



Universidade Federal de Pernambuco

Graduação em Ciência da Computação

Centro de Informática

*Arquitetura para Dispositivos Não-Convencionais de
Interação Utilizando Realidade Aumentada:
Um Estudo de Caso*

Aluna: Nacha Costa Bastos (ncb@cin.ufpe.br)

Orientadora: Judith Kelner (jk@cin.ufpe.br)

Co-Orientadora: Veronica Teichrieb (vt@dsc.upe.br)

Recife, Fevereiro de 2005.

Resumo

A área de Realidade Aumentada tem chamado muita atenção nos últimos anos, seja por ser uma área que pode ser aplicada em âmbitos diferentes, seja por usar tecnologias de ponta (por exemplo, dispositivos móveis) ou ainda por muitas vezes impor um certo avanço em alguns dispositivos. A área de Redes de Ambiente surge como um novo tópico de pesquisa por propor um conceito que há muito tem sido colocado como necessário. No momento em que se unem essas duas áreas muitas aplicações podem ser desenvolvidas, aplicações estas que possuem interfaces em Realidade Aumentada e que são suportadas por Redes de Ambiente. Neste trabalho foi proposta uma arquitetura que estará rodando sobre uma rede desse tipo, e esta arquitetura foi pensada para usar o que há de mais interessante na área de Realidade Aumentada em conjunto com uma série de dispositivos que estão no ápice de sua popularidade, os dispositivos móveis.

Agradecimentos

Antes de tudo e acima de qualquer um, meus pais merecem toda a gratidão que eu possa oferecer. Agradeço pelo apoio nos momentos que achei que não ia suportar tantas obrigações, momentos em que pensei que não era capaz de levar esse curso adiante, mas eles estavam sempre ali prontos para oferecer o carinho e apoio que precisava. Agradeço a meu pai que no momento que eu decidi interromper esse curso estava lá me apoiando, mesmo que a contragosto, mas sem nunca me pressionar a voltar. Agradeço a minha mãe por ser meu eterno ponto de equilíbrio e por tantas vezes colocar meus problemas acima dos dela.

Agradeço aos meus amigos que sempre me ajudaram nas horas difíceis, e com quem as horas alegres eram ainda mais divertidas. Agradeço em particular a dois deles que são quase como irmãos para mim e que me apoiaram em todos os momentos desde que entramos na faculdade. Obrigada Igor e Mouse.

Agradeço a Saulo por estar ao meu lado nesses últimos momentos tão difíceis, por ser meu porto e por tantas vezes agüentar minhas crises sem perder a calma e a paciência.

Agradeço também a minha co-orientadora Veronica, pois sem ela nada disso teria sido possível. Obrigada pelo incentivo, pela paciência e por sempre tentar entender o meu lado, me acalmando e me mostrando que as coisas não eram tão ruins quanto se pensava. Obrigada pelos elogios mesmo quando eu mesma achava que tudo que escrevia era ruim.

Agradeço a minha orientadora, professora Judith, por ter acreditado em mim e me permitido fazer parte do Grupo de Pesquisa em Redes e Telecomunicações, por sempre achar que nada era um grande problema e estar sempre disposta a ajudar.

E finalmente, agradeço aos pesquisadores que escreveram seus *papers* e os disponibilizam de alguma forma na Internet, e agradeço também ao Google que me ajudou a encontrá-los.

Índice

1	INTRODUÇÃO	8
2	REALIDADE AUMENTADA.....	9
2.1	Conceitos.....	9
2.2	Ferramentas e Bibliotecas.....	10
2.2.1	ARToolKit.....	10
2.2.1.1	ARToolKitPlus.....	12
2.2.2	ARTag.....	12
2.2.3	ARStudio.....	14
2.2.4	JARToolKit.....	15
2.2.5	DART.....	15
2.3	Áreas de Aplicação.....	15
2.3.1	Medicina.....	16
2.3.2	Manutenção.....	16
2.3.3	Visualização e Explicação.....	17
2.3.4	Entretenimento.....	18
2.3.5	Planejamento Urbano.....	21
3	INTERAÇÃO EM REALIDADE AUMENTADA.....	22
3.1	Técnicas de Interação Gerais.....	22
3.1.1	Técnicas de Interação para Manipulação 3D.....	22
3.1.1.1	Interação por Apontamento.....	22
3.1.1.2	Manipulação Direta.....	24
3.1.1.3	Mundo em Miniatura.....	25
3.1.1.4	Técnicas de Agregação e Integração.....	25
3.1.1.5	Manipulação Desktop 3D.....	27
3.1.2	Técnicas de Navegação.....	28
3.1.2.1	Técnicas de Locomoção Física.....	28
3.1.2.2	Técnicas de Direcionamento.....	30
3.1.2.3	Técnica de Planejamento de Rotas.....	32
3.1.2.4	Técnicas Baseadas em Alvo.....	32
3.1.2.5	Técnica de Manipulação Manual.....	33
3.1.2.6	Técnica Travel-by-Scaling.....	34
3.1.2.7	Técnicas de Orientação do Viewpoint.....	34
3.1.2.8	Técnica de Especificação da Velocidade.....	34
3.1.2.9	Controles Integrados da Câmera para Ambientes Desktop 3D.....	35
3.1.3	Técnicas de Controle de Sistemas.....	35
3.1.3.1	Técnica dos Menus Gráficos.....	35
3.1.3.2	Técnicas de Comando de Voz.....	37
3.1.3.3	Técnica de Comando de Gestos.....	38
3.1.3.4	Técnicas Ferramentas.....	38
3.1.4	Técnicas de Saída Simbólica.....	38
3.1.4.1	Técnicas Baseadas em Teclado.....	39
3.1.4.2	Técnicas Baseadas em Canetas.....	40
3.1.4.3	Técnicas Baseadas em Gestos.....	41
3.1.4.4	Técnicas Baseadas na Fala.....	41
3.2	Técnicas de Interação Específicas para Realidade Aumentada.....	42
3.2.1	Interação Espacial.....	42

3.2.2	Interação Baseada em Agentes	44
3.2.3	Interação de Controle Virtual	45
3.2.4	Interação de Controle Físico.....	45
3.3	Dispositivos de Interação	46
3.3.1	Sistemas de Exibição.....	46
3.3.2	Dispositivos Hápticos.....	49
3.3.3	Dispositivos Móveis	50
4	ARQUITETURA PARA DISPOSITIVOS NÃO-CONVENCIONAIS DE INTERAÇÃO UTILIZANDO REALIDADE AUMENTADA	53
4.1	Descrição do Cenário	53
4.2	Arquitetura	55
4.3	Dispositivos	57
4.3.1	Notebook	57
4.3.2	PDA	58
4.3.3	Celular.....	58
4.4	Bibliotecas	59
4.4.1	OGRE.....	59
4.4.2	ODE	59
4.4.3	CIDA.....	59
4.4.4	DirectX	60
4.4.5	OpenGL.....	60
4.4.5.1	OpenGL ES.....	60
4.4.6	Klont.....	60
4.5	Back-End	61
4.6	Infra-Estrutura de Comunicação	62
5	CONCLUSÃO E TRABALHOS FUTUROS	63
6	REFERÊNCIAS.....	64
6.1	Bibliografia.....	64
6.2	Referências Adicionais.....	67

Índice de Figuras

Figura 2.1 - Representação simplificada da continuidade virtual [30].....	9
Figura 2.2 - Aplicação <i>table-top</i> em AR [1].....	10
Figura 2.3 - Objeto virtual no marcador [25]	11
Figura 2.4 - Passos básicos do funcionamento do ARToolKit [5]	12
Figura 2.5 - Objetos 3D carregados no marcador ARTag [4]	13
Figura 2.6 - Marcadores ARTag [18]	14
Figura 2.7 - Marcadores ARStudio [28]	14
Figura 2.8 - <i>Screenshot</i> do DART [15]	15
Figura 2.9 - Imagem sobreposta ao paciente [22].....	16
Figura 2.10 - Encanamento industrial [7]	17
Figura 2.11 - <i>Signpost</i> no modo <i>stand-alone</i> (esquerda) e <i>Signpost</i> mostrando o mapa da estrutura do prédio (direita) [50]	18
Figura 2.12 - A propaganda <i>Pacific Bell</i> é virtual [16].....	18
Figura 2.13 - Visão do usuário jogando ARQuake [36].....	19
Figura 2.14 - Dois PDAs rodando o <i>Invisible Train</i> (acima) e a interface do jogo (abaixo) [47].....	19
Figura 2.15 - Aplicação <i>Magic Cubes</i> usada como contador de história [31]	20
Figura 2.16 - <i>Magic Table</i> em uso [8]	20
Figura 2.17 - <i>Workspace</i> para um planejamento urbano [12]	21
Figura 3.1 - Técnica Ray-Casting [41].....	23
Figura 3.2 - Técnica Simple Virtual Hand [41]	24
Figura 3.3 - Técnica Mundo em Miniatura [41].....	25
Figura 3.4 - Técnica HOMER [41]	26
Figura 3.5 - Técnica Voodoo Dolls [41]	27
Figura 3.6 - Sistema <i>battlefield</i> desenvolvido em AR [7].....	29
Figura 3.7 - Técnica Walking in Place [41]	29
Figura 3.8 - Técnica Marcando Pontos ao Longo do Caminho [41].....	32
Figura 3.9 - Técnica Especificação Baseada em Mapa ou em WIM [41].....	33
Figura 3.10 - Técnica Menu 2D Adaptado [41].....	36
Figura 3.11 - Técnica Menu 1-DOF [41]	36
Figura 3.12 - Técnica Menu TULIP [41]	37
Figura 3.13 - Técnica Menu 3D Widgets [41].....	37
Figura 3.14 - Técnica Ferramentas [41]	38
Figura 3.15 - Técnica Digital Ink [41]	41
Figura 3.16 - Aplicação metaDesk, vista do sistema (cima esquerda), lentes mostrando uma vista 3D (cima	

direita), objeto com invólucro de acrílico (baixo esquerda) e objetos físicos reais (baixo direita) [43].	43
Figura 3.17 - Sistema ARTHUR em funcionamento [11]	43
Figura 3.18 - Protótipo de um sistema para a navegação 3D [40]	44
Figura 3.19 - Sistema Welbo [3]	45
Figura 3.20 - Menus 3D representando comandos de um <i>software</i> CAD [11]	45
Figura 3.21 - Exemplo de controle físico, um <i>stage controller</i> [10]	46
Figura 3.22 - Diagrama conceitual do <i>optical see-through</i> HMD, baseado em [6]	47
Figura 3.23 - <i>Optical see-through</i> HMD [14]	47
Figura 3.24 - Diagrama conceitual do <i>video see-through</i> HMD, baseado em [6]	48
Figura 3.25 - <i>Video see-through</i> HMD [13]	48
Figura 3.26 - Sistema <i>monitor-based</i> , baseado em [6]	49
Figura 3.27 - <i>Data gloves</i> [21] [44]	50
Figura 3.28 - <i>Backpack</i> tradicional para AR móvel (esquerda) verso um <i>handheld</i> (direita) [47]	51
Figura 3.29 - Sistema Virtuoso [20]	51
Figura 3.30 - Aplicação AR <i>Comic Book</i> [41]	52
Figura 4.1 - Representação da rede interna do museu	54
Figura 4.2 - Arquitetura para dispositivos não-convencionais de interação utilizando AR	56
Figura 4.3 - Arquitetura interna do <i>laptop</i>	57
Figura 4.4 - Arquitetura interna do PDA	58
Figura 4.5 - Arquitetura interna do celular	59
Figura 4.6 – Comunicação entre dispositivo, PBMAN e BE	61

1 Introdução

Nos últimos anos duas áreas têm despontado na comunidade tecnológica, mas ainda são pouco exploradas, despertando um maior interesse por parte da comunidade acadêmica.

Uma dessas áreas é a de Realidade Aumentada (*Augmented Reality - AR*), que está sendo colocada como sendo uma expansão da Realidade Virtual. Muito tem sido comentado sobre esta área, mas ainda hoje muitas definições continuam em aberto, talvez pela novidade desta e pelo pequeno limiar entre algumas áreas correlatas. Mas, simplesmente falando, AR permite que o usuário veja o mundo real alguns objetos virtuais inseridos nele, permitindo com isso uma maior interatividade entre o usuário e o ambiente, assim como uma maior percepção do mesmo.

Dentro da área de AR muito tem sido estudado e proposto. Algumas dessas propostas em áreas completamente diferentes como a área médica, educacional e principalmente de jogos.

A outra área que está despontando no momento são as Redes de Ambiente (*Ambient Network - AN*). Estas são baseadas em um conjunto de múltiplas redes construídas com diferentes tecnologias. O objetivo das AN é unir os diferentes tipos de redes, tais como BANs (*Body Area Networks*), VANs (*Vehicular Area Networks*), PANs (*Personal Area Networks*), HANs (*Home Area Networks*), redes de acesso sem fio, redes celulares, entre muitas outras.

O Ambient Networks Project pretende criar uma solução para redes de sistemas móveis e sem fio, além do 3G (protocolos de terceira geração). Esta solução tem por objetivo capacitar uma rede de comunicação sem fio escalável e de baixo custo, enquanto fornecerá serviços de comunicação fartos e acessíveis para todos, gerados através do incremento de competição e cooperação em um ambiente repleto por uma grande variedade de dispositivos, tecnologias sem fio, operadores de redes de comunicação e empresários [2]. Esse projeto está sendo dividido em "WorkPackages" e patrocinado por um consórcio constituído por grandes empresas de telecomunicações, entre as quais Nokia, Ericsson e Siemens.

Essas duas áreas relativamente novas mais o avanço, principalmente em termos de *hardware*, dos dispositivos ditos móveis permite o desenvolvimento de inúmeras aplicações nas mais diversas áreas. Com isto este trabalho tem por objetivo propor um novo conceito de interfaces de serviços baseadas em AR a partir de uma proposta de arquitetura para abordar problemas de interação utilizando dispositivos como celulares e PDAs (*Personal Digital Assistant*).

Além disto, pretende-se apresentar um protótipo de sistema para utilização em museus. Este sistema conterá módulos de visualização de conteúdo, troca de informações via *chat* e conferência.

Basicamente este trabalho se divide em dois capítulos que mostrarão o estado da arte, um capítulo que detalhará a arquitetura proposta e o estudo de caso, e um capítulo de conclusão. No primeiro capítulo sobre o estado da arte (capítulo 2) serão explicados alguns conceitos para que o leitor possa perceber as sutis diferenças entre os diversos conceitos em que a AR está sedimentada, algumas das ferramentas e bibliotecas mais usadas e conhecidas e quais as áreas onde a AR tem se destacado. No capítulo seguinte (capítulo 3), serão mostradas as técnicas mais usadas para a concepção de interfaces tridimensionais (3D) em VR, algumas técnicas que são usadas exclusivamente para a concepção de interfaces em AR e alguns dispositivos que são mais usados para interação com essas interfaces específicas em AR. Já no capítulo 4, será proposta uma arquitetura e serão mostrados alguns detalhes desta, o estudo de caso que foi escolhido para este trabalho será explanado e algumas particularidades da arquitetura serão explicadas. Por fim, no capítulo 5, serão mencionadas as principais conclusões deste trabalho e alguns possíveis trabalhos futuros.

2 Realidade Aumentada

Na seqüência serão definidos alguns conceitos relativos à AR, bem como outros considerados importantes para o entendimento deste Trabalho de Graduação. Estes conceitos serão referenciados em todos os capítulos seguintes do trabalho.

2.1 Conceitos

No contexto deste trabalho, o conceito mais amplo é o de Realidade Mista (*Mixed Reality* - MR). MR envolve a união do mundo real com o mundo virtual em algum lugar na “continuidade virtual”, como pode ser visto na Figura 2.1, a qual conecta completamente o ambiente real e o ambiente virtual [30] Em outras palavras, MR é popularmente definida como sendo uma tecnologia que mistura o mundo virtual como, por exemplo, gráficos e/ou imagens geradas por computador, e o mundo real tão naturalmente quanto possível, em tempo real.

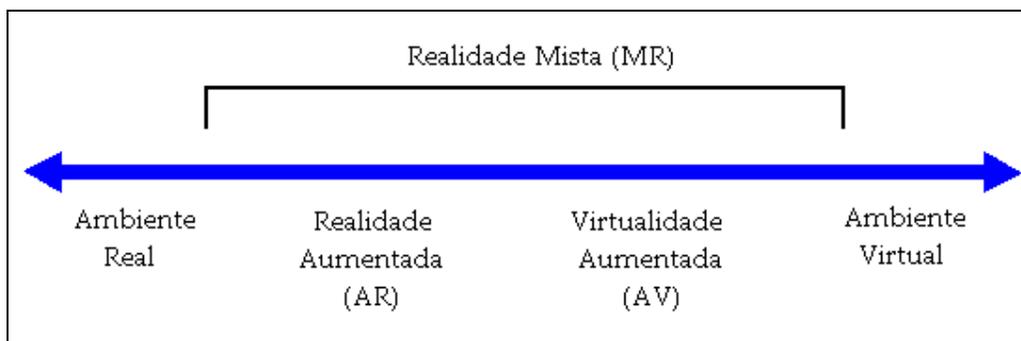


Figura 2.1 - Representação simplificada da continuidade virtual [30]

Realidade Virtual (*Virtual Reality* - VR) nada mais é do que uma simulação em terceira dimensão do mundo real ou de um mundo imaginário qualquer. Simulação esta que é imersiva e interativa, ou seja, através de HMDs (*Head-mounted Displays*), luvas especiais e fones de ouvido, entre outros dispositivos, o usuário tem a sensação de estar em outro mundo, onde sensações de tato, visão e audição podem ser sentidas em tempo real [30]. VR está representada na continuidade virtual na extrema direita, exatamente no oposto dos ambientes reais.

Já AR é uma evolução da VR. Mas, ao contrário do que acontece nesta última, em AR o usuário vê o mundo real com objetos virtuais inseridos neste ou ainda escondendo objetos reais, ou seja, AR complementa o mundo real ao invés de sobrepor-lo [6]. Como ocorre com VR, AR também pode envolver os sentidos da audição e tato, e não somente visão [7]. Como o nome já diz, AR aumenta o ambiente para o usuário, sendo capaz de aumentar a percepção do usuário e a sua interação com o mundo real, provendo informações que este por si só não é capaz de perceber apenas com seus próprios sentidos. Estas informações, conduzidas através de objetos virtuais, ajudarão no desempenho das tarefas do mundo real. A figura abaixo ilustra uma aplicação *table-top* em AR que consiste em prédios virtuais do centro de Augsburg - Alemanha projetados sobre o mapa real da cidade.

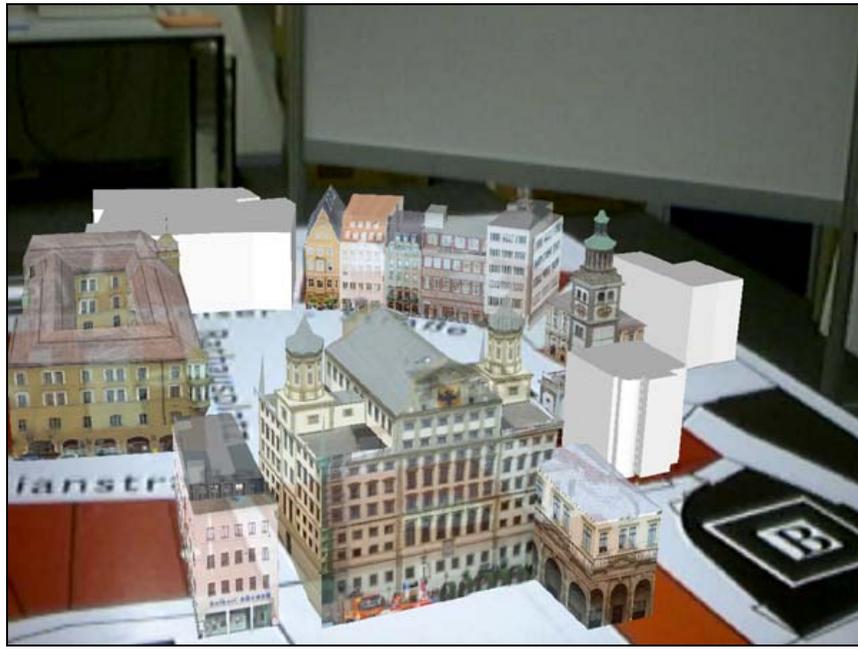


Figura 2.2 - Aplicação *table-top* em AR [1]

Virtualidade Aumentada (*Augmented Virtuality - AV*) é em muito parecida com AR, mas ao contrário desta última, em AV o usuário está imerso em um mundo completamente virtual onde serão aplicadas texturas do mundo real em alguns objetos gráficos 3D. Ou seja, AV é uma simulação imitando o mundo real para apoiar uma experiência imersiva, tal qual a “matrix” do filme homônimo.

2.2 Ferramentas e Bibliotecas

Nesta seção serão mostradas algumas das ferramentas e bibliotecas mais utilizadas no desenvolvimento de aplicações em AR. Não é intuito deste trabalho mostrar todas as bibliotecas e ferramentas existentes e sim as mais conhecidas e divulgadas.

2.2.1 *ARToolKit*

O *ARToolKit* foi originalmente desenvolvido para servir de apoio na concepção de interfaces colaborativas [24] pelo Dr. Hirokazu Kato, na Universidade de Osaka. E, desde então, tem sido mantido pelo *Human Interface Technology Laboratory* (HIT Lab) da Universidade de Washington e pelo HIT Lab NZ da Universidade de Canterbury, em Christchurch.

O *ARToolKit* é uma biblioteca de software *open source*, escrita na linguagem C, para concepção de aplicações em AR. Um exemplo de uma possível aplicação pode ser um personagem virtual 3D que aparecerá em cima de um marcador, como pode ser visto na Figura 2.3. Este marcador vem a ser apenas um cartão onde será impresso um padrão que corresponde a uma imagem específica. Esta imagem pode ser visualizada através de um *display* que o usuário estará usando, e à medida que o marcador é movimentado a imagem também é movida dando a impressão de estar colada ao marcador.

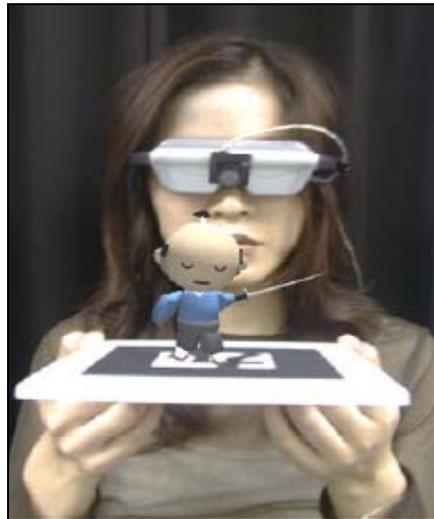


Figura 2.3 - Objeto virtual no marcador [25]

Uma das grandes dificuldades ao se desenvolver aplicações em AR é o problema em calcular o *viewpoint* do usuário em tempo real, de modo que as imagens virtuais estejam exatamente alinhadas com os objetos reais, ou seja, o problema é saber para onde o usuário estará olhando em um determinado momento para que a imagem seja projetada exatamente no lugar certo. O ARToolKit se propõe a resolver este problema usando uma técnica que calcula a posição real da câmera presa ao *display* e a orientação relativa a marcadores, permitindo então que o programador possa sobrepor objetos virtuais nestes.

As principais características do ARToolKit são: *tracking* para posicionamento e orientação de uma única câmera; código que usa como padrão um quadrado de bordas pretas; possibilidade de uso de qualquer marcador, desde que o mesmo obedeça ao padrão; código simples para calibração da câmera; rápido para aplicações em AR em tempo real; distribuição multi-plataforma (por exemplo, SGI IRIX, Linux, MacOS e Windows) e distribuição do código fonte completo.

O funcionamento do ARToolKit é muito simples. Primeiramente, a câmera posicionada no *display* do usuário captura as imagens do mundo real e as envia para um computador, onde este procurará nas imagens *frames* que contenham o padrão do quadrado de bordas pretas (um marcador). Se um marcador for encontrado, o computador efetuará uma série de cálculos matemáticos para saber qual a posição da câmera relativa ao marcador visualizado. Uma vez que a posição da câmera é conhecida um modelo computacional gráfico (objeto virtual) será desenhado na mesma posição. A saída, portanto, nada mais é do que esta imagem virtual “grudada” em cima do marcador, que o usuário poderá visualizar através do *display* que está usando [5]. Este passo a passo pode ser mais bem visualizado na figura abaixo.

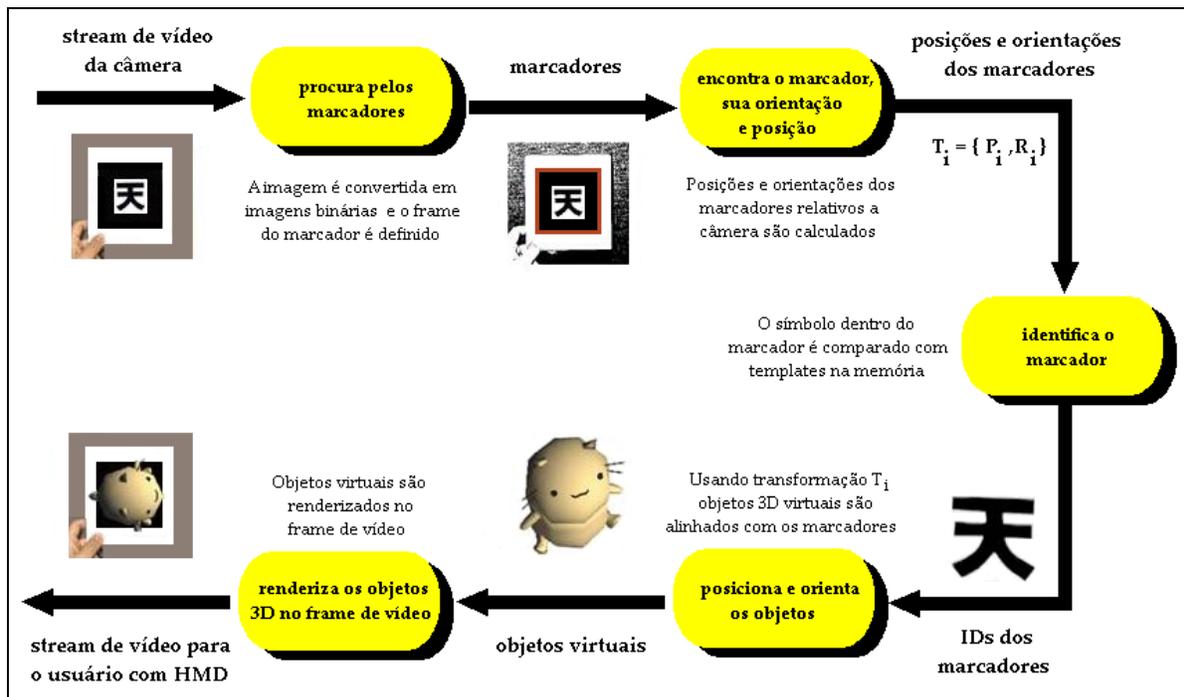


Figura 2.4 - Passos básicos do funcionamento do ARToolKit [5]

2.2.1.1 ARToolKitPlus

O ARToolKitPlus foi desenvolvido na *Graz University Technology* dentro do projeto *Studierstube*, e é uma biblioteca *open source* de AR baseada no ARToolKit, embora a primeira se preocupe exclusivamente com a questão da detecção de marcadores, não oferecendo funções para captura de vídeo ou renderização de objetos 3D [50] [46].

Seu código fonte conta com várias otimizações, tais como, utilização de matemática de ponto fixo com o intuito de gerar aplicações eficientes para os dispositivos móveis como, por exemplo, PDAs e *smartphones*. Os marcadores utilizados por esta biblioteca são semelhantes aos do ARToolKit, com a diferença que o desenho no interior do quadrado de bordas pretas consiste em uma codificação do identificador do marcador. Tal codificação permite que o usuário possa utilizar até 512 diferentes marcadores, diminuindo a ocorrência da confusão entre marcadores diferentes.

O ARToolKitPlus utiliza a técnica de limiar adaptativo, que consiste em perceber alterações de iluminação no ambiente capturado pela câmera, de forma que a detecção dos marcadores não seja prejudicada. Esta biblioteca também oferece suporte ao usuário de utilizar marcadores com borda de largura variável, desta forma pode-se diminuir a largura da borda para melhor aproveitar o espaço para codificação, fazendo com que marcadores menores possam ser detectados. Outra característica consiste na capacidade de realizar uma compensação sobre imagens de câmeras que geram uma queda radial de luminância. Este fato ocorre em algumas câmeras onde a imagem capturada por elas se apresenta mais escura nos cantos, fazendo com que os marcadores situados nessas regiões não seja detectados.

2.2.2 ARTag

O ARTag foi desenvolvido pelo *National Research Council of Canada* e inspirado no ARToolKit. O ARTag, tal como o ARToolKit, também consiste em uma biblioteca de padrões, mas diferentemente deste que compara imagens, o ARTag compara códigos digitais compostos de 0's e 1's. O marcador do ARTag também vem a

ser um quadrado de bordas pretas, mas seu interior é preenchido com uma malha 6x6 de quadrados pretos e brancos que representam os códigos digitais. Abaixo pode ser vista uma figura que mostra objetos sendo carregados em marcadores ARTag.

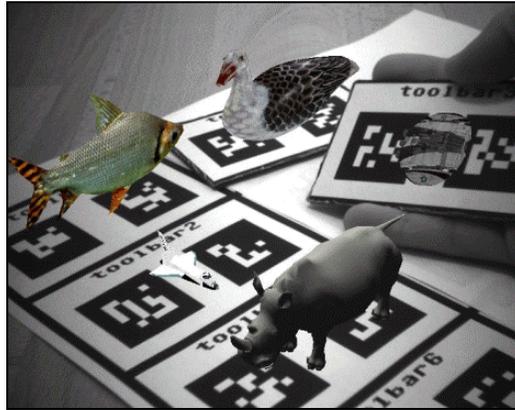


Figura 2.5 - Objetos 3D carregados no marcador ARTag [4]

O ARTag foi concebido para não conter alguns dos problemas que o ARToolKit possui. Entre esses problemas pode-se destacar três que são mais comuns, e que no ARTag foram corrigidos:

- *false positive* -> acontece quando a presença de um marcador é reportada, mas não existe nenhum no ambiente;
- *inter-marker confusion*-> acontece quando a presença de um marcador é reportada, mas com o ID errado;
- *false negative* -> acontece quando existe um marcador no ambiente, mas sua presença não é reportada.

Além de não possuir esses problemas mais comuns presentes no ARToolKit, o ARTag ainda tem a vantagem de possuir 2002 marcadores. Dentre esses 1001 têm o mesmo padrão do ARToolKit, um quadrado de bordas pretas e interior branco, enquanto que os restantes 1001 têm o padrão inverso, um quadrado de bordas brancas e interior preto [18], vide Figura 2.6. Uma outra vantagem sobre o ARToolKit é que ao contrário desse, no ARTag não é necessário que um arquivo contendo todos os padrões seja previamente carregado. Além do que os marcadores do ARTag podem ser reconhecidos mesmo que uma parte deles esteja oculta, o que não acontece com os marcadores do ARToolKit.

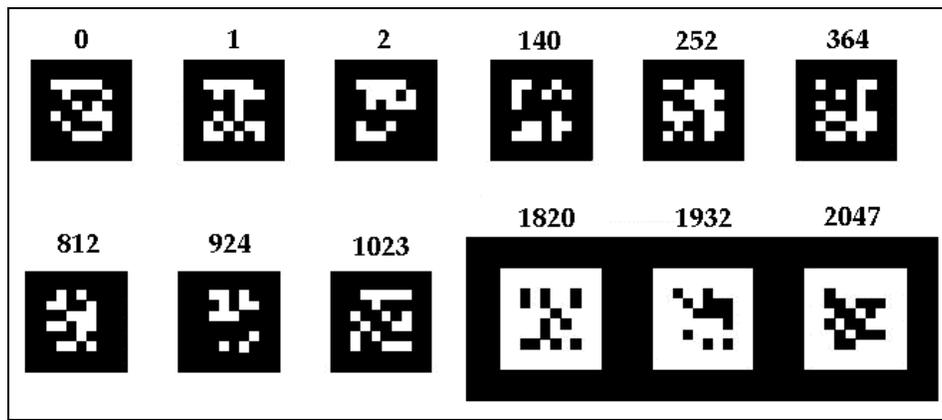


Figura 2.6 - Marcadores ARTag [18]

O processo para o reconhecimento de um marcador no ARTag começa basicamente do mesmo modo que no ARToolKit, primeiramente deve ser localizado o contorno da borda do quadrado. Após isso, a região interna que contém a malha é retirada e são determinados os códigos de 0's e 1's contidos. Todo o processamento subsequente de identificar e verificar o marcador são feitos digitalmente. Quatro seqüências binárias de 36 bits são obtidas do *array* de códigos binários originários da malha, uma para cada uma das quatro possíveis posições rotacionadas dentre essas apenas uma será validada pelo processo de decodificação. A seqüência binária codificada no marcador encapsula um ID de 10 bits usando métodos digitais, os 26 bits extras provêm redundância para reduzir as chances de detecção e identificação falsas e também para prover unicamente uma das quatro possíveis rotações [17].

2.2.3 ARStudio

O ARStudio também foi desenvolvido pelo *National Research Council of Canada* e possui basicamente os mesmos princípios do ARTag. O ARStudio, tal como os anteriores, também tem como padrão um quadrado de bordas pretas, mas no seu interior se encontram retângulos brancos igualmente distribuídos, vide figura abaixo.



Figura 2.7 - Marcadores ARStudio [28]

Para o reconhecimento dos marcadores, os quatro cantos de cada retângulo são armazenados em um arquivo de dados que é associado a um padrão. Embora não seja definido o arranjo dos retângulos nos marcadores, é importante que ao gerar os padrões eles sejam unicamente identificáveis com respeito aos padrões já existentes, e também com relação às rotações [28].

2.2.4 JARToolKit

O JARToolKit foi desenvolvido pelo C-Lab alemão, e nada mais é do que um invólucro escrito na linguagem Java para o ARToolKit, ou seja, ele consiste em várias camadas que irão prover abstrações para as bibliotecas integradas do ARToolKit. O JARToolKit acessa as funções do ARToolKit usando a *Java Native Interface* (JNI) [19].

2.2.5 DART

O *Designers Augmented Reality Toolkit* (DART) é um conjunto de ferramentas que provêem um desenvolvimento rápido em AR, e foi desenvolvido pelo GVV Center no *Georgia Institute of Technology*.

O DART suporta inúmeros tipos de aplicações em AR, e tem como objetivo permitir que *designers*, artistas e pesquisadores de outras áreas tenham experiências com aplicações em AR. Por essa razão, o DART foi desenvolvido como uma coleção de extensões do ambiente de programação multimídia Macromedia Director, consistindo de *behaviors* (extensões do Director escritas na linguagem LINGO) e *Xtras* (*plugins* para o Director escritos na linguagem C++) [27]. Além disso, o DART utiliza uma parte do ARToolKit para captura de vídeo, *tracking* e reconhecimento de marcadores.

O DART foi desenvolvido para uma classe específica de experiências em AR, onde uma mídia gerada por computador é diretamente integrada com a percepção dos participantes. Nesse tipo de experiência, o usuário usa um HMD (vide seção 3.3) que permite uma integração entre computação gráfica e som no mundo físico real. O DART é mais usado para prototipação e experimentação de sistemas em AR ditos “mundanos”, como por exemplo, suporte no reparo de equipamentos, treinamento baseado em situações e turismo aumentado [15].

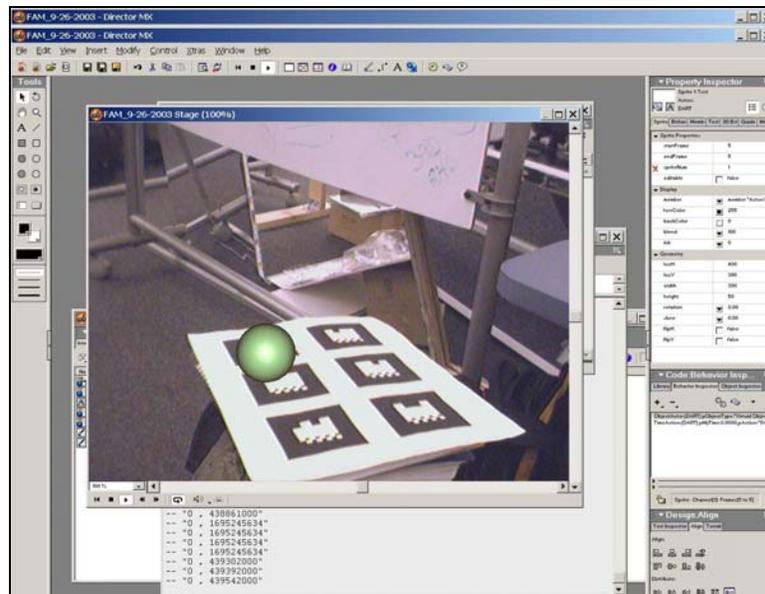


Figura 2.8 - Screenshot do DART [15]

2.3 Áreas de Aplicação

Nesta seção serão mostradas algumas áreas nas quais AR tem se destacado. As aplicações em AR são inúmeras e abrangem diversas áreas desde aplicações na área de entretenimento passando por aplicações

médicas e mais recentemente aplicações móveis e comerciais.

2.3.1 Medicina

Tecnologias que provêm o uso eficiente de imagens são de extrema importância para a área médica, talvez por isso existam inúmeras pesquisas desenvolvendo AR para esta área. A maioria das aplicações médicas são feitas com a orientação de imagens cirúrgicas. Estudos de imagens no pré-operatório como, por exemplo, tomografia computadorizada, ressonância magnética e ultra-som (sensores não-invasivos), provêm ao cirurgião a visão necessária da anatomia interna do paciente, e é através do estudo dessas imagens que a cirurgia é planejada. Os médicos podem usar a tecnologia de AR para a visualização dessas cirurgias. Com o conjunto dos dados coletados através dos sensores não-invasivos, esses dados podem ser renderizados e combinados em tempo real dando ao médico uma espécie de visão de raio-X dos órgãos do paciente, resultando com isso em uma visão interna sem a necessidade de grandes incisões [7], vide figura abaixo.

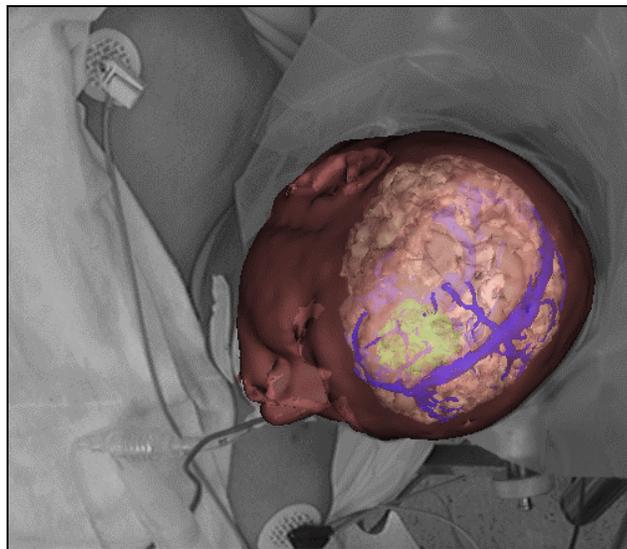


Figura 2.9 - Imagem sobreposta ao paciente [22]

Além das cirurgias, uma outra grande aplicação são as imagens de ultra-som. Usando um *display* o médico poderá ver a imagem do feto renderizada sobreposta ao abdômen da paciente grávida, parecendo assim que a imagem está dentro da barriga, já que a mesma é renderizada em tempo real à medida que o feto se move [22]. AR também pode ser usada para o treinamento de cirurgiões novatos. Instruções virtuais poderiam ir lembrando ao médico os passos requeridos sem a necessidade de que o mesmo deixe de olhar o paciente para ler um manual [7].

Entre as inúmeras vantagens das aplicações de AR na área médica as mais visíveis dizem respeito à fidelidade das imagens coletadas, já que as mesmas são feitas em tempo real na sala de cirurgia, aumentando com isso o desempenho de toda a equipe cirúrgica e também propiciando a eliminação da necessidade dos procedimentos dolorosos e enfadonhos.

2.3.2 Manutenção

Com o intuito de ajudar em reparos e manutenções de um modo geral, pesquisas tem sido feitas para o desenvolvimento de aplicações em AR. Para facilitar o entendimento das instruções, ao invés de ler manuais

e observar figuras nestes, objetos 3D podem ser sobrepostos a um equipamento qualquer mostrando passo a passo as tarefas que devem ser feitas e como fazê-las. Estes objetos podem ainda ser animados para que os passos sejam mostrados de uma maneira mais explícita.

Algumas aplicações já existentes consistem na manutenção de uma impressora a laser [6], e ainda na manutenção de um encanamento industrial [7], onde são visualizados um mapa 2D e um modelo 3D, vide figura abaixo.



Figura 2.10 - Encanamento industrial [7]

2.3.3 Visualização e Explicação

Uma aplicação muito usada em AR consiste em colocar pequenas notas em objetos e ambientes, notas estas que contém informações públicas e particulares. Caso a aplicação venha a ter informações públicas é necessário que haja uma disponibilidade de bases de dados, caso as informações tenham um caráter privado, as mesmas serão anexadas a objetos específicos [6].

O interessante nesse tipo de aplicação é a ajuda em tarefas cotidianas. Como por exemplo, a aplicação *SignPost* [51] que guia o usuário através de um prédio desconhecido mostrando uma variedade de sugestões para a navegação, podendo inclusive prover uma visualização da estrutura do prédio destacando elementos relevantes e a próxima saída a tomar, vide Figura 2.11. Uma outra aplicação similar a essa é um guia para museus, que mostra várias informações à medida que o usuário caminha pelo museu [49]. Ambas aplicações são também aplicações móveis, ou seja, usam os chamados PDAs.

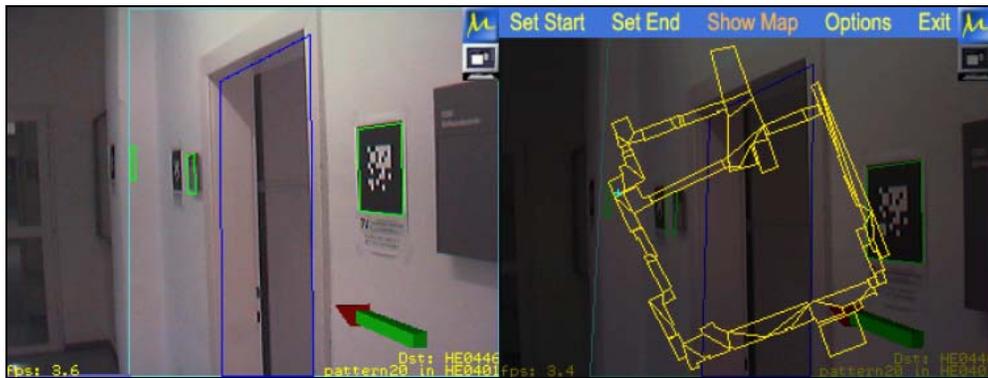


Figura 2.11 - *Signpost* no modo *stand-alone* (esquerda) e *Signpost* mostrando o mapa da estrutura do prédio (direita) [51]

Outra aplicação que vem se tornando comum é na área comercial, mais explicitamente através de um vídeo em tempo real usado como propaganda virtual em um *outdoor* [7], vide figura abaixo.



Figura 2.12 - A propaganda *Pacific Bell* é virtual [16]

2.3.4 *Entretenimento*

As aplicações na área de entretenimento são as mais diversas possíveis, desde jogos dos mais variados tipos até “contadores” de histórias. Entre os jogos criados com AR podemos destacar o ARQuake [36], que foi desenvolvido baseado no jogo Quake, originalmente implementado para plataforma *desktop*. ARQuake é jogado no mundo real dando ao usuário a mobilidade para se mover por onde desejar tudo que é visto é determinado exclusivamente pela orientação e posição da cabeça do usuário que está usando um HMD. Vide figura abaixo.



Figura 2.13 - Visão do usuário jogando ARQuake [36]

Outra aplicação que mistura entretenimento e aplicação móvel é o *Invisible Train* [48]. Esse jogo foi desenvolvido primeiramente para crianças de nível primário, e é um jogo multi-player no qual os jogadores guiam um trem virtual através de um trilho real construído em miniatura. Este trem só é visível para os jogadores através de PDAs, e para esses usuários são permitidas duas ações: operar as junções entre os trilhos e mudar a velocidade do trem, tudo isto através das telas *touch screen* dos próprios PDAs. Vide figura abaixo.



Figura 2.14 - Dois PDAs rodando o *Invisible Train* (acima) e a interface do jogo (abaixo) [48]

Outra aplicação bastante interessante na área de jogos é o CamBall [52]. Este jogo nada mais é do que um simples jogo de tênis, sendo o seu diferencial que os dois jogadores estão jogando através de uma *Local Area Network* (LAN) ou Internet e com raquetes reais. Os jogadores se vêem através de computadores e nas raquetes são colocados marcadores cujas posições serão capturadas por *webcams* instaladas em cada computador.

Saindo um pouco da área de jogos, mas ainda como entretenimento foi desenvolvido o projeto *Magic Cubes* [53]. Este foi desenvolvido para promover interações físicas e sociais dentro das famílias. O *Magic Cubes* consiste basicamente em marcadores em formato de cubo que contam histórias, como se fossem um livro de histórias infantis, servindo também para brincar de casinha, montando a mobília para quartos e salas, e por fim ainda servem para jogos de tabuleiro.

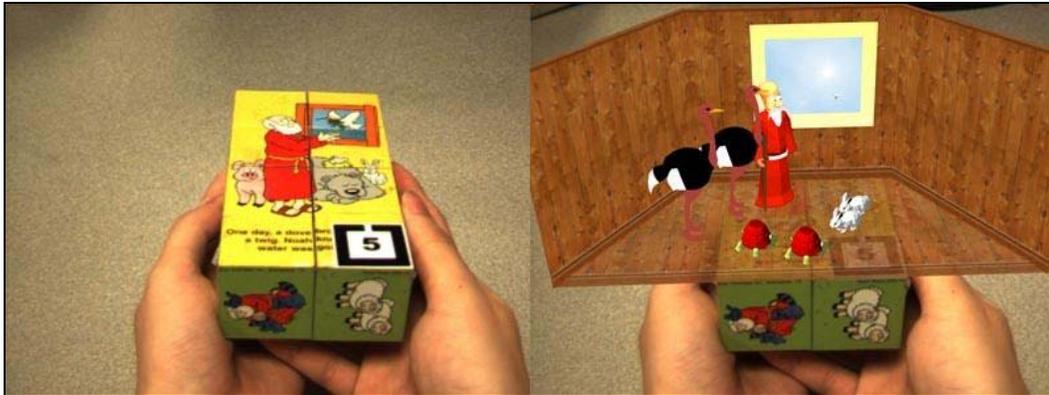


Figura 2.15 - Aplicação *Magic Cubes* usada como contador de história [31]

Por fim tem-se o *Magic Table* [8], que vem a ser um quadro branco onde se escreve, desenha e apaga. A diferença entre um quadro branco comum e o *Magic Table* consiste em *scanners* que capturam o que é escrito no quadro; tudo que foi escrito então é colocado em retângulos (*patches*) que são manipulados através de pequenos círculos (*tokens*) sobre o quadro, vide figura abaixo.

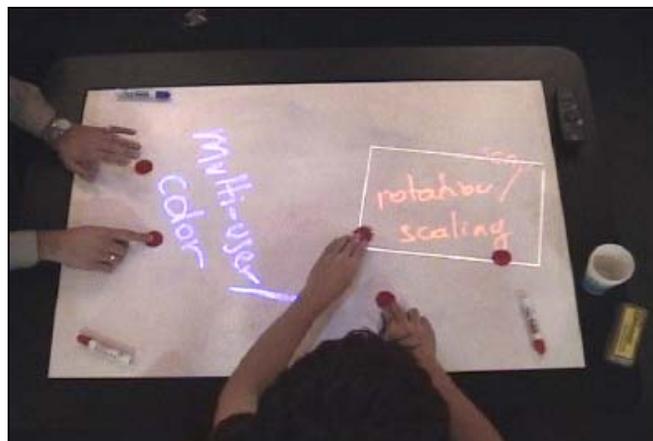


Figura 2.16 - *Magic Table* em uso [8]

2.3.5 Planejamento Urbano

Arquitetos, membros de conselhos de cidade e grupos de interesse são alguns dos muitos outros tipos de usuários que se beneficiam da AR no planejamento urbano, podendo discutir alternativas à medida que visualizam a cidade a sua frente. Dentro dessa área tem-se o Urp [43] que simula prédios e ventos, tem-se também um modelo um pouco mais simples [12] que possui prédios que podem ser facilmente movidos e animados. A grande vantagem em usar AR no planejamento das cidades é a facilidade de interação e visualização da visão de cada usuário, bastando para isso apenas pegar e arrastar os prédios como se os mesmos fossem simples caixas.

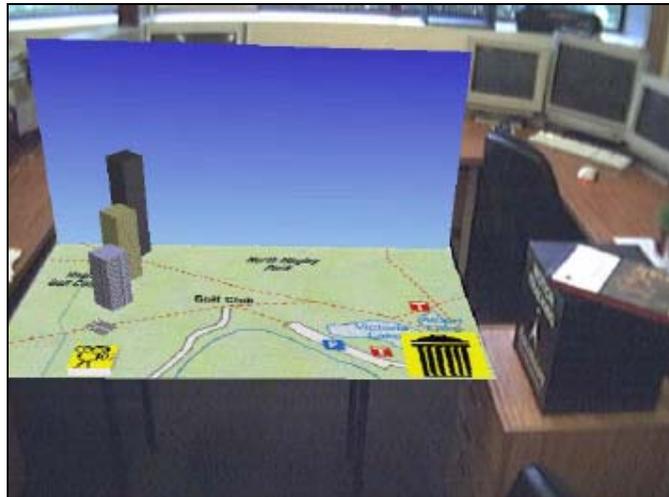


Figura 2.17 - *Workspace* para um planejamento urbano [12]

3 Interação em Realidade Aumentada

Neste capítulo serão abordadas algumas técnicas de interação que são mais usadas nos ambientes de VR, mas que podem ser também usadas em AR. Também serão abordadas técnicas desenvolvidas mais especificamente para ambientes de AR. E por fim, ainda serão mostrados os dispositivos mais usados em ambientes de AR, que também podem ser encontrados em ambientes de VR.

3.1 Técnicas de Interação Gerais

As técnicas abordadas nesta seção são geralmente mais usadas em ambientes de VR, não impedindo que algumas delas também sejam usadas em ambientes de AR. Esta seção em especial foi baseada no livro *3D User Interfaces: Theory and Practice* de Doug A. Bowman et al. [9], que classifica as técnicas de interação em técnicas para manipulação 3D, navegação, controle de sistemas e de saída simbólica.

3.1.1 Técnicas de Interação para Manipulação 3D

O desenvolvimento de técnicas para manipulação 3D é uma importante área de pesquisa dentro da VR. O desafio deste desenvolvimento consiste em basicamente fazer um mapeamento geométrico de corpos e objetos, algumas vezes necessitando se desviar do isomorfismo geométrico por causa dos limites de *hardware* ou pela complexidade do corpo humano.

Até o tópico 3.1.1.5 as técnicas que serão apresentadas vão tratar de seleção e translação de objetos, e a rotação será mapeada diretamente do dispositivo de entrada para o objeto virtual, o que é chamado isomorfismo. Estas técnicas são as por apontamento, por manipulação direta, em um mundo em miniatura e por agregação e integração. A partir do tópico 3.1.1.5, que apresenta técnicas de manipulação para *desktop* 3D, as técnicas usarão a chamada rotação 3D não-isomórfica, que utiliza a matemática dos quatérnios, que é uma matemática não intuitiva. Os cálculos matemáticos envolvidos foram omitidos a fim de se mostrar apenas uma visão geral das técnicas.

3.1.1.1 Interação por Apontamento

Este tipo de interação permite ao usuário facilmente selecionar e manipular objetos localizados fora da sua área de alcance bastando simplesmente apontar para eles.

Este tipo de interação se dá do seguinte modo: quando o vetor definido pela direção do apontamento interceptar um objeto virtual, o usuário poderá selecioná-lo bastando para tanto disparar um evento (por exemplo, apertar um botão, comando de voz) que confirmará a seleção. Após o objeto ser selecionado, ele poderá ser preso no final do vetor de apontamento para que então o usuário possa manipulá-lo.

A partir deste princípio algumas técnicas foram desenvolvidas. Abaixo serão explicadas as mais usadas e conhecidas no meio, a saber, Ray-Casting, Two-Handed, Flashlight e Aperture, Image-Plane e Fishing-Reel.

RAY-CASTING

O funcionamento desta técnica é muito simples; basicamente o usuário apontará para um objeto e um raio virtual definirá a direção do apontamento. Um segmento de reta virtual preso à mão servirá para visualizar esta direção. Caso mais de um objeto seja interceptado pelo raio virtual, apenas o mais próximo ao usuário

deverá ser selecionado.

Em ambientes imersivos, o raio pode ser preso diretamente à mão virtual controlada por um sensor, já em ambientes *desktop* 3D, o raio pode ser preso a um objeto 3D que pode ser controlado por um *mouse*. Ray-Casting é uma das técnicas mais simples e eficientes, e apresenta problemas se houver a necessidade de selecionar objetos muito pequenos ou objetos que estejam muito longe do usuário.

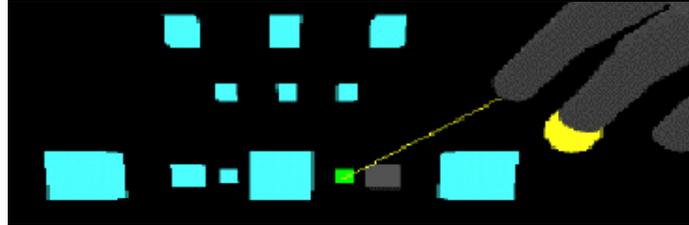


Figura 3.1 - Técnica Ray-Casting [41]

TWO-HANDED

Esta técnica é muito parecida com a anterior só que consiste no uso das duas mãos do usuário. Enquanto que uma mão, geralmente a mais próxima ao usuário, especifica a origem do raio virtual a outra especifica para onde o raio estará apontando.

A desvantagem desta técnica é que as duas mãos estarão sendo usadas. Contudo, isto permite interações mais ricas e eficientes, principalmente quando existem objetos completamente ou parcialmente ocultos ou quando os objetos para os quais se quer apontar estão em direções profundas.

FLASHLIGHT E APERTURE

A técnica Flashlight ou Spotlight foi desenvolvida com o intuito de prover uma seleção suave que não requer precisão ao apontar para objetos virtuais com o raio. Esta técnica imita o apontar para objetos com uma lanterna, a qual pode iluminar um objeto mesmo quando não é apontada diretamente para ele. A direção do apontamento é definida do mesmo modo que na técnica Ray-Casting, mas neste caso o raio virtual é substituído por um volume cônico.

Esta técnica permite uma seleção fácil de pequenos objetos, mesmo se estes estiverem a uma grande distância do usuário. O problema acontece quando mais de um objeto se encontra dentro do *spotlight*, pois o objeto a ser escolhido será aquele que estiver mais próximo da linha central do volume cônico.

Já a técnica Aperture é uma modificação da técnica Flashlight, que permite ao usuário controlar interativamente a abertura do ângulo do volume cônico.

A escolha entre uma das duas técnicas depende dos requisitos da aplicação, mas geralmente se os objetos não estão agrupados a técnica Flashlight é mais simples e efetiva, enquanto que se houver a necessidade de um controle maior a técnica Aperture é mais indicada.

IMAGE-PLANE

Esta família de técnicas provê ao usuário seleção e manipulação de objetos 3D apenas tocando e manipulando suas projeções 2D em uma imagem do plano virtual localizada na sua frente.

A técnica do Image-Plane simula toque direto, por isso é mais intuitiva e fácil de usar para a seleção. Seu problema consiste no fato de que a distância dos objetos para o usuário não pode ser controlada.

FISHING-REEL

Um grande problema das técnicas de apontamento é exatamente controlar a distância do usuário para os objetos virtuais que estão sendo manipulados. Esta técnica foi desenvolvida com o intuito de solucionar este problema.

A técnica Fishing-Reel é similar ao funcionamento de uma vara de pescar, como o próprio nome diz. Ela permite ao usuário selecionar um objeto usando a técnica Ray-Casting e puxá-lo para perto usando um dispositivo de entrada (i.e. um indicador, um par de botões), como se o usuário estivesse pescando o objeto.

3.1.1.2 Manipulação Direta

As técnicas de manipulação direta são também denominadas de técnicas da mão virtual, pois o usuário pode selecionar e manipular diretamente objetos virtuais com as suas mãos. Para isto é usado um cursor 3D que pode ser modelado como uma mão humana virtual.

A posição e a orientação do dispositivo de entrada são mapeadas na posição e orientação da mão virtual. A fim de selecionar um objeto o usuário simplesmente intercepta o cursor com o objeto desejado e dispara um evento (i.e. apertar um botão, comando de voz, gestos); o objeto será preso ao cursor podendo então ser facilmente transladado ou rotacionado. Serão descritas a seguir a técnica Simple Virtual Hand e Go-Go.

SIMPLE VIRTUAL HAND

Esta técnica é um mapeamento direto do movimento da mão do usuário no movimento de uma mão virtual.

A técnica Simple Virtual Hand é classificada como sendo uma técnica de interação isomórfica, pois simula diretamente a nossa interação com objetos do dia-a-dia. O único problema com esse tipo de técnica é que somente objetos que se encontram dentro da área de alcance do usuário é que podem ser selecionados e manipulados.

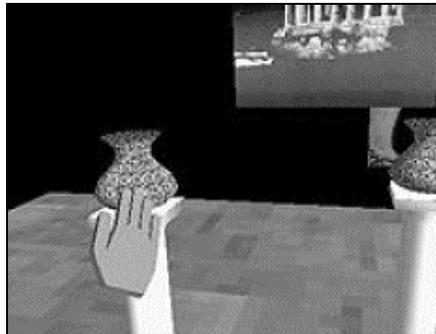


Figura 3.2 - Técnica Simple Virtual Hand [41]

GO-GO

Esta técnica é uma tentativa de melhorar a mão virtual provendo uma simplicidade que permite ao usuário mudar interativamente o comprimento do braço virtual.

A técnica Go-Go provê ao usuário trazer objetos para perto de si e movê-los para longe, mas a distância máxima alcançável ainda é finita.

3.1.1.3 Mundo em Miniatura

Esta técnica é denominada WIM (World-In-Miniature), e provê ao usuário um modelo portátil em miniatura, o qual é uma cópia exata do ambiente virtual, em menor escala. O usuário pode indiretamente manipular objetos virtuais interagindo com as suas representações no WIM.

A técnica WIM combina navegação e manipulação, visto que o usuário pode também mover a sua representação gráfica. O único problema com esta técnica é que às vezes ela pode não escalar muito bem. Apesar de funcionar relativamente bem em pequenos e médios ambientes, como por exemplo, o interior de um prédio virtual ou sala, usá-la em ambientes muito grandes pode requerer um fator escalar grande, resultando em cópias de objetos muito pequenas tornando a manipulação extremamente difícil.

Esta técnica vem se mostrando ser muito bem sucedida quando aplicada em interfaces 3D para AR.

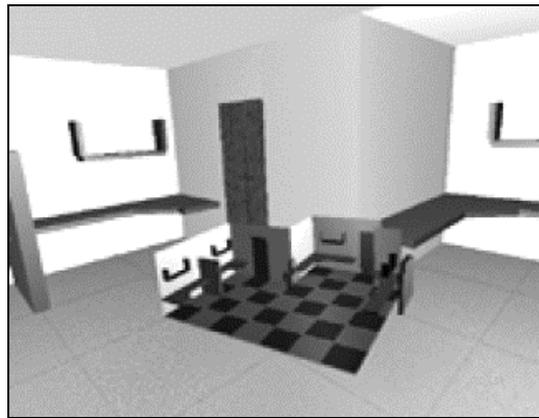


Figura 3.3 - Técnica Mundo em Miniatura [41]

3.1.1.4 Técnicas de Agregação e Integração

As técnicas de agregação simplesmente combinam várias técnicas. Ao usuário é fornecido um meio explícito de escolher a técnica de manipulação desejada dentro de um conjunto limitado de possíveis opções.

Ao se analisar todas as técnicas se notará que a maioria das manipulações é baseada em uma repetição: um objeto deve ser selecionado antes que possa ser manipulado. Neste caso, a interface pode simplesmente trocar de uma técnica de seleção para uma técnica de manipulação após o usuário selecionar um objeto, e depois voltar ao modo anterior. Teoricamente este tipo de técnica pode otimizar o desempenho em cada modo, sendo chamada de técnica de integração. As principais técnicas são HOMER, Scaled-World Grab e Voodoo Dolls, descritas a seguir.

HOMER

Nesta técnica o usuário seleciona um objeto usando a técnica Ray-Casting e, ao invés do objeto ser preso ao raio, ele será preso à mão virtual do usuário. Após isso, é feita uma troca para o modo de manipulação, permitindo ao usuário posicionar e rotacionar o objeto virtual.

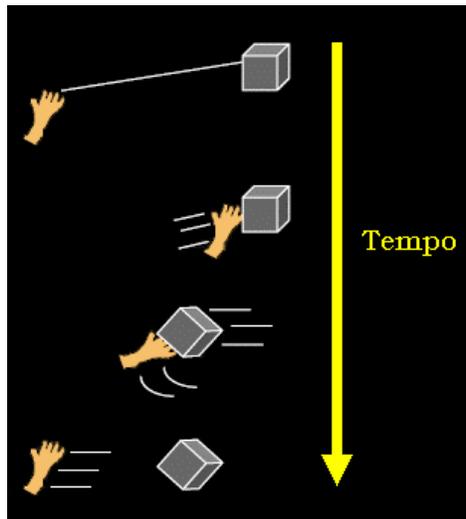


Figura 3.4 - Técnica HOMER [41]

SCALED-WORLD GRAB

Esta técnica é baseada em princípios similares à técnica descrita acima. Primeiramente o usuário seleciona um objeto usando alguma técnica de seleção, como a Image-Plane. Após a seleção, a interface troca para o modo de manipulação, e assim o usuário pode posicionar e rotacionar o objeto virtual no espaço. Contudo, ao invés de escalar a mão do usuário, como na técnica HOMER, Scaled-World Grab reduz todo o ambiente virtual ao redor do ponto de vista do usuário.

Assim como no HOMER, esta técnica não é muito eficiente em pegar um objeto localizado dentro do alcance do usuário e movê-lo para longe.

VOODOO DOLLS

Esta técnica é uma técnica de interação que faz uso de ambas as mãos do usuário, combinando as técnicas Image-Plane e WIM. O Voodoo Dolls usa um par de luvas para permitir ao usuário uma troca entre diferentes *frames* de referência para a manipulação, permitindo que essa seja feita com objetos de vários tamanhos e em diferentes distâncias.

Esta técnica propõe a manipulação de objetos virtuais indiretamente, usando miniaturas temporárias, cópias portáteis dos objetos denominadas *dolls*, similar ao WIM. Mas ao contrário desse, a técnica Voodoo Dolls permite que o usuário decida quais os objetos do ambiente virtual serão usados na interação.

Na primeira etapa, o usuário selecionará o objeto alvo (ou um grupo de objetos) com a técnica do Image-Plane, que criará uma *doll* representando este objeto, e a coloca na mão do usuário, escalando essas cópias em tamanhos convenientes. Na segunda etapa, o usuário irá explicitamente e interativamente especificar um *frame* de referência para a manipulação. A *doll* que o usuário segura em sua mão não-dominante representa um *frame* estacionário de referência, ou seja, o objeto virtual correspondente não se move quando o usuário move a *doll*. Para a manipulação começar é necessário que o usuário passe a *doll* para a mão dominante.

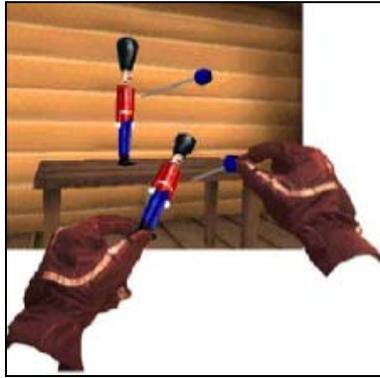


Figura 3.5 - Técnica Voodoo Dolls [41]

3.1.1.5 Manipulação Desktop 3D

Em sua grande maioria os sistemas interativos de computação gráfica ainda existem em computadores *desktop* convencionais e são usados em escritórios e em casa, como por exemplo, modelagem comercial 3D e pacotes de animação, sistemas CAD (*Computer Aided Design*), *software* de visualização de informação 3D, e jogos de computadores. As técnicas de interação usadas para controle e posicionamento de objetos 3D nesses ambientes diferem das técnicas mostradas anteriormente principalmente por causa dos dispositivos de entrada, que em computadores *desktop* são basicamente teclado e *mouse*, ou seja, dispositivos 2D. As principais técnicas são Controles de Interface 2D, Widgets 3D, Esfera Virtual e ARCBALL, descritas a seguir.

CONTROLES DE INTERFACE 2D

O método mais simples para especificar a posição e orientação 3D de objetos virtuais em ambientes *desktop* é, simplesmente, pedir ao usuário para colocar diretamente os valores das coordenadas e ângulos da posição e orientação desejada usando o teclado. Este método é o mais usado na maioria dos pacotes de modelagem 3D e CAD, pois estes precisam de especificações precisas e não-ambíguas.

Um dos grandes problemas é que as pessoas têm dificuldade em fazer cálculos de cabeça, mais especificamente de orientação 3D. Para isso um outro processo pode ser usado: o usuário repetidamente muda a posição de um objeto, avalia, reajusta, e repete este processo até alcançar a posição desejada.

WIDGET 3D

O funcionamento básico desta técnica consiste em colocar os controles diretamente na cena 3D junto aos objetos que estão sendo manipulados. Quando um objeto é selecionado, um número de objetos gráficos 3D (*widgets*) fica visível, e pode ser clicado e arrastado. Cada *widget* é responsável por somente um pequeno conjunto de manipulações.

A principal vantagem desta técnica é a transição fácil entre as diferentes seqüências de manipulação, já que o usuário simplesmente seleciona outro *widget* e faz outra manipulação. Já a sua principal desvantagem é a grande confusão visual, visto que o número de elementos gráficos na cena cresce, e o usuário precisará aprender como cada *widget* responderá ao clique do *mouse*.

ESFERA VIRTUAL

A idéia básica desta técnica consiste em: imagine que um objeto 3D que desejamos rotacionar está dentro de uma esfera de vidro que pode ser rotacionada livremente em torno do seu centro. Então ao invés de

rotacionar o objeto diretamente, é só rotacionar a esfera “cutucando-a” com o *mouse*. Para melhorar a visualização uma metáfora desta esfera de vidro é desenhada em torno do objeto virtual como um círculo ou uma esfera.

ARCBALL

Esta é uma técnica de rotação 3D desenvolvida especificamente para o uso do *mouse*. Similar a técnica anterior, o objeto estará dentro de uma esfera virtual. O usuário poderá estar rotacionando o objeto apenas clicando e arrastando o *mouse* no círculo que representa a projeção da esfera na superfície da tela.

3.1.2 Técnicas de Navegação

As técnicas de navegação, como o próprio nome diz, são as técnicas que são usadas para a interação com o mundo virtual, sendo ele um ambiente imersivo ou um ambiente *desktop*. A seguir serão descritas as técnicas de locomoção física, de direcionamento, de planejamento de rotas, baseadas em alvo, de manipulação visual, *Travel-by-Scaling*, de orientação do *viewpoint*, de especificação da velocidade, e com controles integrados da câmera para ambientes *desktop* 3D.

3.1.2.1 Técnicas de Locomoção Física

Estas técnicas tendem a usar o esforço físico que o usuário faz afim de transportá-lo através do mundo virtual, e são mais usadas em ambientes imersivos. Resumidamente, elas tentam imitar, em menor ou maior grau, o método natural de locomoção do mundo real, e são mais comuns em videogames e alguns sistemas de entretenimento baseados em locomoção. As técnicas *Walking*, *Walking in Place*, *Dispositivo Simulando o Caminhar* e *Cycles* serão descritas a seguir.

WALKING

A técnica mais direta e óbvia para navegar em um mundo 3D é andar através dele, contudo, por causa das limitações tecnológicas e espaciais, andar de verdade nem sempre é praticável e/ou possível. Caminhar de verdade somente pode funcionar quando o tamanho do ambiente é menor que o alcance do sistema de *tracking* (*tracker* = *hardware* de propósito especial usado em VR para medir em tempo real a mudança de orientação e posição de um objeto 3D).

Um problema ao se tentar andar de verdade são cabos, cabos para dispositivos de entrada, para os *trackers* e para os *displays*. Este problema pode ser parcialmente resolvido usando dispositivos *wireless*.

Existem basicamente duas abordagens para o caminhar de fato. A primeira delas é a *outside-in*, onde sensores óticos ou ultra-sônicos são colocados em pontos fixos, ou seja, localizações conhecidas no ambiente e os sensores de localização são colocados no usuário, podendo ser LEDs (*Light-Emitting Diode*) ou marcadores (vide tópico 2.1). A outra abordagem é a *inside-out*, onde os sensores estão fixos no usuário e os LEDs ou marcadores estão no mundo. Ambas as abordagens têm potencial para ser *wireless*, resolvendo o problema dos cabos. Esta técnica é particularmente muito usada em aplicações móveis de AR, como o *battlefield* ilustrado na figura abaixo.



Figura 3.6 - Sistema *battlefield* desenvolvido em AR [7]

WALKING IN PLACE

Uma alternativa para o caminhar de fato é o andar no lugar, ou seja, usuários movem os seus pés para simular um andar sem que o seu corpo se movimente.

Esta técnica é mais bem aplicada em sistemas onde alto nível de realismo é necessário, e onde o ambiente não é muito grande. Já para aplicações nas quais o foco principal é a eficiência e o desempenho da tarefa, contudo, a técnica anterior é mais apropriada.



Figura 3.7 - Técnica Walking in Place [41]

DISPOSITIVO SIMULANDO O CAMINHAR

Quando há necessidade de um maior realismo a técnica Walking in Place pode ser pouco satisfatória, pois não captura o mesmo movimento que um andar real. Técnicas de *real walking*, como as citadas acima, são em sua maioria limitadas pelo espaço físico e pela tecnologia.

Uma terceira abordagem é usar um dispositivo especial para a locomoção que provê uma movimentação similar ao andar real sem que o corpo do usuário seja transladado, este dispositivo especial é uma esteira. Uma alternativa ao uso de esteiras é um dispositivo constituído de uma plataforma onde o corpo fica em cima de duas bases de movimentação que simulam a troca dos passos deslizando para frente e para trás.

O problema no uso de esteiras é que elas são extremamente caras, suscetíveis à falhas mecânicas e respondem muito devagar ao movimento do usuário, e o mais importante, não produzem a percepção natural do caminhar para o usuário. Particularmente, depois o usuário aprende a andar respeitando as características do dispositivo.

CYCLES

Caso o esforço físico seja desejado, mas o caminhar não seja necessário, pode-se usar uma bicicleta ou algum outro dispositivo de pedal. Um exercício típico de bicicleta é muito mais fácil de implementar, pois geralmente estas já dispõem de dispositivos próprios que reportam a velocidade.

Esta técnica é menos efetiva que produzir uma simulação de andar real, e os usuários muitas vezes se queixam da dificuldade de virar o dispositivo ao simular curva, por exemplo.

3.1.2.2 Técnicas de Direcionamento

O direcionamento se refere ao controle contínuo da direção do movimento pelo usuário, ou seja, o usuário constantemente especifica a direção da navegação absoluta (mova ao longo do vetor $[1,0,0]$ em coordenadas mundiais) ou da navegação relativa (mova para a minha esquerda). Estas técnicas são geralmente fáceis de entender e provêem um maior nível de controle para o usuário. São elas Gaze-Directed, Pointing, Torso-Directed, Camera-in-Hand, Physical Props, Virtual Motion Controller e Semiautomated.

GAZE-DIRECTED

É a técnica mais comum, a mais simples e permite ao usuário se mover na direção que está olhando. Em um ambiente imersivo, a direção *gaze-directed* é obtida da orientação do *tracker* na cabeça do usuário (apesar de que na *gaze-directed* original usa-se o *tracker* do olho), já em ambientes *desktop*, a direção *gaze-directed* é dada ao longo de um raio na posição virtual da câmera através do centro da janela de visão.

Esta técnica é muito intuitiva quando usada em planos horizontais 2D, contudo, se for necessária a movimentação completa em 3D ela apresentará problemas. Na verdade seu maior problema é que os usuários não podem olhar em uma direção enquanto navegam em outra, pois seria como querer que não se olhe para outras direções enquanto se anda de bicicleta, caminha ou dirige.

POINTING

Para que não houvesse o problema descrito acima, foi desenvolvida esta técnica que combina a direção *gaze-directed* e a direção de navegação.

Esta consiste em basicamente separar os vetores, sendo um especificamente para a navegação que pode vir da orientação do *tracker* preso à mão do usuário. Em caso de ambientes *desktop*, o *mouse* pode ser usado para fornecer a direção *gaze-directed* e o teclado para fornecer a direção de navegação.

Esta técnica é mais flexível que a anterior, mas bem mais complexa já que o usuário tem o controle de dois valores simultaneamente. Em compensação é mais interessante, pois permite que o usuário olhe em todas as direções enquanto se movimenta.

TORSO-DIRECTED

Esta técnica utiliza o torso do usuário para especificar a direção de navegação. Ela é motivada pelo fato de que as pessoas naturalmente viram seus corpos para frente da direção que desejam caminhar. Para isso um *tracker* é colocado no torso do usuário, preferencialmente próximo à cintura, preso a um cinto, por exemplo.

A maior vantagem desta técnica é a separação das direções de onde se olha e para onde se vai. Além disso, permite ao usuário ficar com as mãos livres para fazer outras tarefas. Já a sua maior desvantagem é que só pode ser usada em ambientes onde se anda horizontalmente, pois é muito difícil apontar o torso para cima e para baixo.

CAMERA-IN-HAND

Esta técnica é mais usada quando se deseja navegar em um ambiente virtual *desktop*. Para isto um *tracker* é colocado na mão do usuário e a posição e orientação absoluta deste irá definir a posição e orientação da câmera da cena 3D. O *tracker* funciona como uma câmera virtual olhando para este mundo.

O problema com esta técnica é que ela pode se tornar confusa, pois o usuário tem uma visão em terceira pessoa do ambiente, mas o ambiente é desenhado de um ponto de vista em primeira pessoa.

PHYSICAL PROPS

As técnicas de direcionamento para a navegação também podem ser implementadas com uma variedade de *props* (dispositivos especializados desenvolvidos para a tarefa de direcionamento) físicos. O mais óbvio destes é o volante similar ao encontrado em um carro, o qual pode ser combinado com pedais de acelerador e freio.

Esta técnica é usada tanto em ambientes imersivos quanto em ambientes *desktop*, e é facilmente entendido por qualquer usuário. Em geral, ela é mais útil quando um certo tipo de veículo está sendo simulado, quando a interface deve ser usável por qualquer um sem nenhum tipo de treinamento, ou quando o direcionamento é uma parte importante da experiência como um todo.

VIRTUAL MOTION CONTROLLER

O VMC (Virtual Motion Controller) consiste em uma plataforma redonda constituída de sensores de pressão embaixo da superfície, que estão distribuídos ao longo da borda. Então, a partir do momento que o usuário se encontra em pé no centro, nenhuma pressão é sentida pelos sensores, mas à medida que o usuário se afasta do centro a pressão vai aumentando. Analisando estas pressões, o dispositivo pode determinar a direção que o usuário está do centro.

Sua maior desvantagem se encontra no fato de que este dispositivo está limitado a movimentações 2D.

SEMIAUTOMATED

O conceito básico por trás desta técnica é a disposição de regras para a movimentação do usuário, e a ele é permitido apenas o controle do direcionamento dentro dessas regras. Esta técnica é usada tanto em ambientes imersivos quanto em ambientes *desktop*.

3.1.2.3 Técnica de Planejamento de Rotas

Este conjunto de técnicas permite ao usuário especificar um caminho ou rota através do ambiente, e então se movimentar ao longo do caminho escolhido. Estas técnicas envolvem basicamente dois passos: o usuário planeja, e então o sistema executa o planejamento. Estas técnicas são as conhecidas como Desenhando um Caminho, Marcando Pontos ao Longo do Caminho e Manipulando a Representação do Usuário.

DESENHANDO UM CAMINHO

Uma maneira de especificar uma rota é desenhar o caminho desejado diretamente na cena 3D. Esta técnica inclui um tipo de mapeamento inteligente do caminho: ao invés de simplesmente projetar a linha 2D na superfície da cena 3D, o algoritmo leva em consideração a continuidade da rota e a superfície a ser seguida até o ponto final. Caso a cena seja muito extensa, esse desenho pode ser feito em um mapa 2D ou 3D que representa o ambiente como um todo.

MARCANDO PONTOS AO LONGO DO CAMINHO

Outra maneira de especificar um caminho em um ambiente 3D é colocar pequenos pontos em localizações-chaves ao longo da rota. Esses pontos podem ser colocados diretamente no ambiente (com a ajuda do *mouse*) ou em mapas 2D e 3D, e o sistema ficará responsável por criar um caminho contínuo que passa por todos esses pontos.

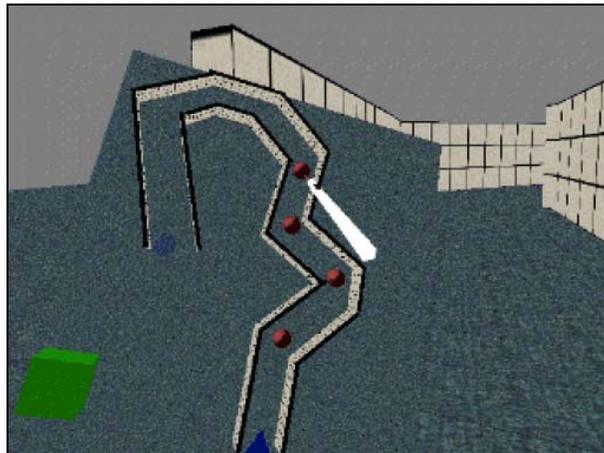


Figura 3.8 - Técnica Marcando Pontos ao Longo do Caminho [41]

MANIPULANDO A REPRESENTAÇÃO DO USUÁRIO

Esta técnica envolve o usuário manipular uma representação virtual dele mesmo a fim de planejar uma rota. Consiste em basicamente o usuário selecionar e manipular a si mesmo no ambiente em miniatura a fim de definir o caminho para um ponto mais adiante então o sistema executa esta movimentação no ambiente real.

3.1.2.4 Técnicas Baseadas em Alvo

Algumas vezes, o usuário apenas especifica o ponto final para onde deseja ir; estas técnicas se baseiam neste tipo de requisito. São elas Especificação Baseada em Mapa ou em WIM e Zoomback.

ESPECIFICAÇÃO BASEADA EM MAPA OU EM WIM

Um mapa 2D ou um WIM 3D pode ser usado para especificar uma localização alvo dentro do ambiente. A característica principal é que o usuário moverá sua própria representação.

A diferença entre esta técnica e a técnica definida no tópico anterior é que nessa o sistema deve gerar um caminho da posição atual até o alvo ao invés de simplesmente usar o caminho definido pela movimentação da representação do usuário no mapa ou no WIM.

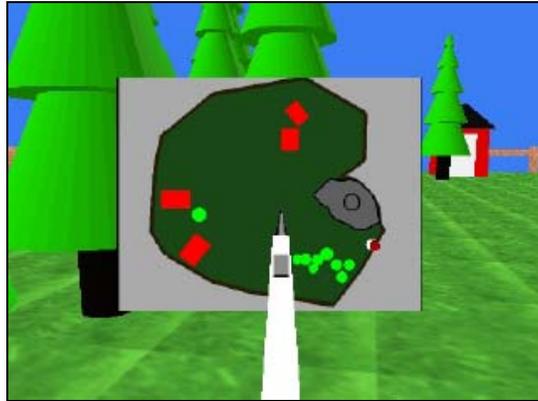


Figura 3.9 - Técnica Especificação Baseada em Mapa ou em WIM [41]

ZOOMBACK

Esta técnica usa uma metáfora da técnica Ray-Casting para selecionar um objeto no ambiente, e então mover o usuário para a posição diretamente na frente deste objeto. Esta técnica consegue reter a informação sobre a posição prévia do usuário e permite ao mesmo retornar à posição inicial depois de inspecionar o objeto alvo.

Um exemplo poderia ser em uma galeria de arte, o usuário seleciona um quadro específico e pode examiná-lo de perto, e então depois retorna à posição inicial de onde ele pode ver vários quadros.

3.1.2.5 Técnica de Manipulação Manual

Estas técnicas usam metáforas de manipulação de objetos *hand-based* para manipular o *viewpoint* ao invés de manipular o objeto virtual. Estas podem ser usadas em situações onde ambas as tarefas, de navegação e de manipulação de objetos, são frequentes e intercaladas. As técnicas são Agarrando o Ar e Objetos Fixos.

AGARRANDO O AR

Neste conceito, o mundo inteiro é visto como um objeto a ser manipulado. Quando o usuário faz um gesto similar ao de agarrar em qualquer ponto do ambiente e então move a sua mão, o mundo inteiro move enquanto o *viewpoint* fica estacionário. Esta técnica permite cobrir distâncias muito maiores usando uma técnica de extensão de braço, por exemplo, a Go-Go.

OBJETOS FIXOS

Nesta técnica o usuário seleciona um objeto no ambiente e faz movimentos com a mão, como se fosse

manipular este objeto. Contudo, o objeto fica parado e o *viewpoint* é movido em relação ao objeto. Como por exemplo, se imagine agarrando um mastro de bandeira, e este está fixo firmemente no solo então, quando você move sua mão de encontro ao corpo, o mastro não se move, ao invés disso você se move para perto dele.

3.1.2.6 Técnica Travel-by-Scaling

No tópico 3.1.2.1, notou-se que o método mais natural e intuitivo para a navegação em um mundo virtual 3D é o caminhar real, mas o andar físico é limitado pelo alcance e pelo espaço. Um meio de aliviar este problema é permitir que o usuário possa mudar a escala do mundo, com isso um passo de um metro pode representar um nanômetro, um quilômetro, ou qualquer outra distância.

São muitos os desafios quando se desenvolve uma técnica de navegação por escalonamento, pois o usuário precisa entender a escala do mundo para que ele possa determinar o quão longe ele pode mover e precisa entender o *feedback* visual que ele obtém quando move. Outro problema é que o ato de escalar e depois re-escalar continuamente pode gerar desconforto e *cybersickness*.

3.1.2.7 Técnicas de Orientação do Viewpoint

Até agora o foco foi quase que exclusivo em técnicas para a mudança da posição (coordenadas xyz) do *viewpoint*, mas a navegação também inclui a tarefa de ajustar a orientação do *viewpoint* (puxar, levar aos lados, girar). As técnicas Head Tracking, Orbital Viewing e Esfera Virtual tratam desta questão.

HEAD TRACKING

Para ambientes virtuais imersivos não há necessidade de definir explicitamente uma orientação para o *viewpoint*, pois este é dado por *default* a partir do *tracker* posicionado na cabeça do usuário.

ORBITAL VIEWING

Esta técnica é usada para ver um único objeto virtual de todos os lados, a fim de ver a parte de baixo do objeto o usuário olha para cima, a fim de ver o lado esquerdo o usuário olha a direita, e assim por diante.

ESFERA VIRTUAL

Esta técnica é mais usada em ambientes virtuais *desktop*. As técnicas mais comuns são Esfera Virtual (vide tópico 3.1.1.5.3) e ARCBALL (vide tópico 3.1.1.5.4), ou seja, pontos de vista em terceira pessoa. Para pontos de vista em primeira pessoa, ou seja, o *viewpoint* ser considerado o centro da esfera imaginária, basta clicar o *mouse* que a esfera é rotacionada ao redor do *viewpoint*.

3.1.2.8 Técnica de Especificação da Velocidade

A maioria das técnicas ignora a mudança da velocidade e simplesmente ajusta o que parece ser uma velocidade constante razoável. Contudo, isto pode gerar uma grande variedade de problemas, visto que uma velocidade constante sempre parecerá muito devagar em algumas situações e muito rápida em outras. A maior desvantagem desta técnica ao permitir que o usuário tenha controle da velocidade é que isto acarretará em uma maior complexidade na interface.

3.1.2.9 Controles Integrados da Câmera para Ambientes Desktop 3D

Em alguns casos, uma aplicação *desktop* requer um conjunto integrado de técnicas de interação desenvolvidas especificamente para uso neste tipo de ambiente. Por exemplo, alguns navegadores VRML (*Virtual Reality Modelling Language*) possuem vários modos de navegação que permitem ao usuário controlar um ou dois dos 6-DOF (*Degrees of Freedom*, dimensões independentes de movimento de um corpo) que a câmera possui, um de cada vez.

3.1.3 Técnicas de Controle de Sistemas

Basicamente os elementos para o controle de sistemas em 2D são os seguintes: menus *pull-down* e *pop-up*, *toolboxes*, *palletes*, *radio buttons*, etc., mas o problema reside no fato de que essas interfaces não podem simplesmente ser mapeadas em um contexto 3D. Para resolver este problema foram criadas as técnicas de controle de sistemas, tais como menus gráficos, comando de voz, comando de gestos e ferramentas.

3.1.3.1 Técnica dos Menus Gráficos

Esta técnica faz uma correspondência entre os menus gráficos 2D para um contexto 3D, e é mais usada por causa da familiaridade dos menus. As técnicas Menu 2D Adaptados, Menu 1-DOF, Menu TULIP e 3D Widgets.

MENUS 2D ADAPTADOS

São simplesmente adaptações dos seus similares em 2D, e funcionam basicamente do mesmo modo que no *desktop*. Exemplos de menus 2D que foram adaptados para um contexto 3D podem ser: menus *pull-down*, menus *pop-up*, menus *floating* e *toolbars* (barras de ferramenta).

Uma adaptação que tem sido muito bem sucedida seria anexar os menus à cabeça do usuário. Desta maneira o menu estaria sempre disponível não importando para onde o usuário estivesse olhando. Outra adaptação bastante interessante é anexar o menu a uma superfície física, um *tablet*, por exemplo, já que encontrar esse menu é tão fácil quanto trazer o *tablet* para a vista.

Uma vantagem em usar esta técnica é que os usuários já estão familiarizados com o estilo de navegação dos menus. Já a desvantagem seria que muitas vezes esses menus podem ocultar o ambiente, e, além disso, os usuários podem achar difícil encontrar e selecionar os itens do menu.

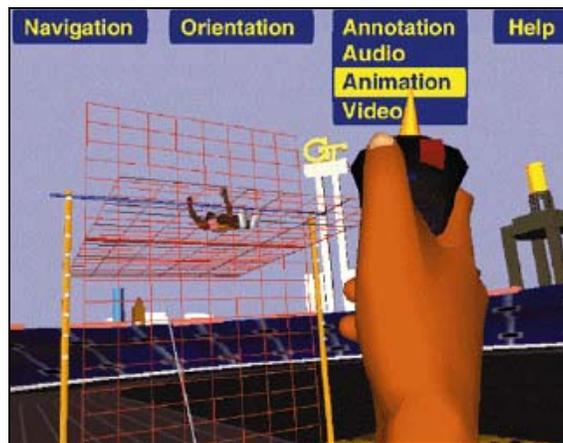


Figura 3.10 - Técnica Menu 2D Adaptado [41]

MENUS 1-DOF

Ao selecionar o item de um menu está acontecendo essencialmente uma operação unidimensional. Esta observação levou ao desenvolvimento de menus 1-DOF.

Este tipo de menu é na maioria das vezes anexado à mão do usuário, com os itens do menu arranjados em um padrão circular em volta dela, bastando que o usuário rotacione a mão para o item desejado cair em uma “cesta de seleção”. Não há nenhum tipo de obrigatoriedade no padrão ser circular, ele podendo ser muito bem linear.

Este menu é muito fácil de usar, pois a seleção é feita rapidamente se houver uma quantidade de itens relativamente pequena, e o acesso é rápido já que o menu se encontra na mão do usuário.

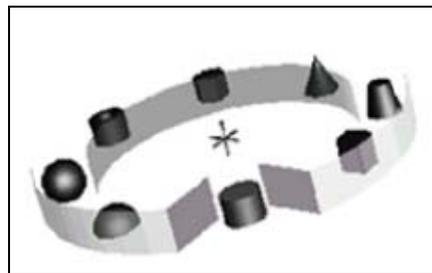


Figura 3.11 - Técnica Menu 1-DOF [41]

MENUS TULIP

O menu TULIP (*Three-Up, Labels In Palm*) funciona do seguinte modo: cada item é anexado a um dedo, usando luvas *pinch* (dispositivo de entrada que determina se um usuário está tocando dois ou mais dedos juntos). Ao tocar um dedo e o polegar da mesma mão isso é interpretado como uma seleção do menu.

Esta técnica é mais usada quando se o menu possui mais de quatro itens. Assim sendo, os três primeiros são mostrados e no dedo mínimo ficará a nomeação “*more*”, e quando este é selecionado os próximos três itens são mostrados nos dedos e os três itens anteriores inativos são deslocados para a palma da mão. Esta técnica provê uma seleção direta, e os usuários podem ver todos os itens e como acessá-los.

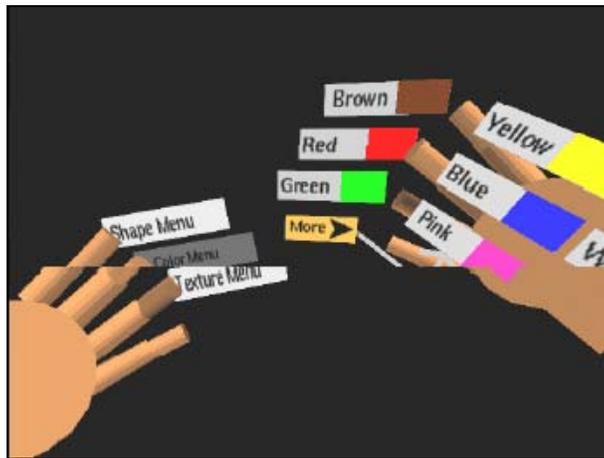


Figura 3.12 - Técnica Menu TULIP [41]

3D WIDGETS

São considerados os menus mais exóticos e são definidos basicamente em dois tipos: sensíveis a contexto e não sensíveis a contexto.

No primeiro tipo a funcionalidade do menu é movida para um objeto dentro do ambiente 3D e a geometria e as funcionalidades são unidas. Estes *widgets* só aparecerão caso o objeto seja selecionado, e este tipo é mais usado para mudar o modo de manipulação (translação, rotação e escalonamento) do objeto.

No segundo tipo o propósito é mais geral, ou seja, o *widget* fornece qualquer tipo de menu não precisando necessariamente ter algo a ver com o objeto.

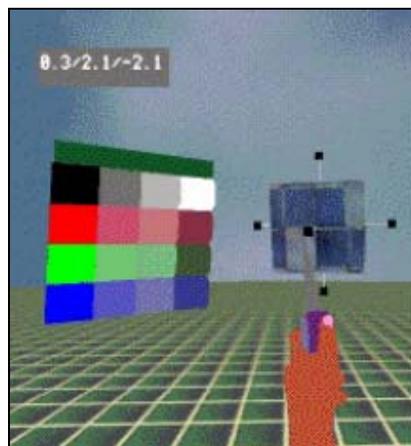


Figura 3.13 - Técnica Menu 3D Widgets [41]

3.1.3.2 Técnicas de Comando de Voz

Um sistema de fala de diálogos provê uma interface entre um usuário e uma aplicação, permitindo uma interação falada de uma maneira natural.

Vários fatores podem influir na taxa de reconhecimento da fala da aplicação, desde a variação entre as vozes dos usuários até o barulho ao fundo. O motor de reconhecimento da voz pode ser de tipos: dependente ou independente do orador. No primeiro o sistema necessitará de um treinamento especial e no

segundo tipo nenhum treinamento é requerido. A resposta gerada como saída deste tipo de técnica pode ser de confirmar a ação ou de informar que precisa de mais entrada para completar um comando de controle.

3.1.3.3 Técnica de Comando de Gestos

Os comandos baseados em gestos podem ser classificados em dois tipos: posturas e gestos. O primeiro é uma configuração estática da mão e o segundo é um movimento dinâmico. A usabilidade de uma aplicação que usa comandos por gestos depende basicamente do número e da complexidade dos gestos; quanto mais gestos mais há a necessidade do usuário aprender.

Existem vários tipos de gestos, como gestos de mão conectados à fala, que nada mais é do que a gesticulação espontânea feita não intencionalmente enquanto falamos, e é mais aplicada em interfaces multimodais. Existem os gestos mímicos, que são gestos que não estão ligados à fala, mas são diretamente usados para descrever um conceito. Tem também os gestos simbólicos, que são gestos usados no dia-a-dia para expressar algo, um sentimento, por exemplo. E, por último, existe ainda a linguagem dos sinais, que são um conjunto específico de posturas e gestos usados para a comunicação.

3.1.3.4 Técnicas Ferramentas

São também chamadas de interfaces tangíveis. Estas consistem em usar elementos físicos que representam um tipo específico de ação, a fim de interagir com o sistema. Por exemplo, uma borracha de verdade pode servir para “deletar” objetos virtuais ou um lápis real pode servir para desenhar em um espaço virtual.

Esta técnica foi desenvolvida tanto para o uso em ambientes virtuais imersivos quanto em ambientes *desktop* e, além disso, alguns dispositivos físicos também facilitam o acesso a operações e informações chave. A abordagem da interface tangível combina representação e controle, combinando representação física e digital. As principais características deste tipo de interface são: representação física é computacionalmente unida com a informação digital base; representação física incorpora mecanismos para o controle da interação e representação física é perceptivelmente unida para ativamente mediar a representação digital.



Figura 3.14 - Técnica Ferramentas [41]

3.1.4 Técnicas de Saída Simbólica

As técnicas de saída simbólica são as baseadas em teclado, em canetas, em gestos e na fala. As mesmas são descritas na seqüência.

3.1.4.1 Técnicas Baseadas em Teclado

Estas técnicas fazem uso de um teclado físico ou uma metáfora de um teclado físico. São elas Teclado Miniatura, Teclado Low Key-Count, Teclado Chord, Teclado Pinch e Teclados Soft.

TECLADO MINIATURA

O modo mais fácil de trazer um teclado para interfaces 3D é simplesmente construir uma miniatura do mesmo para que possa ser carregado. Nesta técnica o *layout* do teclado se mantém o mesmo dos teclados normais (*layout* QWERTY). Este tipo de teclado é muito popular em dispositivos móveis, como os PDAs, mas não são grandes o suficiente para permitir teclar com os dez dedos; ao invés disso, são usados apenas um ou dois dedos.

TECLADO LOW KEY-COUNT

Existe uma segunda maneira de desenvolver teclados pequenos o suficiente para caber em uma mão. O meio é reduzir o número de teclas físicas, por exemplo, o celular. Neste as letras são mapeadas nas teclas dos números do telefone, causando às vezes ineficiência, pois quando se precisa de rapidez muitas vezes é necessário esperar caso as letras desejadas estejam mapeadas na mesma tecla.

Um outro modo desse sistema é o T9, onde ao invés de se digitar letra por letra, o sistema com o auxílio de um dicionário tenta “adivinhar” o que o usuário deseja escrever. Esta tarefa pode se tornar extremamente difícil caso se queira escrever nomes, palavras raras e não-palavras.

TECLADO CHORD

Este tipo de teclado é um dispositivo físico de entrada que provê todas as funcionalidades de um teclado normal, mas contém menos teclas.

A fim de prover mais símbolos do que existem teclas no dispositivo, o usuário necessita apertar várias teclas simultaneamente para produzir alguns símbolos. Geralmente, para executar tal tarefa o usuário precisará ser treinado, pois este não utiliza o padrão comum QWERTY.

Um estudo comparativo feito pelos autores do livro [9] entre esta e outras técnicas de entrada simbólica (teclado *chord*, fala, teclado *pinch* e teclado *pen-and-tablet*) em ambientes virtuais imersivos, e Teclado Chord foi a que teve o desempenho mais lento, apresentando o maior número de erros e a menor preferência do usuário.

TECLADO PINCH

Nesta técnica são usadas as luvas *pinch*, e o padrão usado é o QWERTY. Por isso, o usuário tem a vantagem do conhecimento prévio de como utilizá-lo. O conceito básico é que um simples toque entre o polegar e outro dedo da mesma mão representará uma tecla pressionada por este dedo.

Basicamente, na linha inicial do teclado, o dedo mínimo representa a letra *a*, o dedo anular representa a letra *s*, e assim por diante; as letras do meio, nesse caso as letras *g* e *h*, são selecionadas rotacionando a mão. Para mudar a linha ativa basta apenas mover a mão para perto do corpo, descendo uma linha, ou para longe do corpo, subindo uma linha. Foram criados também alguns gestos especiais para as teclas de espaço, *backspace*, *delete all* e *enter*.

Existem três *feedbacks* para este tipo de aplicação: no primeiro é mostrada na tela a localização de cada caracter e quais são os que estão ativos, no segundo é mostrado na tela o texto que está sendo digitado

pelo usuário e no terceiro o áudio simulando quando a tecla for pressionada.

Um estudo comparativo feito pelos autores do livro [9] indica que este teclado é mais lento que a técnica da fala (vide tópico 3.1.4.4) ou Teclado Soft, mas os usuários o acham mais fácil de aprender e confortável para o uso.

TECLADOS SOFT

Este tipo de teclado é um dispositivo virtual implementado inteiramente em *software*, ou seja, o usuário pressiona teclas virtuais ao invés de teclas físicas.

Alguns destes têm a vantagem de que podem ser facilmente reconfigurados para suportar diferentes *layouts* ou alfabetos, e não necessitam de nenhum dispositivo de entrada especializado.

Sua maior desvantagem é a falta do *feedback* háptico, já que o usuário não sente os contornos da tecla. Dispositivos hápticos servem para ajudar a achar uma tecla sem olhar, ou a sensação da tecla “afundar” para saber se a mesma já foi pressionada.

3.1.4.2 Técnicas Baseadas em Canetas

Algumas plataformas móveis, como, por exemplo, PDAs, também usam este tipo de entrada. São elas Reconhecimento de Gestos Pen-Stroke e Entrada Unrecognized Pen Input.

RECONHECIMENTO DE GESTOS PEN-STROKE

Em entradas que são baseadas em canetas, a unidade de reconhecimento básico é chamada *stroke*. O movimento da caneta começa quando ela toca a superfície de entrada e termina quando a caneta é levantada.

Vários esquemas têm sido desenvolvidos em cima dessa técnica; alguns usam um *stroke* para cada caracter, outros permitem a entrada de múltiplos caracteres com um único *stroke*.

ENTRADA UNRECOGNIZED PEN INPUT (DIGITAL INK)

Consiste em simplesmente desenhar o atual *stroke* da caneta, ou seja, é como se a caneta estivesse escrevendo com tinta digital. Esta técnica é muito popular em PDAs e permite que os usuários facilmente digitem texto e símbolos.

Sua principal desvantagem é que estes textos só podem ser lidos por outra pessoa, e não por computadores, por isso é interessante que sejam feitas somente anotações. Como, por exemplo, um médico fazer algumas pequenas anotações nos raios-X do paciente.



Figura 3.15 - Técnica Digital Ink [41]

3.1.4.3 Técnicas Baseadas em Gestos

É natural considerar que este tipo de técnica tire vantagem da postura, posição, orientação e movimento da mão, ou seja, que ela se baseie em gestos. São elas Gestos da Linguagem dos Sinais, Gestos Numéricos e Gestos Instantâneos.

GESTOS DA LINGUAGEM DOS SINAIS

A linguagem dos sinais é incrivelmente descritiva e permite aos usuários “falar” muito rapidamente. No caso de aplicações 3D uma luva é o dispositivo mais indicado para ser usado. Os maiores problemas ao se usar esta técnica é que apenas uma pequena porcentagem da população conhece a linguagem dos sinais e o sistema de reconhecimento também precisa ser previamente treinado.

GESTOS NUMÉRICOS

Esta técnica de interação oferece o uso dos dedos para representar números. A vantagem destes gestos é que eles são praticamente universais e podem ser feitos com uma mão ou com ambas.

GESTOS INSTANTÂNEOS

Ambas as técnicas descritas acima requerem reconhecimento de gestos contínuos usando luvas, já que é necessário o reportar contínuo dos ângulos entre os dedos. O uso de uma luva *pinch* poderia prover limites aos gestos.

Um exemplo desta técnica é a Teclado Pinch (tópico 3.1.4.1.4), já que muitos gestos tentam imitar o uso de um teclado normal. Existem outros tipos de gestos *pinch* que podem ser usados para representar letras, símbolos, números, impressão ou aumento.

3.1.4.4 Técnicas Baseadas na Fala

Estas técnicas têm um grande número de características desejáveis: permitem ao usuário ficar com as mãos livres, permitem uma entrada eficiente e permitem grandes quantidades de texto, sendo completamente naturais e familiares. Porém, a pergunta que é feita é: “porque então a fala é tão raramente usada para entrada simbólica?”. Um argumento usado há muito tempo é que os sistemas de reconhecimento são

devagar, não precisos e requerem treinamento. Ainda assim, por suas potenciais vantagens para interfaces 3D, algumas técnicas específicas para a fala serão mostradas, tais como Reconhecimento da Fala Single-Character, Reconhecimento da Fala Whole-Word e Entrada da Fala Unrecognized.

RECONHECIMENTO DA FALA SINGLE-CHARACTER

Nesta técnica o usuário deve expressar cada caracter ou símbolo explicitamente. No caso de uma frase ou uma sentença esta abordagem se torna extremamente entediante e inútil, mas em algumas situações como, por exemplo, no caso de serem ditas poucas palavras, esta técnica de soletrar faz sentido.

RECONHECIMENTO DA FALA WHOLE-WORD

A maioria dos *softwares* de reconhecimento da fala reconhece palavras ou frases ao invés de simplesmente caracteres. Esta técnica permite uma entrada mais rápida, mas conseqüentemente aumenta o número de erros. Este tipo de sistema pode ser mais apropriado quando grandes quantidades de texto precisam ser colocadas.

ENTRADA DA FALA UNRECOGNIZED

Pode-se considerar a voz digital o análogo da voz para a tinta digital. Esta técnica consiste em simplesmente salvar como um *stream* de áudio ao invés de símbolos ou palavras. Como a tinta digital, esta técnica se aplica somente a tarefas onde outra pessoa será o recipiente da mensagem.

3.2 Técnicas de Interação Específicas para Realidade Aumentada

A subdivisão feita nesta seção é uma das várias encontradas no meio científico. Visto que a área de AR é uma área considerada nova, ainda é extremamente difícil que todo o meio científico entre em um consenso sobre os tipos de técnicas, por isso outras divisões de técnicas podem ser encontradas na literatura. Não foi intenção propor nenhum tipo de classificação.

A divisão das técnicas apresentada aqui foi baseada no artigo de Wolfgang Broll et al. [10], e menciona técnicas para interação espacial, interação baseada em agentes, interação de controle virtual e de controle físico.

3.2.1 Interação Espacial

Este tipo de interação é baseado em algumas propriedades de manipulação espacial de objetos físicos. A interação espacial é tipicamente realizada por gestos dinâmicos, apontadores e interfaces tangíveis. Sua principal vantagem é que foi muito bem adaptada para selecionar objetos virtuais em um espaço 3D e para transformações espaciais.

Um exemplo bastante interessante de interfaces tangíveis é a metaDESK [44], ilustrada na Figura 3.16 que consiste em um sistema gráfico dirigido pela interação com objetos físicos. Este sistema é composto por uma mesa, que tem uma superfície em que pode ser projetado algo, e lentes e objetos físicos para serem colocados em cima desta. Basicamente este sistema funciona do seguinte modo: colocando um pequeno objeto nesta mesa um mapa 2D aparecerá “dentro” da mesa, o objeto físico terá uma espécie de invólucro de acrílico transparente e através das lentes poderá ser vista uma perspectiva 3D de alguma parte do mapa.



Figura 3.16 - Aplicação metaDesk, vista do sistema (cima esquerda), lentes mostrando uma vista 3D (cima direita), objeto com invólucro de acrílico (baixo esquerda) e objetos físicos reais (baixo direita) [44].

Um outro exemplo deste tipo de técnica, mas usando ponteiro ao invés de objetos físicos, como no caso acima, é o ARTHUR [11]. Esta aplicação é um sistema colaborativo que provê um suporte para as diferentes fases de um desenvolvimento arquitetural e planejamento urbano. Esta aplicação é uma interface colaborativa, pois permite a interação entre os vários *experts* em volta da mesa onde se encontra o sistema em funcionamento.

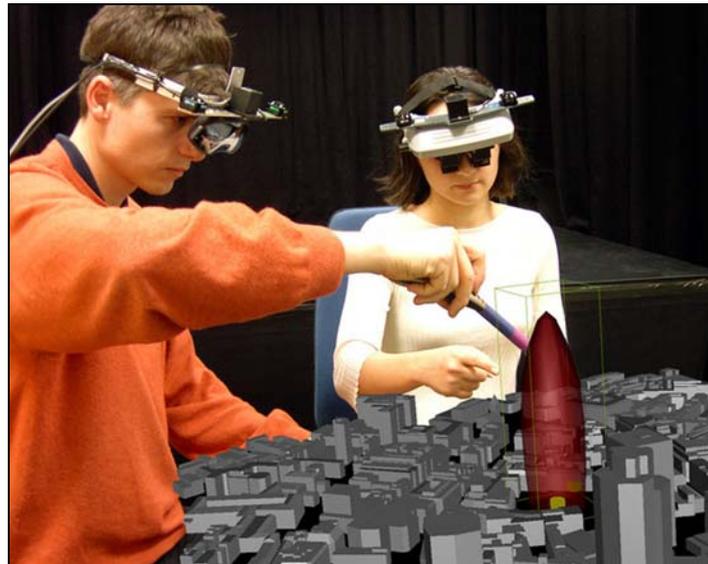


Figura 3.17 - Sistema ARTHUR em funcionamento [11]

3.2.2 Interação Baseada em Agentes

Esta técnica também é chamada de baseada em comandos, e consiste em ligações de entrada simples ou compostas que especificam funcionalidades. Este tipo de interação é tipicamente implementado para usar posturas (gestos estáticos) ou comandos de voz. Sua principal vantagem é ser especialmente adaptada para realizar entradas distintas.

Um exemplo que ilustra esta técnica, quando baseada em posturas, é uma navegação 3D em *displays* imersivos [40]. Esta aplicação consiste em através de três padrões de posturas da mão o usuário navegar entre dois modos de controle. Os padrões são a mão aberta (como se representa o número cinco), a mão apontando e a mão fazendo um OK; a primeira permite ativar o modo de navegação, a segunda permite que ao apontar para o prédio o usuário obtenha seu nome e o terceiro permite a troca entre os dois modos. A figura abaixo ilustra a aplicação.



Figura 3.18 - Protótipo de um sistema para a navegação 3D [40]

Outro exemplo desta mesma técnica, só que baseada em comandos de voz, é a aplicação Welbo [3], ilustrada na Figura 3.19. Este nada mais é do que um agente de conversação que além de morar em um ambiente 3D, pode dividir seu espaço interagindo com outros usuários e possui um “corpo” 3D, podendo ser percebido pelos usuários como um “ser”. O Welbo serve basicamente para guiar e ajudar usuários em um ambiente que simula uma sala, onde os mesmos podem visualizar e simular a localização dos móveis. A interação entre o Welbo e os usuários é basicamente feita através da fala, comportamento e conhecimento.

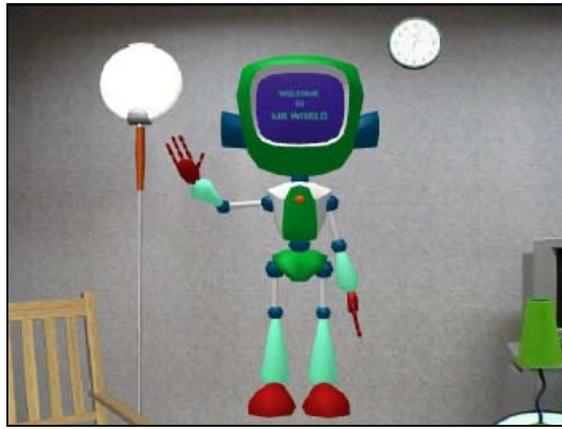


Figura 3.19 - Sistema Welbo [3]

3.2.3 Interação de Controle Virtual

Esta técnica é baseada em *widgets* 3D que representam uma certa funcionalidade. Sua principal vantagem é que ela implementa uma metáfora de interação bem familiar.

Um exemplo desta técnica também é encontrado no sistema ARTHUR [11], conforme Figura 3.20. Uma parte desta aplicação, mais especificamente um sistema CAD, fornece um menu 3D que provê entradas apropriadas representando os comandos ou elementos da interface do usuário.

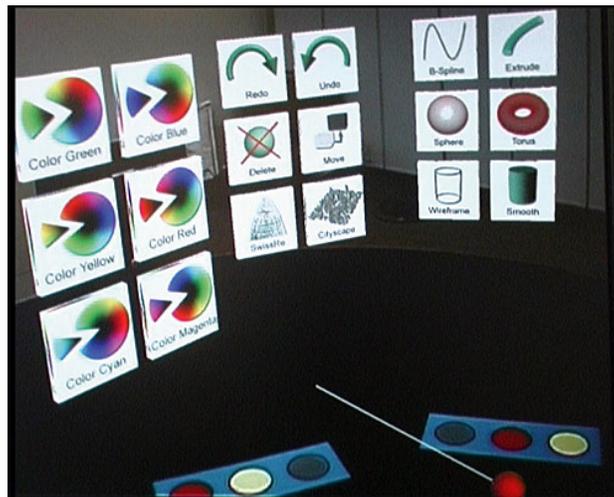


Figura 3.20 - Menus 3D representando comandos de um *software* CAD [11]

3.2.4 Interação de Controle Físico

Esta técnica é baseada em ferramentas físicas ou painéis de controle. Estes podem ser “estendidos” para controlar não somente objetos físicos como também objetos virtuais. Sua grande vantagem é que este tipo de interação permite a integração de ferramentas físicas na interface do usuário. A figura abaixo ilustra um *stage controller*.



Figura 3.21 - Exemplo de controle físico, um *stage controller* [10]

3.3 Dispositivos de Interação

Nesta seção serão mostrados os mais diversos dispositivos usados para interagir em AR. Dentre esses serão abordados os mais comuns como, por exemplo, HMDs e luvas, até os dispositivos ditos avançados como, por exemplo, celulares e PDAs.

3.3.1 *Sistemas de Exibição*

Os ditos sistemas de exibição abordados neste tópico incluem os HMDs ou *Head-Worn Displays* (HWDs) [7], os sistemas *monitor-based* e os *Virtual Retinal Displays* (VRD).

Os HMDs se dividem em dois tipos: os *closed-view* e os *see-through*. O primeiro não permite nenhuma visão direta do mundo real e são os mais usados em VR, enquanto que o segundo advém da necessidade do usuário de ver o mundo real que está a sua volta, ou seja, ele permite que o usuário enxergue o mundo real com objetos virtuais sobrepostos nele.

Os HMDs do tipo *see-through* podem usar dois tipos de tecnologia, ótica ou de vídeo, constituindo respectivamente dois tipos de HMDs, os *optical see-through* HMDs e os *video see-through* HMDs.

O *optical see-through* HMD, ilustrado na Figura 3.22, funciona com combinadores, espelhos semitransparentes, que combinam as imagens do mundo real com as imagens vindas do computador. Esses combinadores são parcialmente transmissivos permitindo que o usuário possa ver o mundo real através dele.

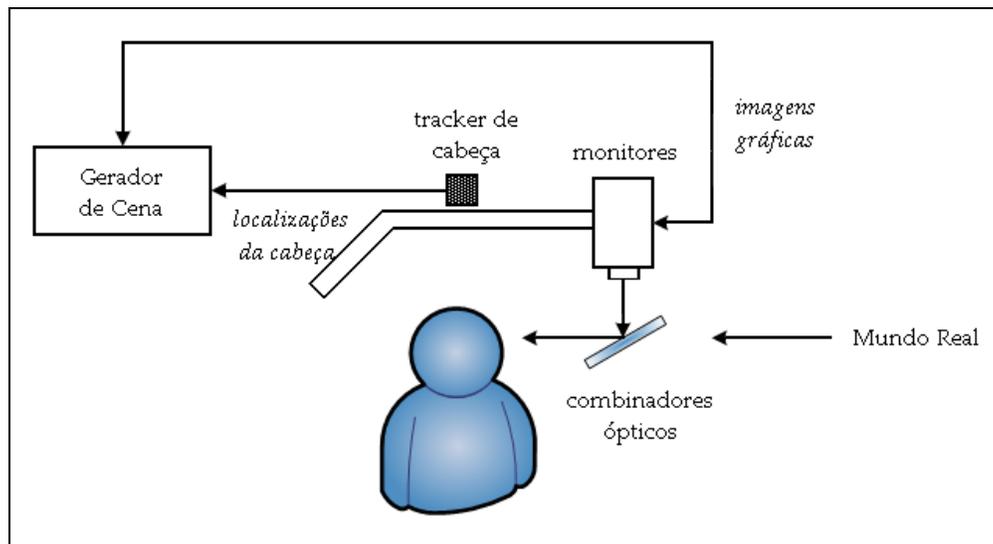


Figura 3.22 - Diagrama conceitual do *optical see-through* HMD, baseado em [6]

As principais vantagens deste HMD, ilustrado na Figura 3.23, são [6]:

- simplicidade: só existe uma *stream* de vídeo para se preocupar, as imagens gráficas;
- resolução: a resolução do mundo real não é limitada pela resolução dos *displays*;
- segurança: se a energia for cortada o usuário apenas terá a sensação de estar usando pesados óculos escuros;
- *no eye offset*: o usuário pode ver através do HMD evitando que haja o problema das câmeras não estarem na mesma posição dos olhos.

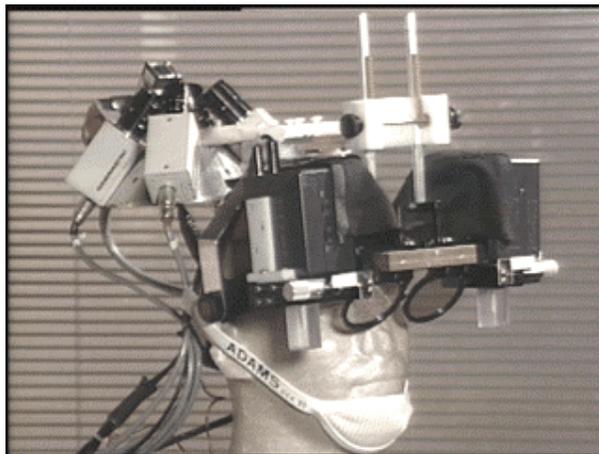


Figura 3.23 - *Optical see-through* HMD [14]

Já o *video see-through* HMD apesar do nome funciona como os HMDs do tipo *closed-view*, ou seja, uma ou duas câmeras são posicionadas na cabeça do usuário provendo as imagens do mundo real. O vídeo que advém dessas câmeras é então combinado com as imagens gráficas que foram criadas pelo gerador de cenas, combinando assim real e virtual. A Figura 3.24 ilustra o seu funcionamento.

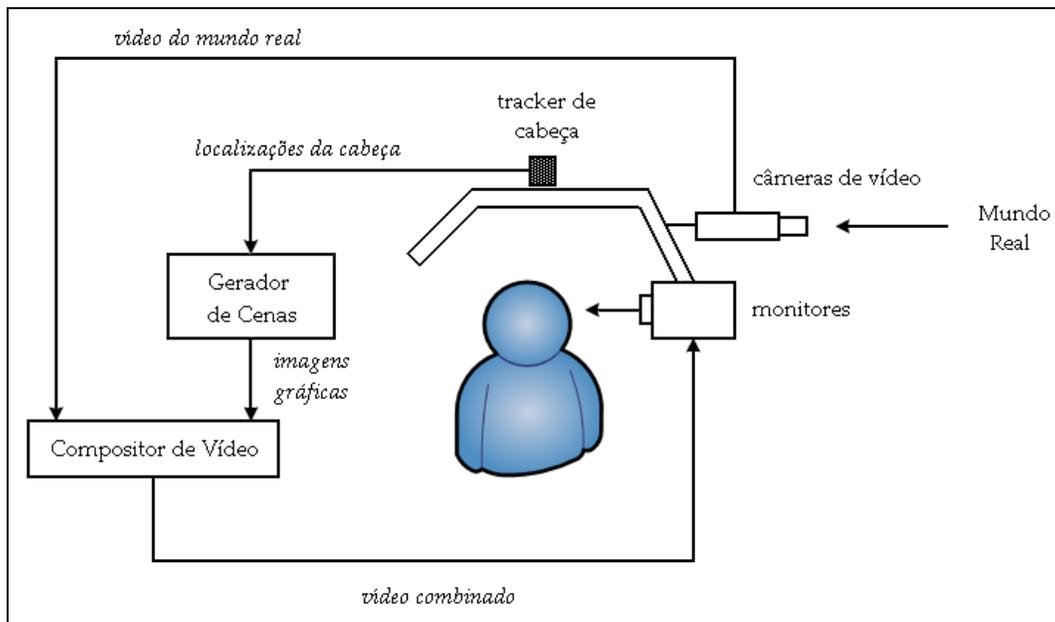


Figura 3.24 - Diagrama conceitual do *video see-through* HMD, baseado em [6]

As principais vantagens deste HMD são [6]:

- flexibilidade na criação de estratégias: já que nesse HMD não existe o problema dos objetos virtuais não esconderem completamente os objetos reais;
- *wide field-of-view*: não há as distorções que ocorrem geralmente nos sistemas óticos;
- os *delays* de visão real e virtual podem ser combinados;
- mais fácil para unir o brilho dos objetos real e virtual.



Figura 3.25 - *Video see-through* HMD [13]

O sistema *monitor-based* é similar ao HMD do tipo *see-through*. O vídeo do mundo real e as imagens gráficas geradas pelo gerador de cenas são combinados, mas no caso do *monitor-based* estas são mostradas em um monitor posicionado na frente do usuário, como ilustra a Figura 3.26. Neste sistema podem ser usadas uma ou duas câmeras que podem ser estáticas ou móveis, e o usuário não necessita usar nenhum dispositivo a não ser que as imagens mostradas no monitor sejam *stereo*, o que acarretará na necessidade de uso de um par de óculos *stereo* [6].

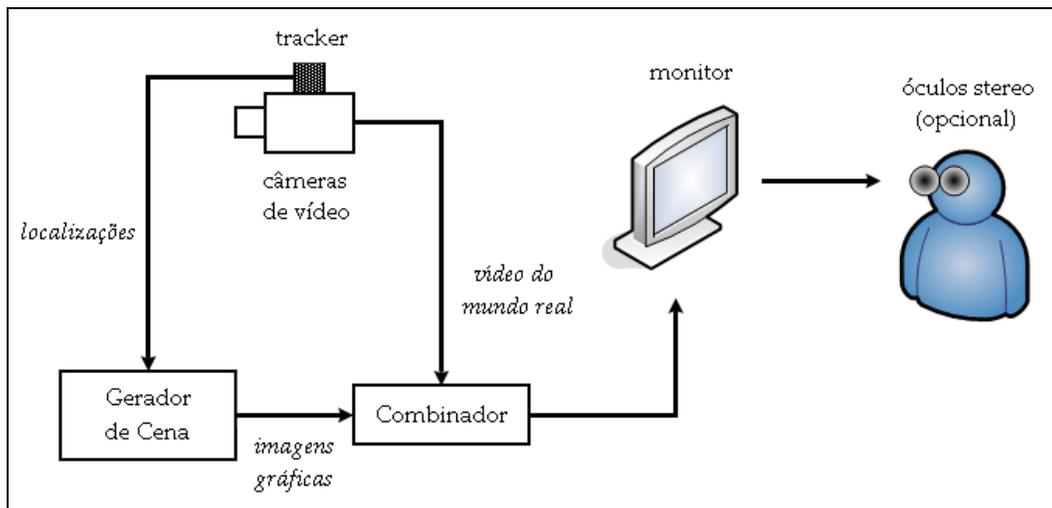


Figura 3.26 - Sistema *monitor-based*, baseado em [6]

Um sistema de exibição mais interessante e simples do ponto de vista do usuário é o VRD, que vem a ser um novo tipo de *display* que mapeia *laser* de baixa potência diretamente na retina do usuário criando a percepção de uma imagem virtual. O modo pelo qual se enxerga é o seguinte: um *laser* de diodo vermelho provê um *display* monocromático, enquanto que um *laser* de gás argônio, que produz linhas verdes e azuis, e o mesmo *laser* de diodo vermelho combinados provêm a criação de um *display full color* [37].

As principais vantagens do VRD sobre os outros *displays* são: maior escala de cores, resolução, profundidade do foco, baixo consumo de energia, custo e maior escala de aplicações, já que os mesmos podem ser usados dentro ou fora de ambientes.

3.3.2 Dispositivos Hápticos

A palavra háptico significa “próprio para tocar, sensível ao tato”. Um dispositivo háptico é aquele que envolve sensação tátil entre o computador e o usuário, usualmente através de dispositivos de entrada e/ou de saída que “sentem” o movimento do corpo. Estes dispositivos podem ser usados por pessoas com deficiências ou pessoas que aprendem melhor através de experiências táteis ou sinestésicas. Existem basicamente dois tipos de dispositivos hápticos: dispositivos que permitem ao usuário “tocar” e manipular objetos virtuais 3D e dispositivos que permitem ao usuário “sentir” texturas de objetos 2D.

As luvas (*data gloves*) pertencem ao primeiro tipo (vide Figura 3.27). Essas luvas são equipadas com sensores que sentem os movimentos da mão e fazem a interface desses movimentos com o computador, fazendo com que o usuário possa manipular objetos e/ou realizar movimentos em um ambiente virtual. Por exemplo, em um ambiente virtual, o usuário pode pegar uma bola de tênis virtual usando as luvas, e o computador mapeia o movimento e move a bola virtual, exibindo o resultado no *display*. Através da interface háptica, o usuário sentirá a bola de tênis virtual na sua mão; esta sensação é transmitida pelo computador para os sensores presentes na luva, imitando a sensação da bola de tênis na mão do usuário.



Figura 3.27 - Data gloves [21] [45]

3.3.3 Dispositivos Móveis

Algumas vezes são também denominados Dispositivos Não-Convencionais. Quando se fala em AR móvel, a maioria quase exclusiva das pesquisas nessa área ainda trabalha com o chamado *backpack* tradicional, que inclui um *notebook*, um HMD, uma ou várias câmeras e algum suporte adicional de *hardware*, vide Figura 3.28. Embora esse sistema funcione muito bem em um ambiente laboratorial quando levado ao uso no dia-a-dia vê-se que o mesmo geralmente é caro, incomodativo e requer um alto nível de especialidade em relação ao usuário.



Figura 3.28 - Backpack tradicional para AR móvel (esquerda) versus um handheld (direita) [48]

Ao mesmo tempo, uma grande massa de consumidores está interessada em dispositivos menores e mais ergonômicos, como celulares e handhelds (computadores de mão, i.e., pocket PC e PDA), acelerando com isso o desenvolvimento desta área.

A maioria das pesquisas em dispositivos móveis é feita usando PDAs, pois estes têm uma boa relação entre poder de processamento, tamanho e peso; são mais aceitáveis no que diz respeito à sociedade e sua interface equipada com tela touch screen e câmera (que funciona como um video see-through HMD) provêem aos usuários comuns uma familiaridade algumas vezes encontrada em desktops.

Este tipo de dispositivo pode ser usado em inúmeras aplicações, como por exemplo, na área de visualização e explicação com um sistema de ajuda em prédios desconhecidos (vide tópico 2.2.3), até na área de entretenimento como o sistema do Invisible Train (vide tópico 2.2.4). Têm-se ainda projetos que utilizam PDAs para dar suporte a jogos educacionais colaborativos, como por exemplo o Virtuoso [20], sendo o objetivo deste ordenar uma coleção de obras de arte de acordo com sua data de criação ao longo de uma linha do tempo, que consiste basicamente em vários marcadores pregados em uma parede, cada qual contendo uma dessas obras, e ainda podendo-se pedir ajuda ao Sr. Virtuoso, que é um personagem virtual 3D. A Figura 3.29 ilustra a aplicação.



Figura 3.29 - Sistema Virtuoso [20]

Outro dispositivo móvel que está sendo usado para as pesquisas nessa área são os telefones celulares. Estes ainda não são muito comuns, talvez devido a sua capacidade de processamento, principalmente com relação aos *notebooks*. Mas isto está mudando, pois a nova geração de telefones celulares possui telas coloridas, câmeras e redes *wireless*, características mínimas requeridas para a implementação de interfaces em AR. Uma das principais desvantagens dos telefones celulares é que com o avanço da tecnologia estes têm diminuído cada vez mais de tamanho, desafiando a construção de uma interface amigável.

Dentro desse paradigma, alguns projetos foram desenvolvidos. Entre os mais interessantes pode-se destacar o AR *Comic Book* [42], que foi desenvolvido pelo Mixed Reality Lab Singapore, e permite que crianças possam visualizar em seus telefones celulares seus personagens favoritos em terceira dimensão (vide Figura 3.30). Outro projeto interessante, também desenvolvido pelo mesmo laboratório, é o AR *Post-it* [42] que combina a rapidez das mensagens SMS (*Short Message Service*) com a tangibilidade dos marcadores de papel. A intenção é que essas mensagens só sejam mostradas se o usuário estiver dentro de um contexto espacial relevante.



Figura 3.30 - Aplicação AR *Comic Book* [42]

Outra aplicação bastante interessante é o CyPhone [38]. Este consiste em um telefone celular que além de fazer chamadas comuns ainda oferece suporte a contextos e serviços multimídia multiusuário.

4 Arquitetura para Dispositivos Não-Convencionais de Interação Utilizando Realidade Aumentada

Neste capítulo serão mostrados a proposta da arquitetura e o cenário envolvido para validar o estudo de caso. Será explicado também o funcionamento dos dispositivos inseridos neste cenário e como estes interagem com os serviços, as bibliotecas utilizadas, como será a implementação do estudo de caso que foi escolhido e um pouco sobre a infra-estrutura de rede que suporta este estudo.

4.1 Descrição do Cenário

A arquitetura proposta neste trabalho é utilizada em um cenário composto basicamente por um museu virtual. Seu principal objetivo é prover aos visitantes do museu um meio de obter informações e/ou serviços automaticamente, sem a necessidade de transações prévias no que diz respeito às redes. Seu maior diferencial consiste em permitir o uso de plataformas móveis.

Com o intuito de oferecer aos usuários maior interatividade e interoperabilidade foram usados conceitos de AR, já explicados nos capítulos anteriores, e de AN. Cada conceito foi escolhido por oferecer soluções interessantes e por representar tecnologias de ponta, fornecendo com isso uma aplicação diferenciada e nova do ponto de vista do mercado de tecnologia e informação.

Quando o visitante chegar ao museu ele receberá um *folder*, que conterà um número de acesso à rede interna do museu e alguns marcadores. É importante lembrar que não importando qual o dispositivo que o usuário esteja usando, esse número será necessário para que o mesmo possa integrar a rede interna do museu. Este número já deverá ter sido cadastrado previamente nesta rede, bastando somente que o usuário se identifique na aplicação. Outro fator de igual importância é que o usuário já deve ter instalado no seu dispositivo o *software* de interação e visualização provido pelo museu.

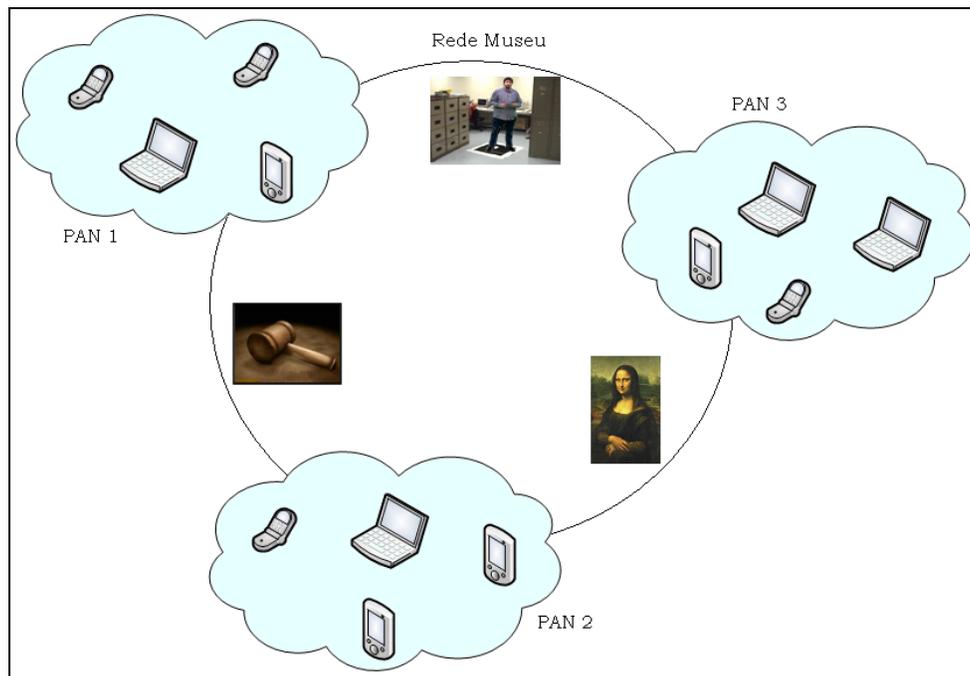


Figura 4.1 - Representação da rede interna do museu

A partir do momento que o usuário abrir o *software* do museu e digitar o número contido no *folder* ele passará a fazer parte da rede interna do museu, ilustrada na Figura 4.1. Este *software* além de prover a interação e visualização do usuário, será também o responsável por enviar à rede a classe correspondente ao dispositivo no qual ele se encontra, ou seja, se o dispositivo é um celular, um PDA ou um *laptop*. Esta informação é de extrema importância para a rede, pois dependendo da classe do dispositivo alguns serviços não serão disponibilizados ou serão disponibilizados de um modo diferenciado dos outros.

Após o usuário se integrar à rede do museu ele entrará no mesmo. Uma ressalva neste ponto é de extrema importância: quando se fala em museu, foi pensado no significado do mesmo, ou seja, um lugar destinado ao agrupamento de uma ou várias coleções de objetos. Com isso em mente, foi imaginado um museu onde ao invés das peças e/ou quadros propriamente ditos, ter-se-iam marcadores, provendo com isso uma maior interatividade já que os mesmos marcadores poderiam, por exemplo, em um dia abrigar o Museu do Louvre e no outro o Museu de Arte Moderna de Nova Iorque. O estudo de caso apresentado neste trabalho tem como cenário físico uma sala vazia onde nas paredes se encontram pregados marcadores, como exemplo pode-se observar a Figura 3.29, mais precisamente a do canto superior esquerdo.

Ao entrar no museu alguns serviços serão oferecidos ao visitante. Cada serviço disponibilizado pela rede é um BE diferente (vide Figura 4.2), os quais serão detalhados um pouco mais à frente. O primeiro serviço e o que será o estudo de caso deste trabalho é o serviço de visualização dos quadros. Este serviço funciona basicamente do seguinte modo: quando a câmera capta um marcador, o quadro que está sendo representado por ele será automaticamente carregado para a visualização do usuário independentemente do dispositivo que este estará usando.

Quando o quadro for carregado no lugar do marcador identificado, o tamanho visualizado será o tamanho original da pintura, por exemplo, o quadro da Mona Lisa tem 77 x 53 cm conseqüentemente a sua visualização em cima do marcador terá o mesmo tamanho.

Ao se aproximar do marcador algumas opções serão oferecidas ao usuário; este poderá ver um vídeo, ou poderá ler informações, sendo todas essas informações e/ou vídeos relativos a cada quadro especificamente. Por exemplo, caso o marcador que o usuário esteja visualizando seja referente à Mona Lisa, estas informações poderiam ser algo como: "o pintor foi Leonardo da Vinci e começou a pintá-lo em 1503. O

original se encontra no Museu do Louvre em Paris”.

Com relação aos vídeos estes se comportam como se o quadro fosse animado, colocado em movimento, ressaltando uma ou várias características presentes nele. Algumas considerações sobre esse serviço devem ser feitas, primeiramente o serviço de vídeo não é disponível para o usuário que estiver usando um PDA, devido ao desempenho do dispositivo não ser adequado para este tipo de aplicação, e para o usuário que estiver usando um celular não serão disponibilizados nem os vídeos nem as informações, pelo mesmo motivo de desempenho.

Outro serviço disponível é o de *chat*, que nada mais é do que um serviço de conversa via voz, como um telefone. A vantagem desse serviço com relação a um telefone convencional vem a ser a interatividade do mesmo com o usuário. Dentro desse mesmo paradigma tem-se também o serviço de conferência, que pode ser entendido como uma palestra transmitida em tempo real para o usuário onde quer que ele esteja. O diferencial desse serviço é que com o auxílio de um marcador o usuário poderá visualizar o palestrante em 3D.

O último serviço a ser oferecido é o leilão, que funciona basicamente como um leilão real em que uma pessoa faz lances com o intuito de adquirir algo. Com a ajuda de um marcador o usuário poderá visualizar as peças que serão leiloadas e ver algumas informações relevantes da mesma. O usuário poderá inclusive dar lances para adquirir a peça. Tudo isto será possível sem a necessidade da pessoa estar no local onde o leilão estará acontecendo.

4.2 Arquitetura

A arquitetura que está sendo usada para a implementação desse cenário de AR será descrita a seguir. Pode-se observar que a figura 4.2 é subdividida em várias partes, e cada uma destas será mais bem detalhada posteriormente.

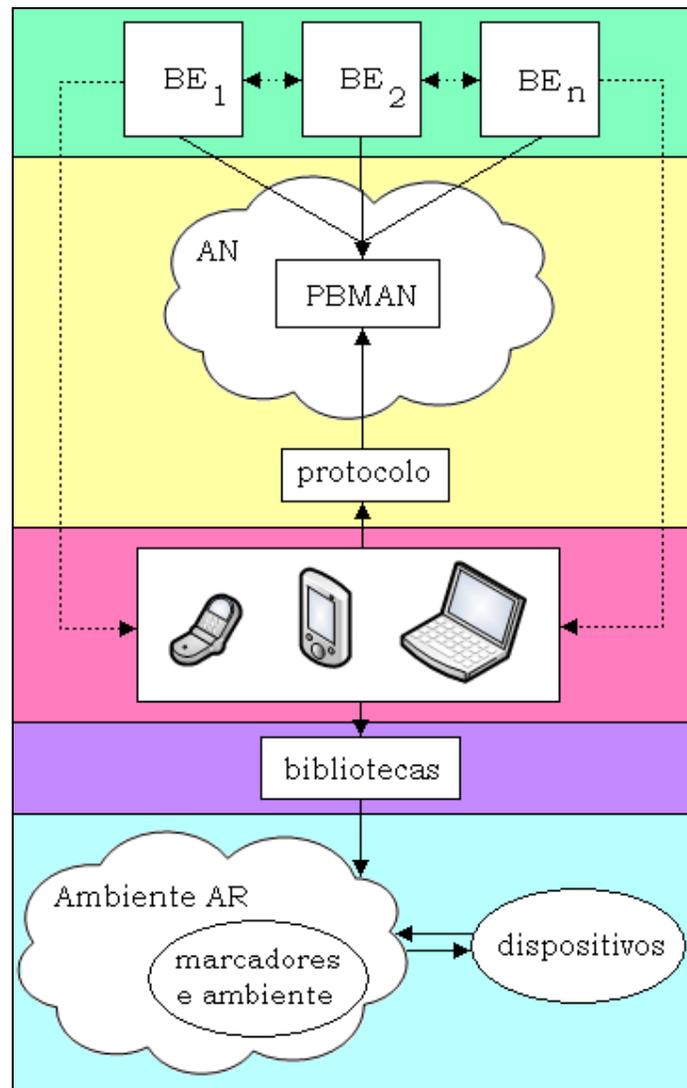


Figura 4.2 - Arquitetura para dispositivos não-convencionais de interação utilizando AR

Cada parte da figura 4.2 representa uma sub-arquitetura (numerada de cima para baixo):

- Parte 1 -> representa o *Back-End* (BE) da aplicação. O funcionamento destes será explicado em mais detalhes na seção 4.5.
- Parte 2 -> representa a tecnologia de AN sobre a qual foi desenvolvido o cenário; esta é a parte do suporte de redes. Será explicada na seção 4.6.
- Parte 3 -> representa todos os dispositivos que foram usados. Estes serão mais detalhados na próxima seção.
- Parte 4 -> representa as bibliotecas que foram usadas na implementação da aplicação de AR para cada dispositivo. Quais bibliotecas foram usadas, como elas interagem com cada dispositivo e entre elas será explicado na seção 4.4.
- Parte 5 -> representa o cenário desenvolvido e como cada dispositivo estará se comunicando com ele. Esta parte já foi descrita na seção anterior.

4.3 Dispositivos

Os dispositivos móveis que foram usados são os seguintes: *notebook*, PDA e celular. Nesta seção será explicado como cada dispositivo foi implementado e mais alguns detalhes adicionais que se façam necessário. As bibliotecas que foram usadas na concepção das arquiteturas internas de cada dispositivo serão explicadas na próxima seção (vide seção 4.4).

4.3.1 Notebook

O *laptop* tem algumas especificações diferentes dos outros dispositivos. Por ser o dispositivo com o maior poder de processamento, ele é o único (até o momento) que poderá receber todos os serviços e o que possibilitará uma maior interação com o cenário. Para o uso deste dispositivo também é necessário o uso do *video see-through* HMD (vide tópico 3.3.1) e das luvas (vide tópico 3.3.2). A interface com o usuário será feita através de projeções no HMD e com as luvas o usuário irá interagir com esta interface.

A interface com o usuário terá alguns cubos, estes com comportamento de esferas, na parte inferior que irão representar os serviços ativos. No canto esquerdo um menu *auto-hide* conterá os serviços que são providos pela rede do museu e alguns ícones úteis como, por exemplo, uma “lixeira” que servirá para fechar os serviços ativos.

A arquitetura interna, ilustrada pela figura 4.3, é composta por uma aplicação principal implementada usando o *engine* OGRE [33]. O ARToolkit e o OGRE AR fazem parte de um módulo de reconhecimento de padrões e aquisição de vídeo, onde este último funciona como uma camada de abstração entre a aplicação principal e o ARToolKit. O ODE [32] é usado na interface com o usuário, pois nesta estarão presentes alguns objetos que irão se comportar com forças físicas atuando sobre eles, e o OGRE ODE é usado como um encapsulamento do ODE para a plataforma principal. Além destes, CIDA [16] será usada para prover uma interação com os dispositivos (luva e HMD), para usá-la é necessário o uso de *plugins*. E, finalmente, a aplicação principal possui alguns *plugins* do DirectX 9 [29] e OpenGL [34] que são os sistemas usados para renderizar a cena.

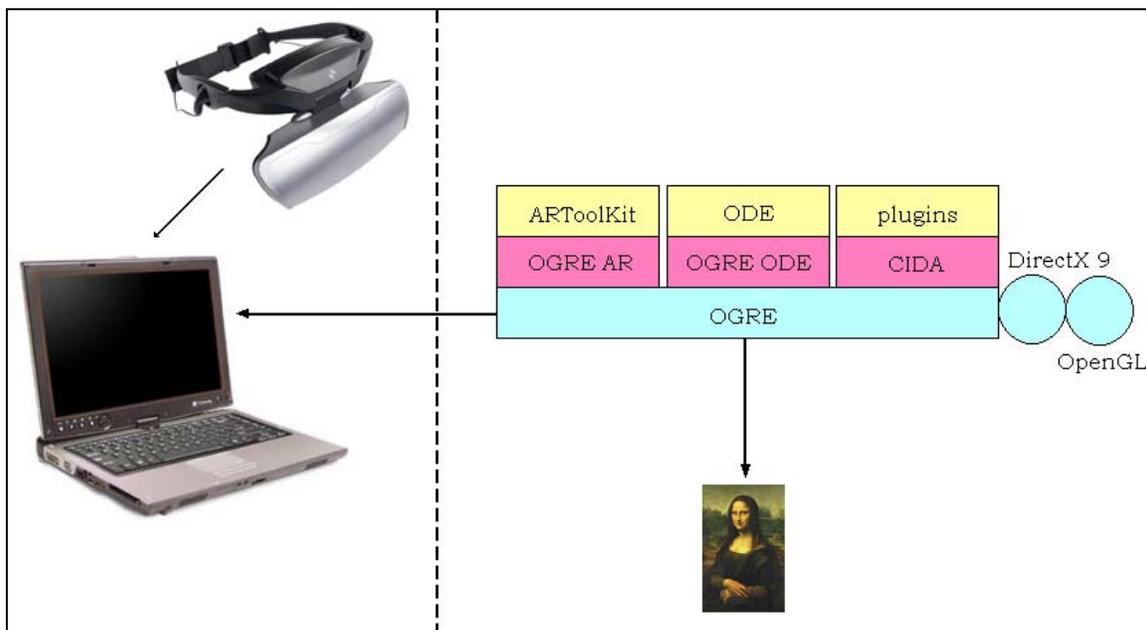


Figura 4.3 - Arquitetura interna do *laptop*

4.3.2 PDA

O *palm* (ou PDA) usado é um PocketPC e a câmera que será usada para a detecção dos marcadores é uma câmera extra, pois este *palm* não possui câmera embutida, o sistema operacional é o Windows CE.

No caso deste dispositivo alguns serviços não poderão ser apresentados, visto que o desempenho deste não é bom, principalmente no caso de exibição de vídeos. A interação do PDA com o usuário é feita do modo convencional, ou seja, usando a caneta, e todas as exibições serão mostradas na tela. Por todos esses fatores, a interface com o usuário é simples e diferente da interface quando o usuário está usando um *laptop*.

O funcionamento interno é basicamente o seguinte: o SDK (*Software Development Kit*) da câmera irá fazer a aquisição do vídeo, enquanto que o ARToolKitPlus irá identificar os padrões desses marcadores e o OGRE, como aplicação principal, irá prover a localização e orientação desses padrões que foram encontrados. O OpenGL ES [35] e o Klimt [26] são os *plugins* dos sistemas que irão fazer a renderização da cena. A Figura 4.4 ilustra a arquitetura.

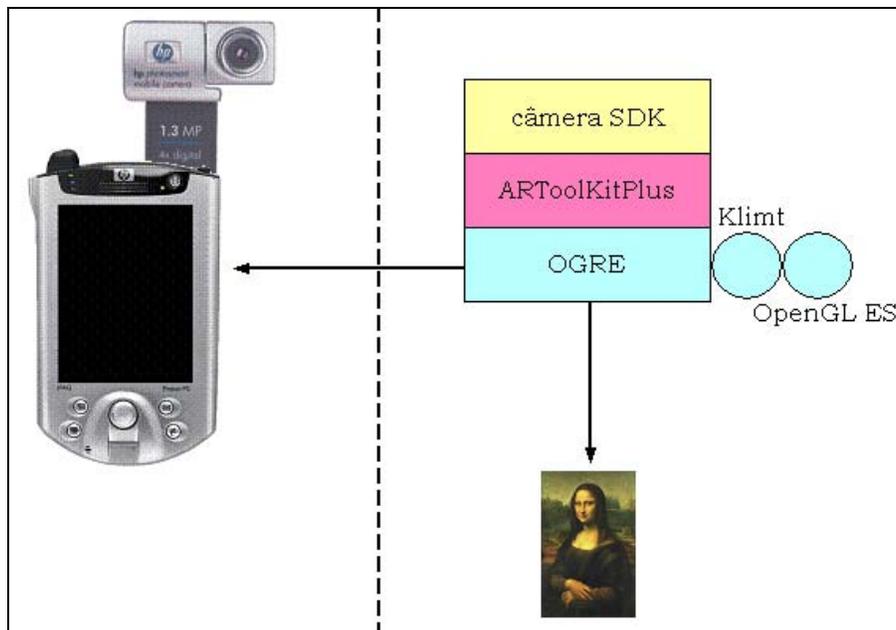


Figura 4.4 - Arquitetura interna do PDA

4.3.3 Celular

O celular que será usado é um Nokia 6600 que possui câmera integrada ao aparelho e o sistema operacional é o Symbian OS.

Este dispositivo é o que irá conter o menor número de serviços disponíveis, principalmente devido a sua baixa performance. Conseqüentemente, isto irá acarretar em uma interface com o usuário extremamente simples e convencional, onde a interação entre este e o usuário será feita através do teclado do mesmo.

O funcionamento interno do celular é basicamente o mesmo do PDA, e a diferença é que a API (*Application Program Interface*) do Symbian é que irá capturar as imagens do ambiente e a aplicação principal agora não é mais o OGRE, e sim o OpenGL ES. A Figura 4.5 ilustra a arquitetura do celular.

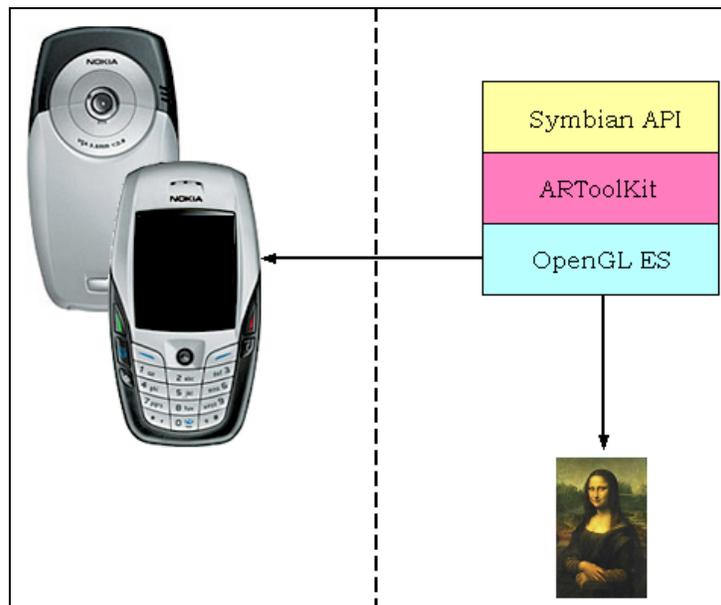


Figura 4.5 - Arquitetura interna do celular

4.4 Bibliotecas

Várias bibliotecas foram usadas no desenvolvimento dessas arquiteturas. Nesta seção serão mostradas e detalhadas algumas delas, com exceção do ARToolKit (vide tópico 2.2.1) e do ARToolKitPlus (vide subtópico 2.2.1.1) que já foram explicadas anteriormente.

4.4.1 OGRE

O OGRE (*Object-Oriented Graphics Rendering Engine*) [33] vem a ser um motor gráfico *open source* que funciona na maioria das plataformas existentes e oferece uma vasta gama de *plugins*, ferramentas e *add-ons* que favorecem a criação de diversos tipos de aplicações gráficas empregando vários conceitos inerentes ao desenvolvimento das mesmas.

O propósito do OGRE não é ser um motor de jogos, e sim um motor de renderização genérico que pode ser incorporado a bibliotecas de tratamento de entradas, processamento de som e plataformas que disponibilizem algoritmos de inteligência artificial.

4.4.2 ODE

O ODE (*Open Dynamics Engine*) [32] é uma biblioteca *open source* de alta performance para simular corpos rígidos dinâmicos. Esta API foi escrita em C e C++, é independente de plataforma e possui entre as suas primitivas corpos rígidos e flexíveis, juntas, contato e colisão, atrito e molas, ou seja, é capaz de produzir diversos comportamentos físicos. Esta biblioteca é muito usada para simulação de veículos, objetos em ambientes de VR e bonecos virtuais.

4.4.3 CIDA

CIDA (*Chaotic Interaction Device Abstraction*) [16] é uma plataforma de gerenciamento de dispositivos de

entrada, cuja principal característica é prover uma camada de abstração entre os sistemas interativos e estes dispositivos. Esta plataforma disponibiliza para a aplicação, em tempo de execução, todos os dispositivos que possuem as características necessárias para interagir com a mesma, como, por exemplo, botões, eixos, graus de liberdade, entre outros. Os dispositivos em CIDA são agrupados em quatro classes distintas, de acordo com as suas características funcionais: *keyboard*, *joystick*, *tracker* e *pointer device*.

O suporte a estes dispositivos de entrada é viabilizado através de *plugins* específicos, que determinam quais são as características de cada classe que o dispositivo oferece. Esta característica simplifica a adição de novos dispositivos de interação a sistemas baseados na plataforma, pois o código fonte da aplicação não precisa ser modificado. Além disso, um mesmo dispositivo pode ser enxergado de formas diferentes pela aplicação, uma vez que as suas capacidades estão especificadas em cada *plugin*. Por exemplo, um teclado convencional poderá ser visto pela aplicação como um dispositivo da classe *keyboard* ou como dois dispositivos da classe *joystick*, dependendo do *plugin* a ser utilizado.

Esta plataforma também oferece transparência de conectividade, pois os dispositivos podem estar conectados a um computador diferente daquele que executa a aplicação, sem que o código fonte necessite ser modificado. Esta característica facilita consideravelmente o gerenciamento de dispositivos de entrada em ambientes de VR complexos, nos quais o uso de vários computadores é necessário para garantir o processamento em tempo real dos dados de entrada. Além disso, o desenvolvedor não precisa se preocupar em usar *device drivers* específicos para determinados periféricos, pois CIDA oferece à aplicação uma visão uniforme das capacidades de cada dispositivo.

4.4.4 DirectX

O DirectX [29] foi desenvolvido em 1995 pela Microsoft, e consiste em um pacote de APIs multimídia que foi desenvolvido em cima do sistema operacional Windows. Esta API provê um acesso especializado para algumas características de *hardware* sem ter que escrever em códigos específicos deste. A API, também, fornece às aplicações multimídia controle de funções de baixo nível como, por exemplo, aceleração gráfica 3D e controle de efeitos de som.

4.4.5 OpenGL

O OpenGL (*Open Graphics Library*) [34] foi criado em 1992 pela *Silicon Graphics Incorporated*, e é um ambiente para o desenvolvimento de aplicações 2D e 3D portáteis e interativas. Esta API foi criada para o desenvolvimento de cenas 3D complexas a partir de primitivas simples, e é amplamente utilizada em sistemas CDS, realidade virtual, visualização científica e de informação e desenvolvimento de videogames.

4.4.5.1 OpenGL ES

O OpenGL ES (*Open Graphics Library for Embedded Systems*) [35] foi criado pelo *Khronos Group*, e vem a ser um subconjunto da API gráfica 3D do OpenGL que foi especificamente desenvolvido para dispositivos como telefones celulares, PDAs e consoles de videogame.

4.4.6 Klimt

O Klimt [26], também conhecido por SoftGL, é uma biblioteca *open source* 3D, específica para PDAs e celulares e foi baseada no OpenGL e OpenGL ES. Esta biblioteca foi desenvolvida para ser independente de *hardware* e rápido, e seu único requisito consiste em que o *framebuffer* do dispositivo a ser usado seja de 16 bits. O Klimt está apto para ser usado nas plataformas Windows, PocketPC, MS Smartphone, Linux em

4.5 Back-End

Como já foi dito anteriormente o BE é a aplicação propriamente dita do *software* do museu, e cada serviço possui um BE específico para o mesmo. Cada serviço tem suas características próprias, mas a comunicação entre cada um deles, o PBMAN e os dispositivos acontece basicamente do mesmo modo.

Como a comunicação entre eles é similar e foi preciso escolher um caso de uso, será falado aqui especificamente sobre o serviço da galeria de quadros, vide Figura 4.6. Para tanto é interessante que seja observada a comunicação entre as três partes descritas acima.

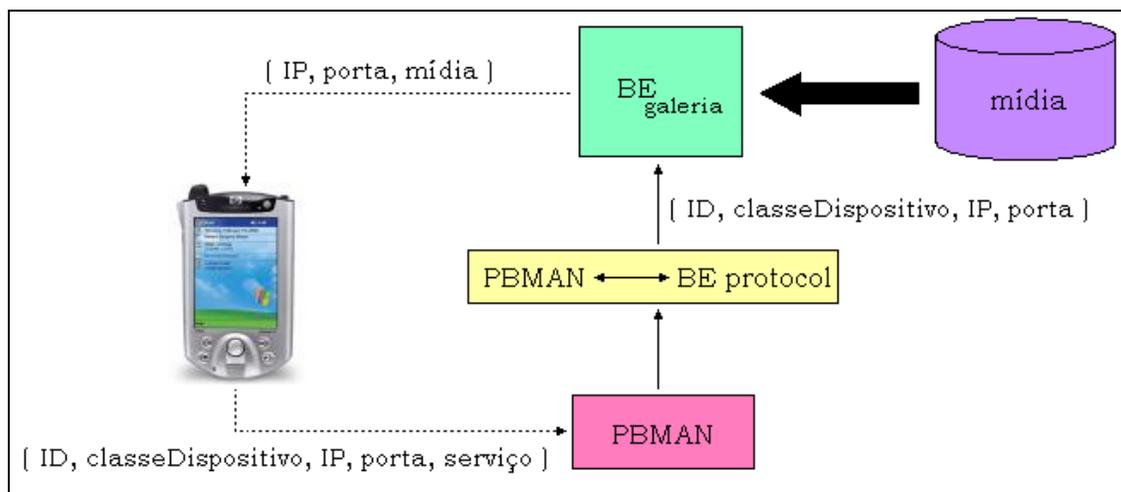


Figura 4.6 – Comunicação entre dispositivo, PBMAN e BE

A figura 4.6 representa a parte superior da Figura 4.2 e ilustra a comunicação descrita anteriormente. Lembrando o que já foi dito anteriormente, quando a câmera do dispositivo detectar a presença de um marcador no ambiente, o dispositivo irá enviar uma mensagem (com formato pré-estabelecido) ao PBMAN solicitando as mídias que sejam referentes àquele marcador específico. Essa mensagem contém o ID do marcador que foi detectado, a classe do dispositivo (*notebook*, *palm* ou celular) que enviou o pedido, o endereço IP (*Internet Protocol*) deste dispositivo, a porta na qual a mídia deve ser recebida e qual é o serviço que requisitou as mídias. De posse dessas informações o PBMAN irá verificar nas políticas que foram implementadas, dependendo do serviço (neste caso o serviço é a galeria de quadros), quais os tipos de mídia que aquele dispositivo poderá receber. Por exemplo, uma de suas políticas diz que se a classe do dispositivo for celular, este só poderá receber a imagem do quadro. Depois de verificar suas políticas, o PBMAN irá enviar uma mensagem ao BE específico do serviço (neste caso o BE galeria) que foi solicitado uma mensagem contendo o ID do marcador detectado, a classe do dispositivo, o endereço IP e a porta. O BE então irá buscar no banco de dados as mídias que podem ser recebidas de acordo com a classe do dispositivo que a requisitou. Após a busca este irá retornar ao dispositivo uma mensagem contendo o endereço IP do dispositivo, a porta na qual a mídia será recebida e a(s) mídia(s) correspondentes ao serviço e ao dispositivo.

Este ciclo se repete para todos os serviços disponíveis, ciclo este que pode diferir em alguns aspectos, principalmente dependendo do dispositivo, mas que basicamente se mantém o mesmo.

4.6 Infra-Estrutura de Comunicação

As AN por sua natureza são ambientes sensitivos, adaptativos e compreensivos. Elas são capazes de fazer a composição entre quaisquer tipos de redes com o intuito de compartilhar serviços com seus usuários, por isso adaptativas, e ainda agrupam os usuários que estão interessados em um determinado serviço já que as redes irão se compor formando uma grande rede, por isso compreensivas. Uma AN pode, então, ser assim caracterizada por alguns elementos básicos: ubiqüidade (presença em todo lugar), transparência e inteligência [23].

O cenário descrito na seção anterior estará rodando sobre uma AN dentro desta se encontra o PBMAN (*Police-Based Management for Ambient Networks*), que é quem vai gerenciar os pedidos requisitados pelos dispositivos. Explicando de um modo mais simples, os dispositivos irão requisitar ao PBMAN os arquivos de mídia e este irá requisitar ao BE, para que então o BE mande diretamente ao dispositivo a resposta ao pedido feito. Dentro do PBMAN foram implementadas as políticas que irão definir quais os dispositivos que podem receber e quais tipos de mídia eles podem receber, visto que nem todos os dispositivos podem receber qualquer tipo de mídia, e que os mesmos também não sabem que tipos de mídia podem receber. Basicamente então é o PBMAN que vai direcionar os pedidos a cada BE específico, simplificando com isto a implementação deste último.

5 Conclusão e Trabalhos Futuros

Neste trabalho foi mostrado o estado da arte na área de AR, onde a mesma está sendo usada e como ela pode ajudar em algumas outras áreas. Também foram descritas algumas das várias técnicas usadas em ambientes tanto de VR quanto de AR.

Também foi apresentada aqui uma proposta para uma arquitetura de uma aplicação para um museu, e um estudo de caso de um serviço referente a essa aplicação. A arquitetura mostrou ser suficiente para uma primeira demonstração, mas ainda há muito a ser aprimorado.

Alguns projetos futuros podem incluir uma maior oferta de serviços e uma expansão da aplicação, podendo ser usada não somente em museus, mas também em feiras de exposição, por exemplo. Um serviço muito útil seria o chamado “*You see what I see*” que consiste em permitir que um usuário possa visualizar algo que outro usuário está vendo no momento. Seria interessante também que todos os dispositivos pudessem permitir o uso de todos os serviços oferecidos. Outro trabalho interessante seria ter um serviço de conferência que permitisse uma interação entre o palestrante e o usuário para que esse pudesse, por exemplo, fazer perguntas e tê-las respondidas em tempo real.

Este trabalho contribuiu em vários aspectos. Primeiramente não existem muitos trabalhos que abordem os vários pontos sobre AR, nem em português e nem em inglês. *Surveys*, como este tipo de trabalho é chamado, são de extrema importância, pois sintetizam os vários conceitos e diferentes pontos de vista de um mesmo assunto. Segundo, o cenário proposto neste trabalho utiliza uma grande gama de conceitos e tecnologias que são consideradas novas no meio acadêmico como, por exemplo, as AN e os dispositivos móveis.

6 Referências

6.1 Bibliografia

- [1] A Table-Top Augmented Reality Application. Disponível: *Multimedia Concepts and Applications* site. URL: http://mm-werkstatt.informatik.uni-augsburg.de/project_details.php?id=1. Consultado em 26 dez. 2005.
- [2] Ambient Networks Project. Disponível: *Ambient Networks Project* site. URL: <http://www.ambient-networks.org>. Consultado em 15 jan. 2006.
- [3] ANABUKI, M.; KAKUTA, H.; YAMAMOTO, H.; TAMURA, H. Welbo: An Embodied Conversational Agent Living in Mixed Reality Space. In: CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 2000, The Netherlands, *Demo...* New York: ACM Press, 2000. p.10-11.
- [4] ARTag. Disponível: *ARTag* site. URL: <http://www.cv.iit.nrc.ca/research/ar/artag/>. Consultado em 25 dez. 2005.
- [5] ARToolKit Documentation. Disponível: *Human Interface Technology Laboratory* site. URL: <http://www.hitl.washington.edu/artoolkit/documentation/>. Consultado em 25 dez. 2005.
- [6] AZUMA, R. T. A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments*, v. 6, n. 4, p. 355-385, ago. 1997.
- [7] AZUMA, R. T.; BAILLOT, Y.; BEHRINGER, R.; FEINER, S.; JULIER, S.; MACINTYRE, B. Recent Advances in Augmented Reality. *IEEE Computer Graphics and Applications*, v. 21, n. 6, p. 34-47, nov./dez. 2001.
- [8] BÉRARD, F. The Magic Table: Computer-Vision Based Augmentation of a Whiteboard for Creative Meetings. In: WORKSHOP ON PROJECTOR-CAMERA SYSTEMS, 2003, France, *Proceedings...* Washington: IEEE Computer Society Press, 2003.
- [9] BOWMAN, D. A.; KRUIJFF, E.; LAVIOLA, JR, J. J.; POUPYREV, I. *3D User Interfaces: Theory and Practice*. USA: Addison-Wesley, 2004. 512 p.
- [10] BROLL, W.; LINDT, I.; OHLENBURG, J.; HERBST, I.; WITTKÄMPER, M.; NOVOTNY, T. An Infrastructure for Realizing Custom - Tailored Augmented Reality User Interfaces. *IEEE Transactions on Visualization and Computer Graphics*, v. 11, n. 6, p. 722-733, nov. 2005.
- [11] BROLL, W.; LINDT, I.; OHLENBURG, J.; WITTKÄMPER, M.; YUAN, C.; NOVOTNY, T.; SCHIECK, A. F.; MOTTRAM, C.; STROTHMANN, A. ARTHUR: A Collaborative Augmented Environment for Architectural Design and Urban Planning. *Journal of Virtual Reality and Broadcasting*, v. 1, n. 1, dez. 2004.
- [12] BUCHMANN, V.; VIOLICH, S.; BILLINGHURST, M.; COCKBURN, A. FingARtips – Gesture Based Direct Manipulation in Augmented Reality. In: INTERNATIONAL CONFERENCE ON COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES (GRAPHITE), 2., 2004, Singapore, *Proceedings...* New York: ACM Press, 2004. p. 212-221.

- [13] Canon Technology - Mixed Reality Technology. Disponível: *Canon Inc* site. URL: <http://www.canon.com/technology/software/12.html>. Consultado em 17 jan. 2006.
- [14] Computer Graphics and Scientific Visualization. Disponível: *Ronald Azuma's Computer Science Department Home Page* site. URL: http://www.cs.unc.edu/~azuma/azuma_AR.html. Consultado em 17 jan. 2006.
- [15] DART. Disponível: *DART Homepage* site. URL: <http://www.cc.gatech.edu/dart/>. Consultado em 10 jan. 2006.
- [16] FARIAS, T. S. M. C. F.; TEIXEIRA, J. M. X. N.; RODRIGUES, C. E. M.; PESSOA, S. A.; TEICHRIEB, V.; KELNER, J. CIDA: Uma Plataforma de Gerenciamento de Dispositivos. In: SIMPÓSIO BRASILEIRO DE JOGOS PARA COMPUTADOR E ENTRETENIMENTO, 2005, São Paulo, *Posters...* 2005.
- [17] FIALA, M. ARTag, an Improvement Marker System Based on ARToolKit. *NRC Technical Report*, n. 47166, jul. 2004.
- [18] FIALA, M. ARTag Revision 1, a Fiducial Marker System Using Digital Techniques. *NRC Technical Report*, n. 47419, nov. 2004.
- [19] GEIGER, C.; SCHMIDT, T.; STOCKLEIN, J. Rapid Development of Expressive AR Applications. In: INTERNATIONAL SYMPOSIUM ON MIXED AND AUGMENTED REALITY (ISMAR), 3., 2004, USA, *Proceedings...* Washington: IEEE Computer Society, 2004. p. 292-293.
- [20] Handheld Augmented Reality. Disponível: *Studierstube Augmented Reality Project* site. URL: http://studierstube.icg.tu-graz.ac.at/handheld_ar/. Consultado em 18 jan. 2006.
- [21] Input Devices. Disponível: *ICS online* site. URL: <http://www.ics.uci.edu/~kobsa/courses/ICS104/course-notes/input-devices.htm>. Consultado em 17 jan. 2006.
- [22] Introduction to Augmented Reality. Disponível: *Jim Vallino's RIT SE Department Home Page* site. URL: <http://www.se.rit.edu/~jrv/>. Consultado em 11 jan. 2006.
- [23] JACQUET, P. ERCIM News. Disponível: *The European Research Consortium for Informatics and Mathematics* site. URL: http://www.ercim.org/publication/Ercim_News/enw48/jacquet.html. Consultado em 15 jan. 2006.
- [24] KATO, H.; BILLINGHURST, M. Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System. In: INTERNATIONAL WORKSHOP ON AUGMENTED REALITY (IWAR), 2., 1999, USA, *Proceedings...* Washington: IEEE Computer Society, 1999. p. 85-94.
- [25] KATO, H.; BILLINGHURST, M.; POUPYREV, I.; IMAMOTO, K.; TACHIBANA, K. Virtual Object Manipulation on a Table-Top AR Environment. In: INTERNATIONAL SYMPOSIUM ON AUGMENTED REALITY (ISAR), 2000, Germany, *Proceedings...* 2000. p. 111-119.
- [26] Klimt - the Open Source 3D Graphics Library for Mobile Devices. Disponível: *Studierstube Augmented Reality Project* site. URL: <http://studierstube.icg.tu-graz.ac.at/klimt/>. Consultado em 30 jan. 2006.

- [27] MACINTYRE, B.; GANDY, M.; BOLTER, J.; DOW, S.; HANNIGAN, B. DART: The Designer's Augmented Reality Toolkit. In: INTERNATIONAL SYMPOSIUM ON MIXED AND AUGMENTED REALITY (ISMAR), 2., 2003, *Proceedings...* Washington: IEEE Computer Society, 2003. p. 329.
- [28] MALIK, S.; ROTH, G.; MCDONALD, C. Robust 2D Tracking for Real-Time Augmented Reality. In: VISION INTERFACE (VI), 2002, Canada, *Proceedings...* p. 399-406.
- [29] Microsoft DirectX Technology. Disponível: *Microsoft* site. URL: <http://www.microsoft.com/windows/directx/default.aspx?url=/windows/directx/productinfo/overview/default.htm>. Consultado em 30 jan. 2006.
- [30] MILGRAM, P.; KISHINO, F. A Taxonomy of Mixed Reality Visual Displays. *IEICE Transactions on Information Systems*, v. E77-D, n. 12, p. 1321-1329, dez. 1994. Special Issue on Networked Reality.
- [31] Mixed Reality Research. Disponível: *Mixed Reality Lab Singapore* site. URL: <http://mixedreality.ntu.edu.sg/research.htm>. Consultado em 27 jan. 2006.
- [32] ODE. Disponível: *Open Dynamics Engine - ODE* site. URL: <http://www.ode.org/>. Consultado em 28 jan. 2006.
- [33] OGRE. Disponível: *OGRE 3D: Open Source Graphics Engine* site. URL: <http://www.ogre3d.org/>. Consultado em 28 jan. 2006.
- [34] OpenGL - The Industry Standard for High Performance Graphics. Disponível: *OpenGL* site. URL: <http://www.opengl.org/>. Consultado em 30 jan. 2006.
- [35] OpenGL ES. Disponível: *The Khronos Group* site. URL: <http://www.khronos.org/>. Consultado em 30 jan. 2006.
- [36] PIEKARSKI, W.; THOMAS, B. ARQuake: The Outdoor Augmented Reality Gaming System. *Communications of the ACM*, v. 45, n. 1, p. 36-38, jan. 2002.
- [37] PRYOR, H. L.; FURNESS III, T. A.; VIIRRE, E. The Virtual Retinal Display: A New Display Technology Using Scanned Laser Light. In: HUMAN FACTORS AND ERGONOMIC SOCIETY, 42., 1998, USA, *Proceedings...* 1998. p. 1570-1574.
- [38] PYSSYSALO, T.; REPO, T.; TURUNEN, T.; LANKILA, T.; RÖNING, J. CyPhone: Bringing Augmented Reality to Next Generation Mobile Phones. In: DESIGNING AUGMENTED REALITY ENVIRONMENTS (DARE), 2000, Denmark, *Proceedings...* New York: ACM Press, 2000. p. 11-21.
- [39] REITMAYR, G.; SCHMALSTIEG, D. Location based Applications for Mobile Augmented Reality. In: AUSTRALIAN USER INTERFACE CONFERENCE, 4., 2003, Australia, *Proceedings...* Australian Computer Society Inc., 2003. p. 65-73.
- [40] SATO, Y.; SAITO, M.; KOIKE, H. Real-Time Input of 3D Pose and Gestures of a User's Hand and Its Applications for HCI. In: VIRTUAL REALITY CONFERENCE, 2001, *Proceedings...* Washington: IEEE Computer Society, 2001. p. 79-86.
- [41] SIGGRAPH Course Notes. Disponível: *Three Dimensional User Interface* site. URL: www.3dui.org. Consultado em 20 fev. 2006.

- [42] SINGH, S.; CHEOK, A. D.; NG, G. L.; FARBIZ, F. 3D Augmented Reality Comic Book and Notes for Children using Mobile Phones. In: CONFERENCE ON INTERACTION DESIGN AND CHILDREN: BUILDING A COMMUNITY, 2004, USA, *Proceedings...* New York: ACM Press, 2004. p. 149-150.
- [43] ULLMER, B.; ISHII, H. Emerging Frameworks for Tangible User Interfaces. *IBM Systems Journal*, v. 39, n. 3/4, p. 915-931, 2000.
- [44] ULLMER, B.; ISHII, H. The metaDESK: Models and Prototypes for Tangible User Interfaces. In: SYMPOSIUM ON USER INTERFACE SOFTWARE AND TECHNOLOGY, 10., 1997, Canada, *Proceedings...* New York: ACM Press, 1997. p. 223-232.
- [45] Virtual Reality Experience. Disponível: *Cursus Fotografie* site. URL: <http://www.fotobeurs.com/cursus/stereografie/?id=95&nm=HMD%203D-bril>. Consultado em 17 jan. 2006.
- [46] WAGNER, D. Augmented Reality on Truly Mobile Devices. Disponível: *Studierstube Augmented Reality Project* site. URL: http://studierstube.icg.tu-graz.ac.at/handheld_ar/artoolkitplus.php. Consultado em 30 jan. 2006.
- [47] WAGNER, D.; BARAKONYI, I. Augmented Reality Kanji. In: INTERNATIONAL SYMPOSIUM ON MIXED AND AUGMENTED REALITY (ISMAR), 2., 2003, Japan, *Demo...* IEEE/ACM, 2003.
- [48] WAGNER, D.; PINTARIC, T.; LEDERMANN, F.; SCHMALSTIEG, D. Towards Massively Multi-User Augmented Reality on Handheld Devices. In: INTERNATIONAL CONFERENCE ON PERVASIVE COMPUTING, 3., 2005, Germany, *Proceedings...* 2005. p. 208-219.
- [49] WAGNER, D.; SCHMALSTIEG, D. A Handheld Augmented Reality Museum Guide. In: IADIS INTERNATIONAL CONFERENCE ON MOBILE LEARNING, 2005, *Proceedings...*
- [50] WAGNER, D.; SCHMALSTIEG, D. ARToolKit on the PocketPC Platform. In: IEEE INTERNATIONAL AUGMENTED REALITY TOOLKIT WORKSHOP, 2., 2003, Japan.
- [51] WAGNER, D.; SCHMALSTIEG, D. First Steps Towards Handheld Augmented Reality. In: INTERNATIONAL SYMPOSIUM ON WEARABLE COMPUTERS (ISWC), 7., 2003, USA, *Proceedings...* Washington: IEEE Computer Society, 2003. p. 127-137.
- [52] WOODWARD, C.; HONKAMAA, P.; JÄPPINEN, J.; PYÖKKIMIES, E. CamBall – Augmented Networked Table Tennis Played with Real Rackets. In: INTERNATIONAL CONFERENCE ON ADVANCES IN COMPUTER ENTERTAINMENT TECHNOLOGY, 2004, Singapore, *Proceedings...* New York: ACM Press, 2004. p. 275-276.
- [53] ZHOU, Z.; CHEOK, A. D.; LI, Y.; KATO, H. Magic Cubes for Social and Physical Family Entertainment. In: CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS, 2005, USA, *Proceedings...* New York: ACM Press, 2005. p. 1156-1157.

6.2 Referências Adicionais

- [54] CHATFIELD, C.; CARMICHAEL, D.; HEXEL, J.; KUMMERFELD, B. AR phone: Accessible Augmented Reality in the Intelligent Environment. In: CONFERENCE OF THE COMPUTER-HUMAN INTERACTION SPECIAL INTEREST GROUP (CHISIG), 19., 2005, Australia, *Proceedings...* Australia: CHISIG of Australia, 2005. p. 1-10.

- [55] DIVERDI, S.; NURMI, D.; HÖLLERER, T. ARWin – A Desktop Augmented Reality Window Manager. In: INTERNATIONAL SYMPOSIUM ON MIXED AND AUGMENTED REALITY (ISMAR), 2., 2003, Japan, *Proceedings...* Washington: IEEE Computer Society, 2003. p. 298-299.
- [56] KAWASHIMA, T.; IMAMOTO, K.; KATO, H.; TACHIBANA, K.; BILLINGHURST, M. Magic Paddle: A Tangible Augmented Reality Interface for Object Manipulation. In: INTERNATIONAL SYMPOSIUM ON MIXED REALITY (ISMAR), 2., 2001, Japan, *Proceedings...* Washington: IEEE Computer Society, 2001. p. 194-195.
- [57] KIM, K.; LEE, M.; PARK, Y.; LEE, J. ARPushPush: Augmented Reality Game in Indoor Environment. In: INTERNACIONAL WORKSHOP ON PERVASIVE GAMING APPLICATIONS, 2., 2005, Germany, *Proceedings...* 2005.
- [58] LEE, G. A.; BILLINGHURST, M.; KIM, G. J. Occlusion Based Interaction Methods for Tangible Augmented Reality Environments. In: SIGGRAPH INTERNATIONAL CONFERENCE ON VIRTUAL REALITY CONTINUUM AND ITS APPLICATIONS IN INDUSTRY (VRCAI), 2004, Singapore, *Proceedings...* New York: ACM Press, 2004. p. 419-426.
- [59] LEE, M.; WOO, W. ARKB: 3D Vision-Based Augmented Reality Keyboard. In: INTERNATIONAL CONFERENCE ON ARTIFICIAL REALITY AND TELEXISTENCE (ICAT), 2003, Japan, *Proceedings...* 2003. p. 54-57.
- [60] MARTENS, J. B.; QI, W.; ALIAKSEYEU, D.; KOK, A. J. F.; LIERE, R. van. Experiencing 3D Interactions in Virtual Reality and Augmented Reality. In: EUROPEAN SYMPOSIUM ON AMBIANT INTELLIGENCE (EUSAI), 2., 2004, The Netherlands, *Proceedings...* 2004. p. 25-28.
- [61] RASKAR, R.; WELCH, G.; CUTTS, M.; LAKE, A.; STESIN, L.; FUCHS, H. The Office of the Future: A Unified Approach to Image-Based Modeling and Spatially Immersive Displays. In: CONFERENCE ON COMPUTER GRAPHICS AND INTERACTIVE TECHNIQUES, 25., 1998, USA, *Proceedings...* New York: ACM Press, 1998. p. 179-188.
- [62] REKIMOTO, J.; NAGAO, K. The World through the Computer: Computer Augmented Interaction with Real World Environments. In: SYMPOSIUM ON USER INTERFACE SOFTWARE AND TECHNOLOGY (UIST), 8., 1995, USA, *Proceedings...* New York: ACM Press, 1995. p. 29-36.
- [63] RYOKAI, K.; MARTI, S.; ISHII, H. I/O Brush: Drawing with Everyday Objects as Ink. In: CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS (CHI), 2004, Austria, *Proceedings...* New York: ACM Press, 2004. p. 303-310.
- [64] SINCLAIR, P.; MARTINEZ, K. Tangible Hypermedia Using the ARToolKit. In: INTERNATIONAL AUGMENTED REALITY TOOLKIT WORKSHOP, 1., 2002, Germany, *Proceedings...* 2002.

Judith Kelner

Nacha Costa Bastos