o marcador INSPECTOR na posição desejada, em relação a determinada REF, e clicar com o botão esquerdo do mouse. Nesse momento, é pedido o nome do arquivo (.dat) que deverá ser associado ao ponto cadastrado.

A segunda possibilidade consiste no próprio usuário editar o arquivo que contém as posições associadas à referência, adicionando valores de X, Y e Z em milímetros da distância do ponto desejado até o REF e o nome do arquivo representante dos objetos virtuais, que serão visualizados na posição. Os valores da orientação dos objetos virtuais, a serem exibidos nesse ponto, assumem valores default, que podem ser alterados com a modificação da orientação do objeto através do uso do marcador de transporte.

Durante o cadastramento, todos os pontos cadastrados na REF remota são enviados para as REFS remotas dos usuários remotos, permitindo que esses usuários, geograficamente separados, visualizem os mesmos objetos nas mesmas posições, desde que possuam os determinados objetos virtuais com os mesmos.

6.3.2 TÉCNICAS DE INTERAÇÃO

Os marcadores de ações são responsáveis pela interação do usuário no ambiente do SACRA. A Figura 6.9 mostra um exemplo dos marcadores de ação e um marcador de referência com seus respectivos pontos e objetos virtuais associados.

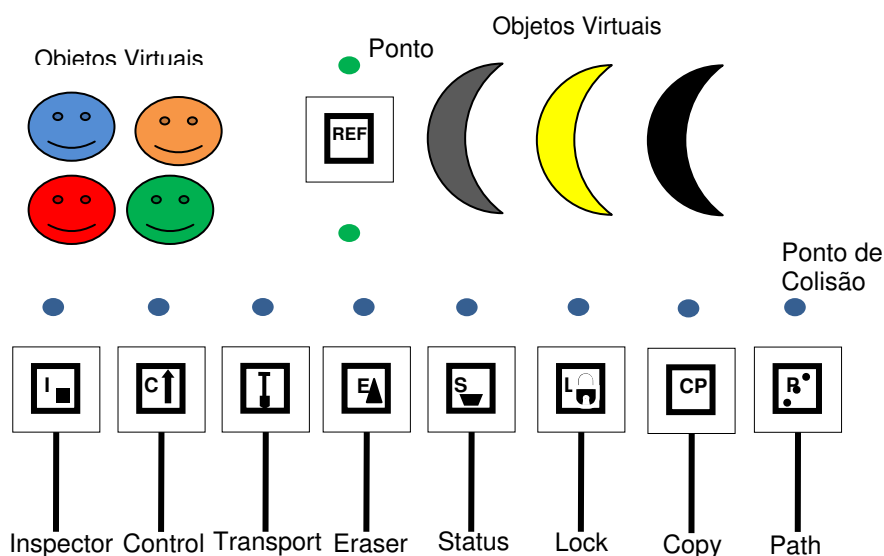


FIGURA 6.9. EXEMPLO DOS MARCADORES DE AÇÕES E REFERÊNCIA.

Os marcadores de ações atuam nos pontos associados às referências, através de um ponto denominado ponto de colisão, que pode ser configurado para coincidir com o centro do marcador, ou ser deslocado, como os círculos em azul acima dos marcadores de ações na Figura 6.9.

A atuação desses marcadores de ações é realizada por meio da comparação da distância entre os pontos associados ao REF e os pontos de colisão dos marcadores de ações. Para isso, foi estipulada a distância de colisão dos marcadores de ações. Para isso, foi estipulada a distância de colisão, que pode ser alterada, durante a execução, para o marcador de ação interagir com determinado ponto da REF. O exemplo da Figura 6.10 mostra uma ilustração das variações que o raio de atuação da distância de colisão pode assumir. Nesse caso, o marcador de ação atuaria em pontos nos interior da circunferência de raio r .

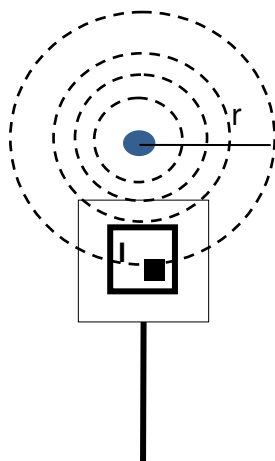


FIGURA 6.10. DISTÂNCIA PARA A ATUAÇÃO DO MARCADOR DE AÇÃO.

6.3.3 FUNCIONALIDADES DOS MARCADORES DE AÇÕES

6.3.3.1 INSPECTOR

O marcador denominado INSPECTOR é responsável pela inspeção dos pontos associados aos marcadores de referência, além de ser utilizado para realizar o cadastramento de novos pontos.

A inspeção de pontos consiste no mecanismo de controle da presença de objetos virtuais na cena, permitindo o usuário ativar ou desativar a persistência dos objetos virtuais associados aos pontos.

Ao desativar a persistência dos objetos virtuais o usuário deixa de visualizar o objeto, exigindo que ele memorize a posição daquele determinado ponto, a fim de ativá-lo futuramente. Porém, uma grande quantidade de objetos desativados pode dificultar o trabalho do usuário, visto que necessitaria uma morosa inspeção no ambiente, para encontrar o objeto desejado. Para isso, foi desenvolvido um mecanismo opcional de suporte que, no momento de desativação da persistência do ponto, é inserida uma referência visual em sua posição. Essa referência visual pode ser um objeto VRML.

6.3.3.2 CONTROL

O CONTROL consiste no marcador que permite a troca dos objetos virtuais, caso exista mais de um objeto associado a um ponto. Esse marcador realiza a troca seqüencial dos objetos visualizados. A troca é realizada, através da inserção do controle na cena, seguida da colisão com o ponto que contém a lista de objetos. Ao chegar ao fim da lista, a próxima interação retorna ao seu início, mostrando, novamente, o primeiro objeto visualizado.

6.3.3.3 COPY

O marcador COPY permite ao usuário copiar um objeto virtual associado a determinada REF e replicá-lo na própria REF ou inserí-lo numa outra REF.

Para o usuário copiar um objeto, é necessário selecionar primeiramente a REF de destino, através da aproximação do COPY e, por conseguinte, copiar o objeto virtual desejado, por meio de colisão com o ponto representante do objeto. A fixação do objeto na posição desejada é realizada pela oclusão do marcador de copia.

A seleção do marcador de destino é visualizada pela modificação da cor do objeto virtual associado ao REF. Para alterar o estado de selecionado do REF, o usuário deve realizar uma nova aproximação entre esses marcadores.

A cópia de um objeto virtual para a REF remota promove o envio dos dados referentes ao objeto a REF remota dos usuários remotos, permitindo que todos visualizem o objeto copiado.

6.3.3.4 TRANSPORT

O TRANSPORT é o marcador responsável pela re-orientação e re-posicionamento de pontos. A colisão desse marcador com um ponto

promove o acoplamento do ponto ao TRANSPORT, possibilitando ao usuário transportar o ponto para qualquer posição em relação a um determinado REF.

Caso o marcador de referência seja remoto, o transporte enviará para os usuários as atualizações da orientação e da posição dos pontos. Para o usuário transportar um ponto atrelado a REF remota, é necessário que ele tenha inserido o determinado ponto ou que esse esteja desbloqueado.

O transporte remoto recebeu um filtro de envio de mensagens, que realiza o envio de mensagens, somente quando a movimentação do ponto supera uma distância estabelecida. Assim, o SACRA não onera a rede com o envio e recebimento de um grande número de mensagens.

6.3.3.5 ERASER

O marcador ERASER possibilita a exclusão dos pontos e a desalocação dos respectivos objetos virtuais alocados na memória do computador. Para a exclusão de determinado ponto, é necessário realizar a colisão do ERASER com o ponto desejado.

Um determinado ponto, pertencente à REF remota, pode ser excluído pelo usuário que o inseriu, ou por usuários remotos, caso o ponto esteja desbloqueado.

6.3.3.6 STATUS

O marcador STATUS mostra ao usuário as principais informações do estado do sistema. As informações são apresentadas em um painel desenvolvido em VRML, que mostra o valor das variáveis de persistência, suporte, rastro, bloqueio, distância de atuação do ponto de colisão e a distância do ponto de colisão do centro do marcador.

Como essas variáveis são alteradas, durante a execução do sistema, foi necessário utilizar mecanismos para a modificação do objeto virtual

no tempo de execução. Para isso, foram implementadas funções para atuar em determinado nó do grafo de cena do objeto virtual. A Figura 6.11 demonstra a função `changeString`, que altera o valor das Strings do nó `text` nos objetos virtuais.

```
void changeString(int id, char* node, char* str){
    VrmlScene* sc = sceneGraph[id]->vrmlScene;
    if (sc == NULL) return;

    VrmlNamespace* ns = sc->scope();
    if (ns==NULL) return;

    VrmlNode* p_no = ns->findNode(node);
    if (p_no==NULL) return;

    VrmlNodeText* nd = p_no->toText();
    if (nd==NULL) return;

    VrmlMFString mf(str);
    nd->setField("string",mf);
}
```

FIGURA 6.11. FUNÇÃO UTILIZADA PARA A MODIFICAÇÃO DA STRING NO NÓ TEXT DO PAINEL DE STATUS.

6.3.3.7 PATH

O marcador `PATH` possibilita ao usuário construir um rastro de objetos virtuais, permitindo propagar a indicação de um percurso entre as `REFs` remotas. Esse rastro consiste na fixação de objetos virtuais, durante a manipulação do marcador `PATH`. Assim, conforme o usuário movimentar o `PATH`, o percurso realizado é marcado pelos objetos associados a esse marcador.

O rastro construído oferece os recursos de controle de presença dos objetos virtuais, permitindo que o usuário ative e desative a sua visualização no ambiente local, ou então apague-o completamente, tanto localmente, quanto remotamente.

O rastro elaborado representa uma forma de comunicação no ambiente colaborativo, visto que transmite o movimento realizado pelo PATH aos usuários remotos.

6.3.3.8 Lock

A técnica de coordenação do trabalho colaborativo no SACRA é aplicada, através do marcador denominado LOCK. Esse marcador permite ao usuário bloquear ou desbloquear as operações remotas sobre os seus objetos na REF remota. A restrição sobre operações remotas sobre determinado objeto é alterada somente pelo usuário que o inseriu.

O Sacra oferece as opções de envio do objeto bloqueado ou desbloqueado às REFs remotas, pois os usuários que necessitassem do bloqueio perderiam o controle de seus objetos, caso fosse necessário realizar primeiro o envio e posteriormente o bloqueio. Os usuários remotos poderiam realizar alguma modificação nesse intervalo de tempo.

O LOCK atua sobre os pontos da REF remota, através da colisão de seu centro de colisão com o ponto determinado. Para visualizar se o ponto está bloqueado ou desbloqueado, o usuário deve utilizar o marcador STATUS, aproximando-o do ponto desejado.

6.4 AMBIENTE DE AUTORIA COLABORATIVA.

O SACRA possibilita a configuração de ambientes virtuais, através do uso das ferramentas para a interação com objetos virtuais, denominadas marcadores de ações.

Os marcadores de ações atuam sobre pontos atrelados a outros tipos de marcadores conhecidos como referência. Cada referência representa um sistema de coordenadas, que orienta o posicionamento de seus pontos. Um ponto pode ter um ou mais objetos virtuais associados, sendo a visualização desses objetos controlada pelo marcador de controle.

As referências podem ser de dois tipos: local ou remota. A referência local permite a elaboração e visualização de um ambiente virtual no ambiente local. Já a referência remota consiste numa área comum entre os usuários remotos, possibilitando a configuração e visualização de ambientes virtuais remotamente.

6.4.1 REFERÊNCIA LOCAL

O SACRA possibilita o uso de múltiplas referências locais, viabilizando diversos grupos de objetos virtuais representados por essas referências.

A REF local permite a colaboração face a face, visto a possibilidade de mais pessoas estarem trabalhando na mesma aplicação no mesmo local e ao mesmo tempo.

Os trabalhos realizados nas REFs locais são salvos em arquivos, possibilitando também, tanto a colaboração assíncrona, quanto a assíncrona distribuída.

A colaboração assíncrona é viável no SACRA, pois os arquivos salvos das REF são sempre lidos na inicialização do sistema, viabilizando o trabalho no mesmo local, mas em tempos diferentes.

A colaboração assíncrona e distribuída pode ser realizada, através do envio dos arquivos por e-mail, realizando, dessa maneira, o trabalho tanto em locais, quanto em tempos diferentes.

6.4.2 REFERÊNCIA REMOTA

A referência remota oferece uma área compartilhada entre os usuários remotos, possibilitando o desenvolvimento do trabalho colaborativo síncrono e distribuído. A Figura 6.12 demonstra a esquematização do uso da referência remota. No esquema, cada nó adiciona um objeto virtual, que é distribuído para os outros nós participantes do trabalho colaborativo.

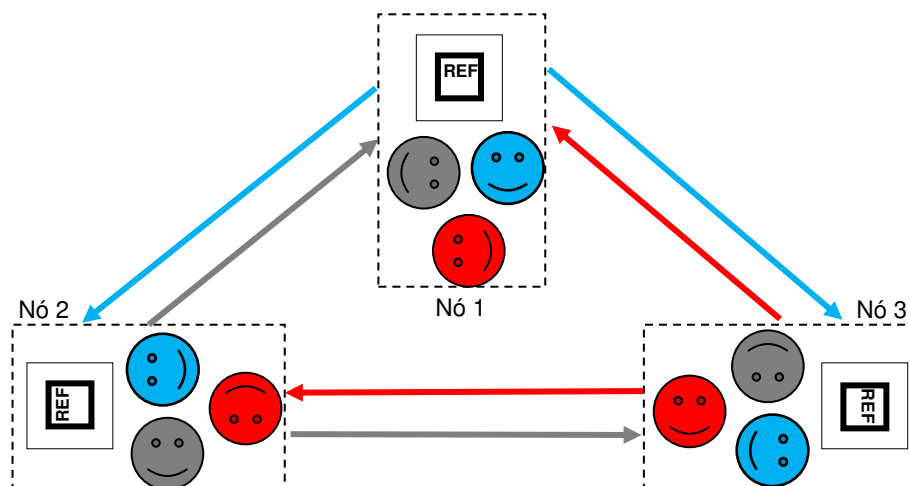


FIGURA 6.12. ESQUEMATIZAÇÃO DO USO DA REFERÊNCIA REMOTA PARA COLABORAÇÃO SÍNCRONA.

O modelo de comunicação em rede, utilizado no SACRA, consiste no modelo peer-to-peer, enviando e recebendo mensagens de todos os nós cadastrados no arquivo `r_hosts`. O arquivo `r_hosts` consiste num arquivo texto, contendo os endereços IPs dos hosts e as portas requisitadas pelo sistema. A Figura 6.13 demonstra um exemplo do arquivo `r_hosts`.

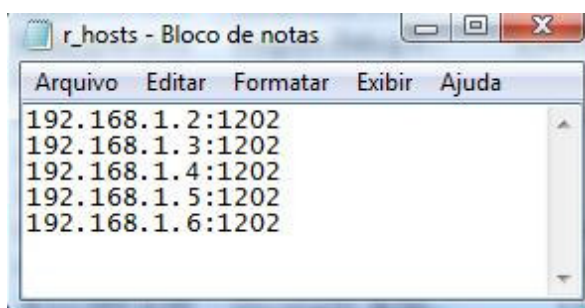


FIGURA 6.13. EXEMPLO DO ARQUIVO DE HOSTS DO SACRA.

A referência remota realiza o salvamento em arquivo do trabalho elaborado. Assim, o trabalho colaborativo, referente a essa referência, também pode ser o assíncrono e assíncrono distribuído, como na referência local.

A diferença entre as referências remota e local está no fato da referência remota promover o envio de mensagens aos usuários remotos, através da execução de ações sobre os pontos associados a esse marcador.

As mensagens enviadas pelo SACRA são os comandos a serem executados e os seus parâmetros. A Tabela 6.1 demonstra o nome dos comandos a suas operações e as ações que o dispararam.

TABELA 6.2. POSSÍVEIS COMANDOS ENVIADOS AOS USUÁRIOS REMOTOS

Comandos	Operações	Executor
add_object	Adiciona um objeto	Cadastramento de pontos e Copy
remove_object	Remove um objeto	Eraser
update_object	Atualiza a orientação e a posição do objeto	Transport
lock_object	Bloqueio ou desbloqueia o objeto	Lock
remove_path	Remove o rastro	Tecla 'c'

O marcador de referência possibilita que o usuários enviem trabalhos já elaborados nesse marcador, através de uma funcionalidade oferecida pelo SACRA, denominada difusão. A difusão é executada ao pressionar a tecla 'd' do teclado.

Outra funcionalidade oferecida pela referência remota consiste no envio do rastro da trajetória do marcador Path em relação à referência. O rastro consiste numa seqüência de objetos virtuais que possibilitam oferecer uma indicação visual de um caminho aos usuários remotos. A Figura 6.14 demonstra o exemplo da utilização do rastro representado pelas esferas em azul. Uma esfera vermelha percorre a trajetória, a fim de indicar a sua direção aos usuários.

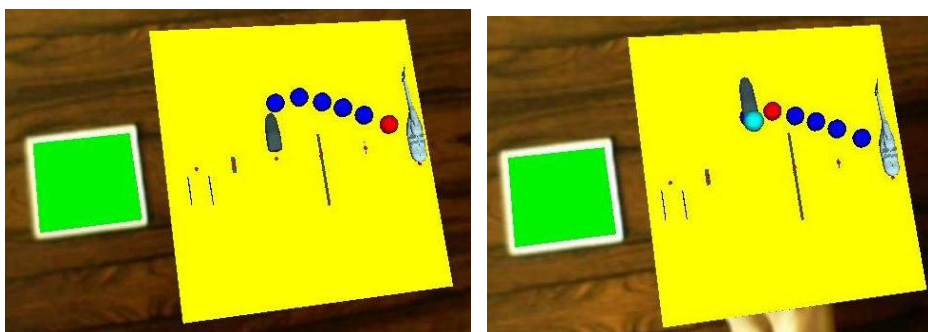


FIGURA 6.14. USO DO RASTRO PARA INDICAR A TRAJETÓRIA.

A referência remota implementa as técnicas do trabalho colaborativo: cooperação, coordenação e a comunicação. A cooperação é possível pela área comum aos usuários remotos oferecida pela referência remota. A coordenação das ações dos usuários sobre os objetos virtuais é executada através do uso do marcador Lock, que bloqueia e desbloqueia a atuação dos usuários remotos sobre determinados objetos. A comunicação, nesse caso, pode ser realizada por programas externos de chats e videoconferências, além do uso rastro para indicar a trajetória que os usuários remotos devam realizar.

6.5 TESTE DE FUNCIONALIDADE.

Os diversos recursos de interação, oferecidos pelo SACRA, permitem estender o seu uso a diversas aplicações. A seguir, será demonstrada uma aplicação do SACRA para a montagem de ambientes virtuais, tanto de maneira local, quanto remotamente.

6.5.1 TRABALHO LOCAL

Para demonstrar o trabalho no ambiente local, o SACRA foi configurado com vários objetos virtuais para a montagem de um cômodo de uma casa. No caso, o objetivo estipulado foi montar uma sala de televisão.

A referência local foi utilizada para agrupar vários objetos relacionados à mobília de uma casa. Os objetos virtuais foram associados a pontos cadastrados referentes à referência local. Dessa maneira, através dos marcadores de ações, o usuário realizava a interação com os objetos virtuais. A Figura 6.15 demonstra o usuário ativando a visualização dos objetos virtuais com o marcador de inspeção. Para visualizar os objetos, era necessário aproximar a esfera do marcador de inspeção da esfera que representava o ponto.

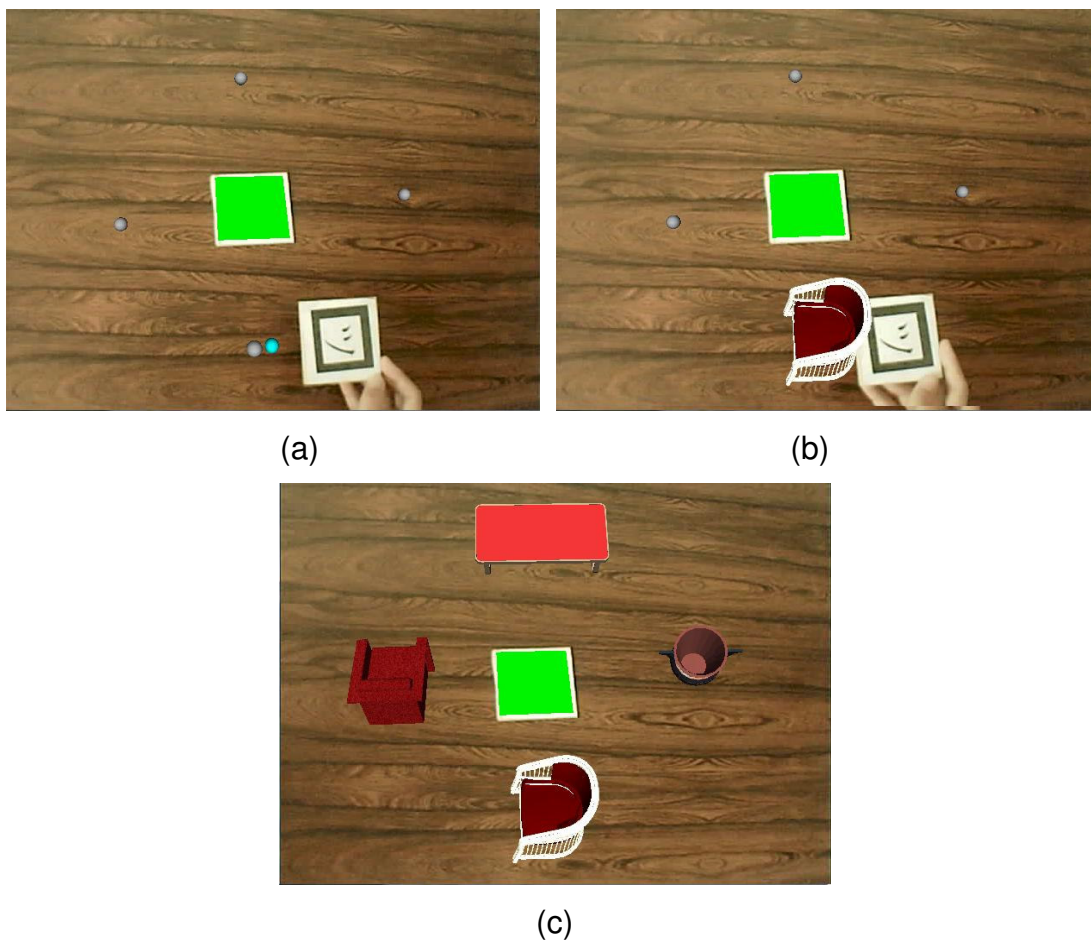


FIGURA 6.15. USO DO MARCADOR DE INSPEÇÃO: A) APROXIMAÇÃO DA ESFERA DO MAÇADOR DE INSPEÇÃO DA ESFERA REPRESENTANTE DO PONTO, B) ATIVAÇÃO DA VISUALIZAÇÃO DO OBJETO E C) VISUALIZAÇÃO DOS OBJETOS VIRTUAIS REPRESENTADOS PELOS PONTOS.

Realizada a ativação da visualização dos objetos virtuais representados pelos pontos, o usuário tinha a possibilidade de alterar os objetos virtuais utilizando o marcador de controle. A Figura 6.16 demonstra a alteração do objeto virtual, utilizando o marcador de controle.



FIGURA 6.16. USO DO MARCADOR DE CONTROLE PARA A TROCA DE OBJETOS VIRTUAIS: A) TELEVISÃO E B) TAPETE

O próximo passo, após o usuário encontrar o objeto desejado, consiste em selecionar o marcador de referência que será o local de destino do objeto a ser copiado. A seleção da referência de destino é realizada com a aproximação do marcador de cópia do marcador de referência, como mostra a Figura 6.17.

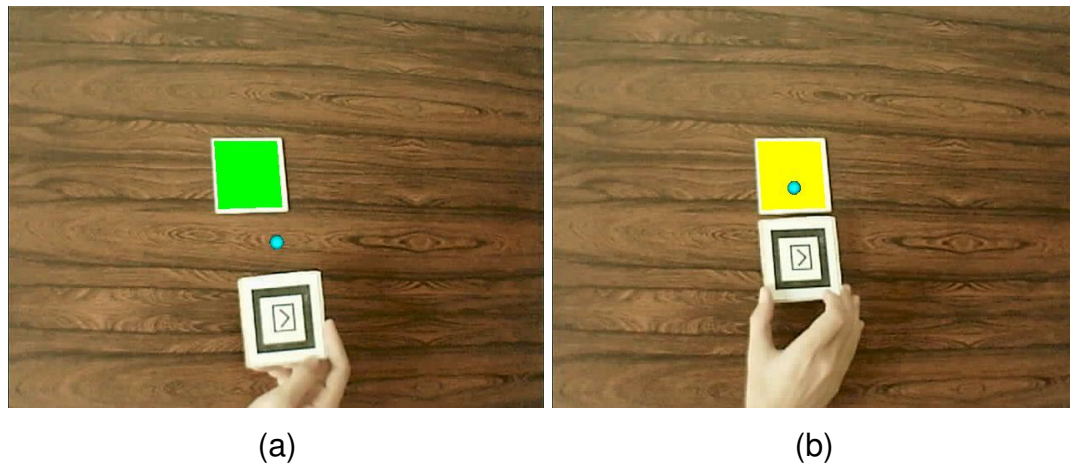


FIGURA 6.17. SELEÇÃO DA REFERÊNCIA DE DESTINO: A) APROXIMAÇÃO DA ESFERA DO MARCADOR DE CÓPIA DA REFERÊNCIA E B) REFERÊNCIA SELECIONADA.

Selecionada a referência de destino, o usuário deve copiar o objeto virtual com o próprio marcador de cópia e fixar o objeto copiado na referência de destino, através da sua oclusão. A Figura 6.18 mostra os passos para realizar a cópia do objeto virtual.

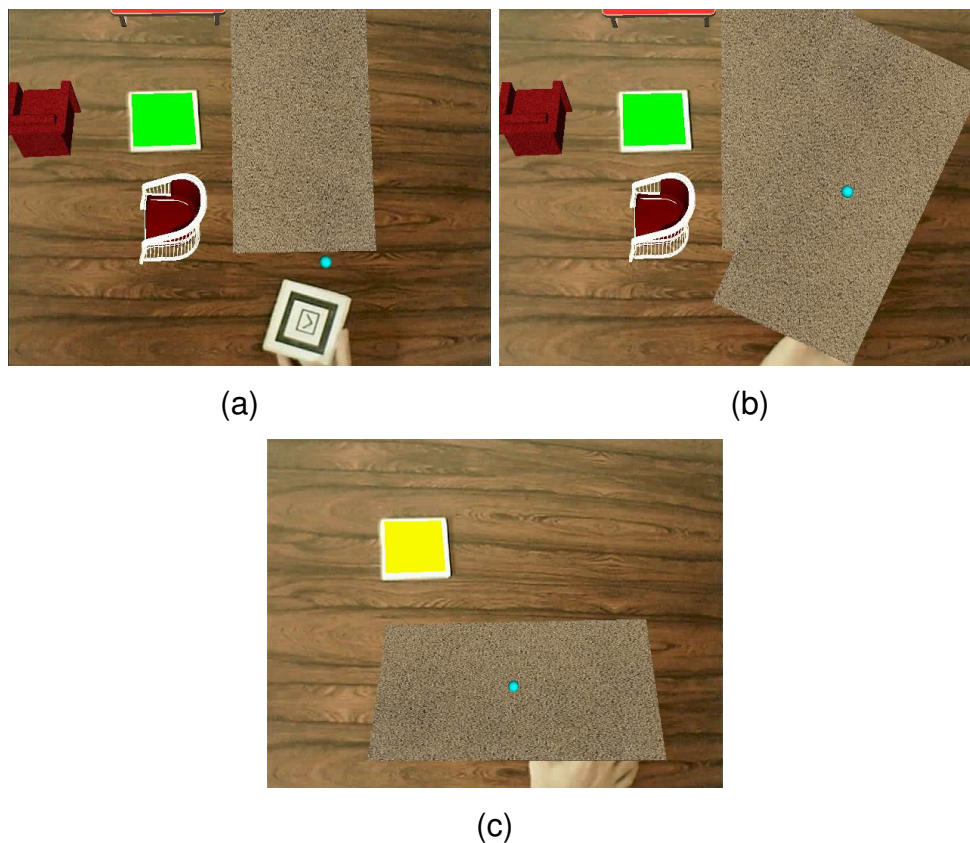


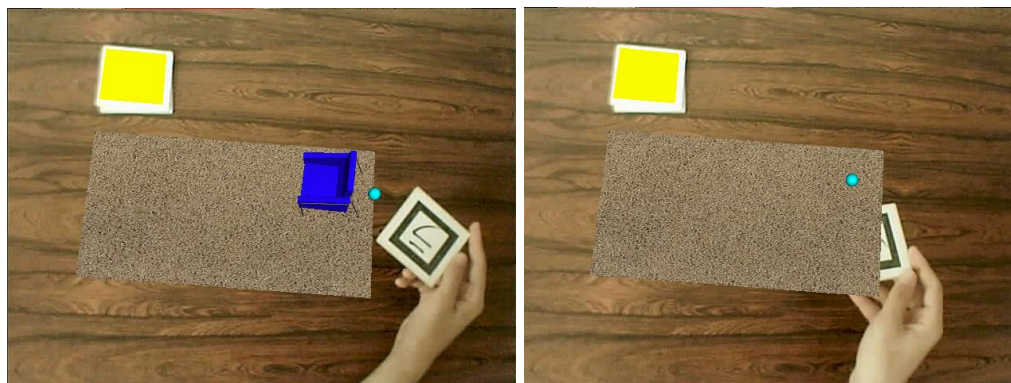
FIGURA 6.18. CÓPIA DE OBJETO VIRTUAL: A) APROXIMAÇÃO DA ESFERA DO MARCADOR DE CÓPIA DO OBJETO, B) OBJETO COPIADO E C) FIXAÇÃO DO OBJETO COPIADO NO MARCADOR DE DESTINO.

Após a fixação, o usuário pode modificar a posição e orientação do objeto virtual fixado, através do uso do marcador de transporte. A Figura 6.19 mostra o reposicionamento de uma cadeira.



FIGURA 6.19. TRANSPORTE DO OBJETO VIRTUAL: A) APROXIMAÇÃO DA ESFERA DO MARCADOR DE TRANSPORTE DO OBJETO A SER COPIADO E B) OBJETO REPOSICIONADO E REORIENTADO ATRAVÉS DO MARCADOR DE TRANSPORTE.

A possibilidade de remoção do objeto fixado é oferecida pelo marcador de remoção. A figura 6.20 demonstra o usuário removendo um objeto virtual.



(a)

(b)

FIGURA 6.20. REMOÇÃO DO OBJETO VIRTUAL: A) APROXIMAÇÃO DO MARCADOR DE REMOÇÃO DO OBJETO VIRTUAL A SER REMOVIDO E B) OBJETO VIRTUAL REMOVIDO.

O resultado final da montagem do ambiente virtual solicitado é demonstrado na Figura 6.21.

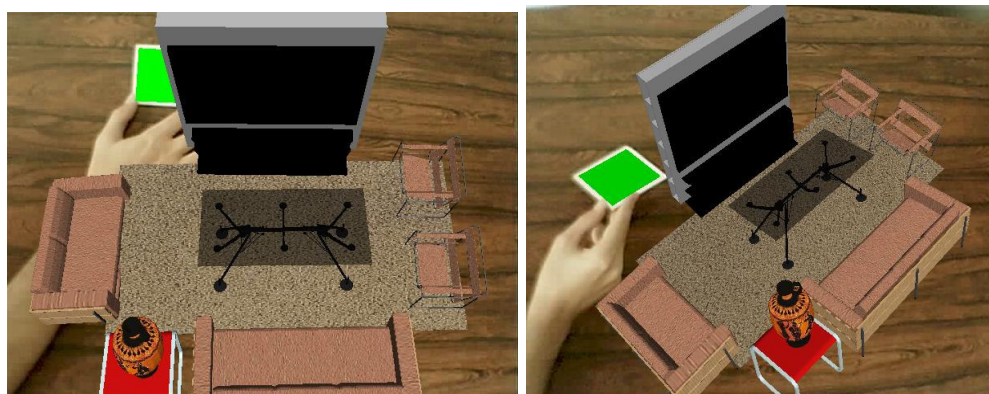


FIGURA 6.21. MONTAGEM DE UMA SALA VIRTUAL.

6.5.2 TRABALHO REMOTO

O exemplo do trabalho remoto foi elaborado, utilizando dois ambientes computacionais diferentes. O objetivo estabelecido para o trabalho de autoria remoto era montar um helicóptero. Nesse caso, o usuário do ambiente 1, que será denominado de usuário1, insere os objetos do helicóptero na referência remota e os envia por difusão para o usuário do ambiente 2,

denominado usuário2. Realizada a difusão, ambos os usuários possuem os mesmos objetos virtuais. A Figura 6.22 mostra a visualização das peças do helicóptero pelos usuários.

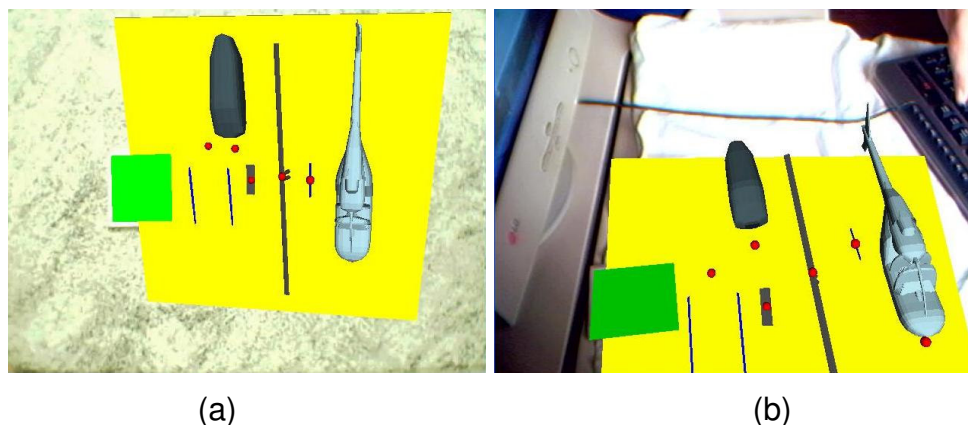


FIGURA 6.22. VISUALIZAÇÃO DA REFERÊNCIA REMOTA COM AS PEÇAS DO HELICÓPTERO: A) USUÁRIO1 E B) USUÁRIO2.

A operação de difusão realizada pelo usuário1, nesse caso, foi efetuada sem bloqueio no envio, permitindo que o usuário2 interagisse com os objetos virtuais, através dos marcadores de ações.

A indicação da ação que será realizada é elaborada com o marcador de rastro. Dessa maneira, o usuário1 indica o caminho que o usuário2 deve realizar com a peça do helicóptero. A Figura 6.23 mostra a trajetória indicando o caminho a ser seguido.

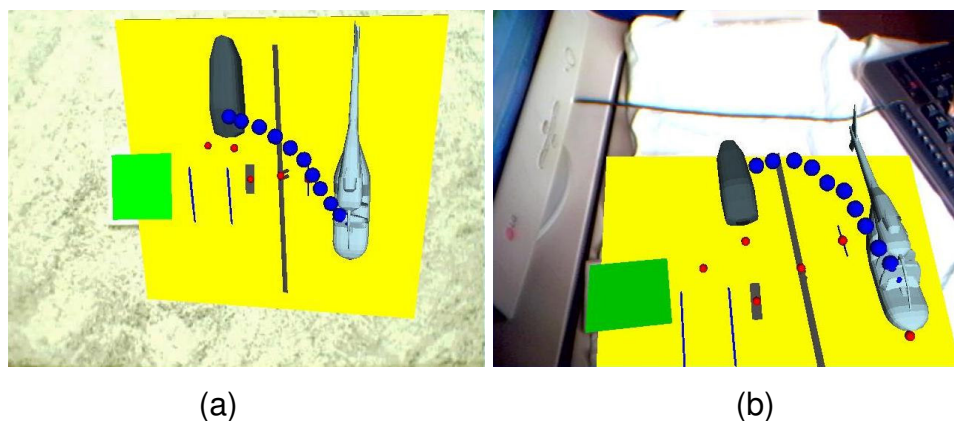
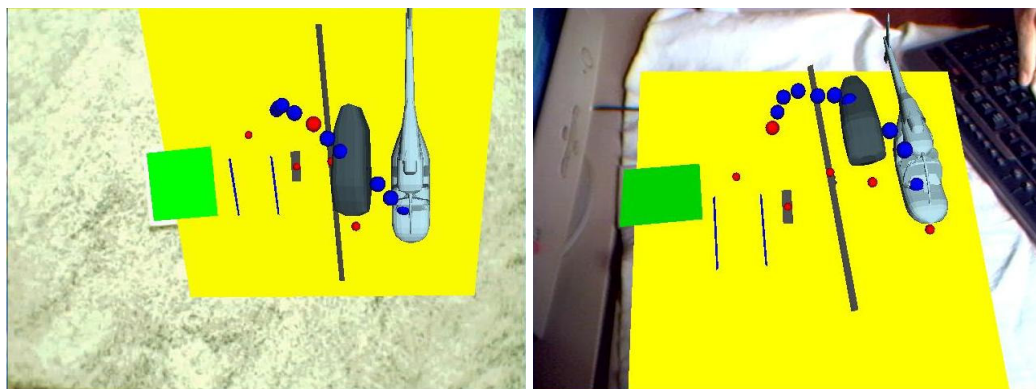


FIGURA 6.23. TRAJETÓRIA INDICADA PELA SEQÜÊNCIA DE ESFERAS: A) TRAJETÓRIA VISUALIZADA PELO USUÁRIO1 E B) TRAJETÓRIA VISUALIZADA PELO USUÁRIO2.

Indicada a rota a ser seguida, o próximo passo consiste em transportar a peça do helicóptero, seguindo o rastro visualizado através do uso do marcador de transporte. A Figura 6.24 exibe a peça do helicóptero movimentada pelo usuário2.



(a)

(b)

FIGURA 6.24. MOVIMENTAÇÃO DA PEÇA DO HELICÓPTERO PELO USUÁRIO2: A) VISUALIZAÇÃO DO MOVIMENTO PELO USUÁRIO1 E B) PONTO DE VISTA DO USUÁRIO2.

O fim do trabalho remoto ocorre com a completa montagem do helicóptero. A Figura 6.25 mostra o helicóptero montado.

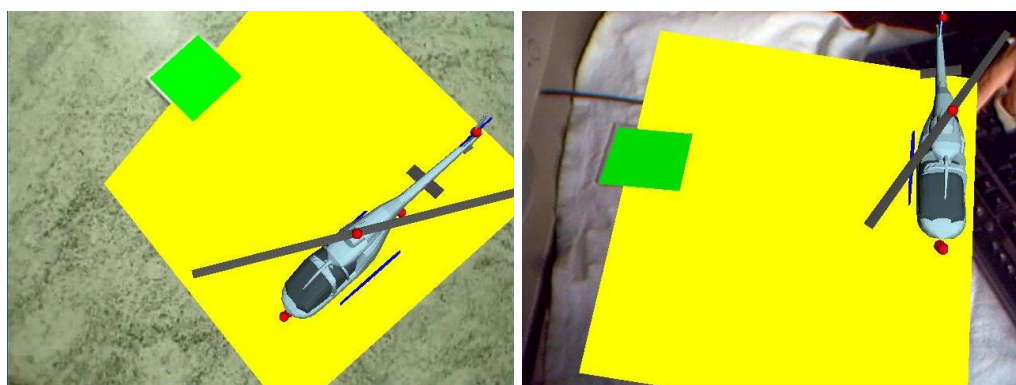


FIGURA 6.25. MONTAGEM COMPLETA DO HELICÓPTERO.

7 CONCLUSÃO

A realidade aumentada é considerada uma interface avançada computacional que realiza, em tempo real, a sobreposição de elementos sintetizados por computador na imagem do mundo real. A potencialidade da visualização dos elementos computacionais no ambiente real, juntamente com as técnicas de interação tangíveis, permitem aos usuários manipularem intuitivamente objetos virtuais, como se fossem reais.

O ARToolKit consiste em uma biblioteca de programação que viabiliza o desenvolvimento de aplicações de RA tangíveis, possibilitando realizar a interação dos usuários com dados computacionais complexos, através da manipulação de simples e módicos dispositivos, os marcadores.

Existem inúmeros sistemas de RA que utilizaram o ARToolKit, para os seus desenvolvimentos. Alguns desses sistemas são ferramentas de autoria, que buscam utilizar os benefícios da interação atrelada à visualização para permitir que usuários não programadores possam criar as suas próprias aplicações. Exemplos de alguns dessas ferramentas de autoria que utilizam recursos do ARToolKit são o TILES, iaTAR e AMIRE. Embora essas ferramentas permitam facilitar e agilizar o trabalho de seus usuários, nenhuma explora os benefícios do trabalho colaborativo.

Nesse sentido, o presente trabalho abordou o desenvolvimento de um sistema de autoria colaborativo de realidade aumentada, denominado SACRA. Esse sistema é caracterizado por oferecer técnicas de interação avançadas para a autoria colaborativa de ambientes virtuais.

A interação no SACRA é realizada por meio dos marcadores de ações, que implementam funcionalidades especializadas para a manipulação de objetos virtuais. Os marcadores de ações possibilitam que os usuários controlem a presença e alterem algumas características desses objetos virtuais.

Um aspecto inovador implementado no sistema consiste na utilização de pontos, no ambiente real do usuário, para a representação dos objetos virtuais. Dessa maneira, o SACRA oferece a possibilidade da visualização de objetos virtuais atrelados não só aos marcadores, mas também, aos pontos que são orientados no ambiente pelos marcadores de referência.

Assim, o uso dos pontos expandiu a abrangência do sistema desenvolvido a diversas aplicações, visto a possibilidade de associar objetos virtuais a objetos reais, somente com a fixação de um marcador de referência no ambiente do usuário. Além disso, foi possível reduzir o número de marcadores utilizados no sistema para a visualização de objetos virtuais.

A colaboração no SACRA pode ocorrer: face a face, através do uso de dispositivos de visão compartilhada como capacetes, projeções ou monitores, possibilitando mais de um usuário envolvido na elaboração de um ambiente; assíncrona, visto que o sistema salva o trabalho e este pode ser retomado posteriormente; assíncrona distribuída, pois o trabalho salvo pode ser enviado por email para a colaboração de outros usuários; síncrona distribuída, devido a troca de mensagens entre os usuários remotos, que possibilita a construção cooperativa e coordenada dos ambientes virtuais.

Enfim, o trabalho proposto realizou estudos sobre os assuntos relacionados com realidade aumentada, realidade virtual, ambientes de autoria e ambientes colaborativos, visando desenvolver um sistema capaz de aliar os benefícios dos ambientes colaborativos e de autoria com o ambiente de realidade aumentada altamente interativo. Como resultado final obteve-se o sistema de autoria colaborativa com realidade aumentada, denominado SACRA, que implementa recursos inovadores de interação com objetos virtuais, servindo de base para inúmeras aplicações de realidade aumentada.

Para os desdobramentos do trabalho, o SACRA pode incorporar técnicas de inteligência artificial aos objetos virtuais, permitindo aos usuários influenciarem nos seus comportamentos. Outro trabalho, já em andamento, consiste em desenvolver uma versão do SACRA em Python e

utilizá-la com o software Vizard, permitindo explorar os recursos avançados oferecidos por essa ferramenta em ambientes de realidade aumentada.

Referências Bibliográficas

- ABDULLAH, J.; MARTINEZ, K. (2002) **Camera Self-Calibration for the ARToolkit**. In Proceedings of First International Augmented Reality Toolkit Workshop, pp. 84-88, Darmstadt, Germany
- ABSOLUT TECHNOLOGIES. **Visualizações Inteligentes em Realidade Virtual**. <www.abs-tech.com > acesso em mar. 2007.
- AGRAWALA, M. et al. **The two-user responsive workbench: Support for collaboration through individual views of a shared space**. In: Computer Graphics Proceedings. Los Angeles, 1997, Annual Conference Series, ACM SIGGRAPH, P. 327-332
- ARANHA, M. L. A. ; MARTINS, M. H. P. **Filosofando: Introdução à filosofia**. São Paulo: Moderna, 1993.
- ARToolKit. Human Interface Technology Laboratory <<http://www.hitl.washington.edu/artoolkit/>> acesso em fev, 2007.
- ARToolworks, Inc.< <http://www.artoolworks.com/>>acesso em fev, 2007.
- AZUMA, R. T. (1997). **A Survey of Augmented Reality**. Teleoperators and Virtual Environments, 355-385, 1997.
- BAUER, M. **Design of a Component-Based Augmented Reality Framework**. In: Proceedings of the 2nd IEEE and ACM International Symposium on Augmented Reality, ISAR 2001, New York, NY, Outubro, 2001, IEEE Computer Society.
- BILLINGHURST, M.; KATO, H., (1999) **Real World Teleconferencing**. In: Proceedings of the conference on Human Factors in Computing Systems .CHI 99. Pittsburgh, USA, Maio, 1999.
- BILLINGHURST, M.; KATO, H. 2002. **Collaborative augmented reality**. Commun. ACM 45, 7 , jul. 2002, P. 64-70
- BILLINGHURST, M. **No More WIMPs: Designing Interfaces for the Real World**. Keynote Address, Symposium on Computer Human Interaction (SigCHI NZ), July 3-4, 2003, University of Otago, Dunedin, New Zealand.

- BILLINGHURST, M.; GRASSET, R.; LOOSER, J. (2005). **Designing augmented reality interfaces**. SIGGRAPH Comput. Graph. 39, 1. Feb. 2005, p17-22.
- BOWMAN, D. et al. **An Introduction to 3-D User Interface Design**. PRESENCE: Teleoperators and Virtual Environments, 10(1):96-108, February 2001
- BRUEGGE, B.; KLINKER, G.(2003) **DWARF Distributed Wearable Augmented Reality Framework**. Disponível em: < <http://campar.in.tum.de/twiki/pub/Chair/DwarfWhitePaper/dwarf.pdf> > acesso em : mar, 2007
- BÜSHER, M. et al.(1999). **Supporting cooperation across shared virtual environments**. In: Proceedings of the international ACM SIGGROUP Conference on Supporting Group Work .Phoenix, Arizona, United States, Novembro, 1999. GROUP '99. ACM Press, New York, NY. P. 61-70.
- CLAUS, D.; FITZGIBBON, A. W. **Reliable Automatic Calibration of a Marker-Based Position Tracking System**. *wacv-motion*, pp. 300-305, Seventh IEEE Workshops on Application of Computer Vision (WACV/MOTION'05) - Volume 1, 2005.
- CLIBURN, D.C. Virtual Reality for Small Colleges. **Journal of Computing Sciences in Colleges**, v.19 n.4, p.28-38, abr. 2004
- COMMER, D. E. **Interligação em rede com TCP/IP**. Rio de Janeiro: Elsevier, 1998
- CONSULARO, L.A.; CALONEGO JR, N.; DAINESE, C.A.; GARBIN, T. R.; KIRNER, C.;TRINDADE, J.; FIOLHAIS, C.(2004) **ARToolKit: Aspectos Técnicos e Aplicações Educacionais**. In: CARDOSO, A.; LAMOUNIER JR, E. editores. Realidade Virtual: Uma Abordagem Prática. Livro dos Minicursos do SVR2004, SBC, São Paulo, 2004, p. 141-183.
- CONWAY, M. et al. **Alice: Lessons Learned from Building a 3D System for Novices**. in : Proceedings CHI 2000, p. 486-493.

- COULOURIS, G.; DOLLIMORE, J.; KINDBERG, T. **Distributed Systems: Concepts and Design**. 3 ed. Pearson Education. 2001. 772 pp. ISBN: 0201-61918-0
- ELLIS, C. A., GIBBS, S. J., AND REIN, G. (1991). **Groupware: some issues and experiences**. Commun. ACM 34, 1, jan. 1991. P. 39-58
- ENGDAHL, T. **3D glasses and other 3D display devices**. <<http://www.epanorama.net/documents/pc/3dglass.html>> acesso em mar. 2007.
- FOUSS, J. D.; CHANG, K. H. (2000). **Classifying groupware**. In: Proceedings of the 38th Annual on Southeast Regional Conference, Clemson, South Carolina, Abril, 2000. ACM-SE 38. ACM Press, New York, NY. P.117-124.
- FUKS, H., RAPOSO, A.B. & GEROSA, M.A. (2003) **Do Modelo de Colaboração 3C à Engenharia de Groupware**. Simpósio Brasileiro de Sistemas Multimídia e Web – Webmidia 2003, Trilha especial de Trabalho Cooperativo Assistido por Computador, 03 a 06 de Novembro de 2003, Salvador-BA
- GEIGER, C.; REIMANN, C.; STICKLEIN, J.; PAELKE, V. (2002) **JARToolKit- A Java binding for ARToolKit**. In: The First IEEE Augmented Reality Toolkit International Workshop, p124, 2002.
- GNU General Public License< <http://www.gnu.org/licenses/gpl.html> #TOC1 >acesso em fev, 2007
- GOEBBELS, G.; LALIOTI, V.; GÖBEL, M. (2003). **Design and evaluation of team work in distributed collaborative virtual environments**. In: Proceedings of the ACM Symposium on Virtual Reality Software and Technology. Osaka, Japan, Outubro, 2003. VRST '03. ACM Press, New York, NY, P. 231-238
- GOMES, C. (2002) **ECONOMIA DO SISTEMA COMUNITÁRIO - Enquanto a mercadoria e a moeda não existem**. Disponível em <<http://www.eumed.net/libros/2005/cg/index.htm>> acesso em 10 mar, 2007

- GOSSWEILER, R. et al. **An Introductory Tutorial for Developing Multi-User Virtual Environments**, Computer Science Department, University of Virginia, Charlottesville. Presence, 3(4):255–264, 1994.
- GRIMM, P. et al. **AMIRE—Authoring Mixed Reality**. First IEEE International Augmented Reality Toolkit Workshop, Darmstadt, Germany, 2002
- GRUDIN, J. (1994). **Computer-Suported Cooperative Work: History and Focus**, IEEE Computer, 27(5), p.19-26, 1994
- GRUDIN, J.; POLTROCK, S. (1996). **CSCW, groupware and workflow: experiences, state of art, and future trends**. In: Conference Companion on Human Factors in Computing Systems: Common Ground. Vancouver, British Columbia, Canada, Abril, 1996. M. J. Tauber, Ed. CHI '96. ACM Press, New York, NY. P. 338-339.
- HAWKING, S. **O Universo numa casca de noz**. São Paulo : Arx, 2002.
- HENDRICKS, Z.; MARSDEN, G.; BLAKE, E.(2003). **A meta-authoring tool for specifying interactions in virtual reality environments**. In: Proceedings of the 2nd international Conference on Computer Graphics, Virtual Reality, Visualisation and interaction in Africa. Cape Town, South Africa, February 03 - 05, 2003. AFRIGRAPH '03. ACM Press, New York, NY, p .171-180.
- HORNECKER, E.; PSIK, T. **Using ARToolKit Markers to Build Tangible Prototypes and Simulate Other Technologies**. Proc. of INTERACT 2005 Rome. 2005.
- INSLEY, S. **Obstacles to general purpose augmented reality**. Information Security e Criptography, Oregon, EUA, 2003.
- KATO, H.; BILLINGHURST, M.(1999). **Marker Tracking and HMD Calibration for a Videobased Augmented Reality Conferencing System**. In: Proceedings of the 2nd IEEE and ACM Internationall Workshop on Augmented Reality, San Francisco, CA,USA, p85-94, 1999.
- KATO, H.; BILLINGHURST, M.; POUPYREV, I. (2000) “ARToolKit Version 2.33”, Human Interface Lab, Universidade de Washington.

- KATO, H. Inside ARToolKit.(2002).< <http://www.hitl.washington.edu/artoolkit/Papers/ART02-Tutorial.pdf>> acesso em fev, 2007.
- KIRNER, C. et al. (2004a). **Uso de Realidade Aumentada em Ambientes Virtuais de Visualização de Dados** Proc. of VII Symposium on Virtual Reality, SP, outubro de 2004.
- KIRNER, C.; TORI, R. (2004b) **Introdução à Realidade Virtual, Realidade Misturada e Hiper-realidade**. In: Claudio Kirner; Romero Tori. (Ed.). Realidade Virtual: Conceitos, Tecnologia e Tendências. 1ed. São Paulo, 2004, v. 1, p. 3-20.
- KIRNER, C.; TORI, R. **Fundamentos de Realidade Aumentada**. In: Claudio Kirner; Romero Tori; Robson Siscoutto. (Ed.). Fundamentos e Tecnologia de Realidade Virtual e Aumentada. Pré Simpósio SVR 2006, SBC, Belém, 2006, pp. 22-38.
- KIRNER, C.; KIRNER, T.G. (2007) **Virtual Reality and Augmented Reality Applied to Simulation Visualization**. In: El Sheikh, A.A.R.; Al Ajeeli, A.; Abu-Taieh, E.M.O.. (Org.). Simulation and Modeling: Current Technologies and Applications. 1 ed. Hershey-NY: IGI Publishing, 2007, v. 1, p. 391-419.
- KIRNER, C. (2007) pyARToolKit www.ckirner.com
- LEDERMANN, F. **An Authoring Framework for Augmented Reality Presentations**. Tese (Mestrado), Universidade de Tecnologia de Viena , Maio 2004.
- LEE, G. A.; KIM, G. J.; BILLINGHURST, M. (2005). **Immersive authoring: What You eXperience Is What You Get (WYXIWYG)**. Commun. ACM 48, 7 .Jul. 2005. P. 76-81.
- LEPETIT, V.; FUA, P.(2005) **Monocular Model-Based 3D Tracking of Rigid Objects: A Survey**. Foundations and Trends in Computer Graphics and Vision, Vol.1, N.1, pp. 1-89, Out 2005.
- LOOSER, J.; GRASSET, R.; SEICHTER, H.; BILLINGHURST, M.(2006) **OSGART - A pragmatic approach to MR**. Proceedings of the Fifth IEEE

and ACM International Symposium on Mixed and Augmented Reality (ISMAR06).

MACINTYRE, B.; GANDY, M. **Prototyping applications with DART, the designer's augmented reality toolkit**, in: Proceedings of International Workshop on Software Technology for Augmented Reality Systems '03, Tokyo, Japan, 7 Outubro, 2003, pp. 19-22.

MACREDIE, R. *et al.* **Virtual reality and simulation: an overview**. In Proceedings of the 28th Conference on Winter Simulation Coronado, California, Estados Unidos, dez. 1996.

MATTHIESEN, R. C. **Virtual World Builder :Agentes Móveis com Realidade Aumentada**. Piracicaba: Unimep, 2006. 93 p. Dissertação (Mestrado). Programa de Mestrado em Ciência da Computação da Faculdade de Ciências Matemáticas e da Natureza da Universidade Metodista de Piracicaba – UNIMEP, 2006.

Media Machine. <<http://william.mediamachines.com/press/pressrelease-02272007.php>> acesso em: abr, 2007.

Moodle. **Modular Object Oriented Dynamic Learning Enviroment** Diponível em : <<http://moodle.org/>> acesso em : abr, 2007.

NAJORK, M. **Programming in Three Dimensions**. Dissertação (Ph.D.), Universidade de Illinois em Urbana-Champaign, 1994.

NETTO, V. et al. **Realidade virtual - definições, dispositivos e aplicações**. 2002. p. 1-29. In: REIC - Revista Eletrônica de Iniciação Científica v. 2, n. 1,p. 1-29, 2002.

OLIVEIRA,L.A.; CALONEGO, N. J. **Uma Aplicação Cliente-Servidor Usando ARToolKit** . In: Workshop de Realidade Aumentada, 2006,Rio de Janeiro. Anais do 3º Workshop de Realidade Aumentada. Rio de Janeiro: UERJ, 2006. v. 3. p. 31-34.

OTTO, O.; ROBERTS, D.; WOLFF, R. (2006). **A review on effective closely-coupled collaboration using immersive CVE's**. In: Proceedings of the 2006 ACM international Conference on Virtual Reality Continuum and Its

Applications .Hong Kong, China. VRCIA '06. ACM Press, New York, NY, P. 145-154

PIEKARSKI, W.; THOMAS, B. H.(2002) **Using ARToolKit for 3D Hand Position Tracking in Mobile Outdoor Environments**. In 2nd International Augmented Reality Toolkit Workshop, Darmstadt, Set, 2002.

PIERCE, J. S. et al. (1997) **Alice: Easy to Use Interactive 3D Graphics**. UIST 97 Conference Proceedings, p 77-78.

PINHO, M. S.; KIRNER, C. **Uma introdução à Realidade Virtual**. Disponível em <<http://grv.inf.pucrs.br/Pagina/TutRV/tutrv.htm>.> Acesso em 4 fev. 2007.

PINKER, S. **Como a Mente Funciona**. São Paulo: Companhia das Letras,1998.

PIRGHEIM, C. **Visual Programming of User Interfaces for Distributed Graphics Applications**. Tese (Mestrado), Institut Für Maschinelles Sehen und Darstellen. Universidade Tecnológica de Graz, 2006.

Postech < <http://home.postech.ac.kr/~endovert/iatar/> > acesso em abr, 2007.

POUPYREV, I. et al. **TILES: A Mixed Reality Authoring Interface**. Proceedings of Interact 2001. 2001: p. 334-341.

POUPYREV, I. et al. (2002). **Developing a Generic Augmented-Reality Interface**. Computer 35, 3. mar. 2002, p 44-50.

PSIK, T. **Designing multimodal interaction for configurable distributed systems**. Tese (Phd), Universidade de Tecnologia de Viena, 2004.

RAMA, J.; BISHOP, J.(2006). **A survey and comparison of CSCW groupware applications**. In : Proceedings of the 2006 Annual Research Conference of the South African institute of Computer Scientists and information Technologists on IT Research in Developing Countries, Somerset West, South Africa, Outubro, 2006. J. Bishop and D. Kourie, Eds. SAICSIT '06, vol. 204. ACM Press, New York, NY. P. 198-205

REGENBRECHT, H. T.; WAGNER, M. T. (2002). **Interaction in a collaborative augmented reality environment**. In: CHI '02 Extended Abstracts on Human Factors in Computing Systems, Minneapolis,

Minnesota, USA, Abril, 2002. CHI '02. ACM Press, New York, NY, P. 504-505.

SANTIN, R.; KIRNER, C. **Alteração dinâmica de objetos virtuais em Realidade Aumentada**. In: Workshop de Realidade Aumentada, 2005, Piracicaba. Anais do 2º Workshop de Realidade Aumentada. Piracicaba : Unimep, 2005. v. 2. p. 37-40

SECOND LIFE, < <http://secondlife.com/>> Acesso em fev, 2007.

SCHMALSTIEG, D.; FUHRMANN, A.; HESINA, G. **Bridging multiple user interface dimensions with augmented reality**. In: Proceedings of the 3rd International Symposium on Augmented Reality, ISAR 2000, Munich, Germany. IEEE, October 2000.

SCHMALSTIEG, D. et al. **The Studierstube Augmented Reality Project** PRESENCE - Teleoperators and Virtual Environments, Vol. 11, No. 1, pp. 32-54, MIT Press, February 2002

SCHWALD, B.(2003). **An Augmented Reality System for Training and Assistance to Maintenance in the Industrial Context**. <http://wscg.zcu.cz/wscg2003/Papers_2003/I23.pdf> acesso em mar.2007

SHERMAN, R. William; Craig, B. Alan. **Understanding Virtual Reality; Interface, Application and Design**. San Francisco: Morgan Kaufmann,2003.

SILBERSCHATZ, A.; GALVIN, P.; GREG, G. **Sistemas Operacionais Conceitos e Aplicações**. Rio de Janeiro: Editora Campus, 2000.

Stereo Vision. <<http://www.vision3d.com/stereo.html>> acesso em 02 mar.2007.

SUTCLIFFE, A. **Multimedia and Virtual Reality: designing multisensory user interface**. Nova Jersey: Lawrence Erlbaum, 2002.

TANENBAUM, A.S. Distributed Operating Systems. 1. ed. New Jersey: Prentice-Hall. 1995. 614 pp. ISBN: 0-13-219908-4.

TILLEY,T.; POLLITT, S.; EKLUND, P. **Evaluating Distributed Graphical User Interface Communication**. In Fifth Australian World Wide Web Conference (AusWeb99), 1999.

- TORI, R.; KIRNER, C. **Fundamentos de Realidade Virtual**. In: Claudio Kirner; Romero Tori; Robson Siscoutto. (Ed.). Fundamentos e Tecnologia de Realidade Virtual e Aumentada. Pré Simpósio SVR 2006, SBC, Belém, 2006, pp. 2-21.
- ULBRICHT, C.; SCHMALSTIEG, D. **Tangible augmented reality for computer games**. Technical report, The 3rd IASTED International Conference on Visualization, Imaging, and Image Processing (VIIP 2003), September 2003
- VINCE, J. **Introduction to Virtual Reality**. Londres :Springer-Verlag, 2004.
- WAGNER, D.; SCHMALSTIEG, D.(2003) **ARToolKit on the PocketPC Platform**. In:Proceedings of the 2nd Augmented Reality Toolkit Workshop, p14-15, 2003.
- Web3d <<http://www.web3d.org/x3d/content/README.X3D-Edit.html>> acesso em abr, 2007.
- WITMER, B. G.; SINGER, M. J. (1998). **Measuring presence in virtual environments: A presence questionnaire**. Presence: Teleoperators and Virtual Envrionments, 7(3), 225 - 240.
- WorldViz < <http://www.worldviz.com/products/vizard/index.html>> acesso em abr, 2007.
- XAVIER, A. J. et al. **Desafios de interação e acessibilidade para o usuário Idoso**. In: Congresso Brasileiro de Informática em Saúde, 9., 2004, Ribeirão Preto. Anais.
- YANG,Z.; DUDDY, K. **CORBA: A Platform for Distributed Object Computing (A State-of-the-Art Report on OMG/CORBA)**. ACM Operating Systems Review, Volume 30, Number 2, Abril,1996, pp 4-31

ANEXOS

ANEXO 1- COMPILAÇÃO DO ARTOOLKIT NA PLATAFORMA WINDOWS

Para compilar as bibliotecas e aplicações da versão 2.72.1 do ARToolKit no Windows, por exemplo, são pré-requisitos:

- Microsoft Visual Studio .NET 2003 ou Visual Studio 6, ou Cygwin.
- DSVideoLib-0.0.8b-win32
- GLUT
- OpenVRML-0.16.1-bin-win32 ou OpenVRML-0.14.3-win32 (opcional)

No caso dessa versão 2.72.1 na plataforma Windows, é necessário seguir os seguintes passos para compilar os seus projetos:

1. Instalar um dos compiladores da Microsoft (Visual Studio .NET 2003 ou Visual Studio 6) ou o CygWin.
2. Baixar e descompactar a biblioteca “DSVideoLib-0.0.8b-win32” (DSVideoLib, 2007) dentro do diretório raiz “{ARToolKit}”, certificando-se que o diretório seja denominado “DSVL”. Copiar os arquivos “DSVL.dll” e “DSVLd.dll” para a pasta “bin” do ARToolKit (“{ARToolKit}\bin”).
3. Baixar o GLUT (GLUT, 2007), descompactar e copiar o arquivo “glut32.dll” para pasta “system” do Windows, copiar o arquivo “glut32.lib” para a pasta “lib” do ARToolkit (“{ARToolKit}\lib”) e criar uma pasta “GL” no diretório “include” do ARToolKit (“{ARToolKit}\include \GL”), copiando para essa nova pasta o arquivo “glut.h”.

4. Executar o script “Configure.win32.bat” que está em “{ARToolKit}\Configure.win32.bat” para criar o arquivo “{ARToolKit}\include\AR\config.h”.
5. Abrir o arquivo “ARToolKit.sln” no VisualStudio .NET ou o arquivo “ARToolkit.dsw” no Visual Studio 6.
6. Compilar os projetos.
7. Executar as aplicações geradas em “{ARToolKit}\bin”

Caso o usuário deseje compilar a aplicação “SimpleVRML”, que renderiza objetos virtuais implementados em VRML, deve-se :

8. Baixar a biblioteca “OpenVRML” (OpenVRML-0.16.1-bin-win32 ou OpenVRML-0.14.3-win32 (OpenVRML, 2007) e descompactá-la no diretório raiz do ARToolKit.
9. Copiar o arquivo “js32.dll” da pasta “{ARToolKit} \OpenVRML\bin” para a pasta “{ARToolKit}\bin”.
10. Habilitar e compilar, no VisualStudio .NET, os projetos “libARvrml” e “simpleVRML”. Já, no VisualStudio 6, é necessário criar um projeto para compilar a biblioteca estática gerada pelo “libARvrml” e um projeto de aplicação win32, com as devidas dependências ajustadas para gerar o executável “simpleVRML.exe”.

No momento de execução das aplicações, podem surgir mensagens de erro, acusando a falta de algumas dlls da Microsoft. Nesse caso, é necessário baixá-las de sites na Internet e copiá-las para a pasta “system32”, existente no diretório Windows.

Referências

DSVideoLib, <http://sourceforge.net/project/showfiles.php?group_id=75107&package_id=75491> acesso em fev, 2007.

GLUT, The OpenGL Utility Toolkit.< <http://www.xmission.com/~nate/glut.html>>
acesso em fev, 2007

OpenVRML< <http://www.openvrml.org/>> acesso em fev, 2007.

ANEXO 2- PUBLICAÇÕES

Este anexo apresenta as publicações realizadas durante o desenvolvimento desse trabalho no período de 2006 a 2007.

- SANTIN, R. ; KIRNER, C. . TÉCNICAS PARA FIXAR OBJETOS VIRTUAIS EM AMBIENTES DE AUTORIA COM REALIDADE AUMENTADA. In: III Workshop de Realidade Aumentada, 2006, Rio de Janeiro. Anais do III Workshop de Realidade Aumentada. Porto Alegre : SBC, 2006. v. 1. p. 63-66.
- SANTIN, R. ; KIRNER, C. . ARToolKit. In: Cardoso, A.; Kirner, C.; Lamounier, E.; Kelner, j.. (Org.). Tecnologias para o Desenvolvimento de Sistemas de Realidade Virtual e Aumentada. 1 ed. Recife-PE: Ed. Universitária da UFPE, 2007, v. 1, p. 91-109.
- KIRNER, C. ; SANTIN, R. ; KIRNER, Tereza Gonçalves ; ZORZAL, E. R. . Collaborative Augmented Reality Environment for Educational Applications. In: Ninth International Conference on Enterprise Information Systems, 2007, Funchal. Proceedings of the 9th International Conference on Enterprise Information Systems. Lisboa : INSTICC, 2007. v. HCI. p. 257-262.
- SANTIN, R. ; KIRNER, C. . CLASSIFICANDO AS INTERAÇÕES EM AMBIENTES DE REALIDADE AUMENTADA. In: IV Workshop de Realidade Virtual e Aumentada - WRVA 2007, 2007, Itumbiara - GO. Anais do IV Workshop de Realidade Virtual e Aumentada. Porto Alegre - RS : SBC, 2007. v. 1. p. 1-4.
- KIRNER, C.; OLIVEIRA, F, C.; SANTIN, R. . USO DO LIVRO INTERATIVO COM REALIDADE AUMENTADA EM APLICAÇÕES EDUCACIONAIS. In: IV Workshop de Realidade Virtual e Aumentada - WRVA 2007, 2007, Itumbiara - GO. Anais do IV Workshop de Realidade Virtual e Aumentada. Porto Alegre - RS : SBC, 2007. v. 1. p. 13-16.