

Capítulo

7

Interação com ARToolkit

Wender A. Silva¹, Edgard Lamounier Júnior², Alexandre Cardoso² e
Marcos Wagner Ribeiro³

¹Universidade Federal de Roraima/Universidade Aberta do Brasil

²Universidade Federal de Uberlândia/Faculdade de Engenharia Elétrica

³Universidade Federal de Goiás/Faculdade de Ciência da Computação

Abstract

This chapter presents techniques and technologies needed to realize the implementation of interaction in virtual environments Augmented Reality based ArtoolKit. Thus, we describe an Augmented Reality interface that enables the User interaction with the virtual environment through interactive menus, keyboard and markers.

Resumo

Este capítulo apresenta técnicas e tecnologias necessárias para realizar a implementação de interação em ambientes virtuais de Realidade Aumentada baseadas em ARToolkit. Desta forma, descreve-se uma interface de Realidade Aumentada que possibilita a interação do usuário com o ambiente virtual por meio de menus de interação, teclado e marcadores.

7.1. Introdução

Neste Capítulo destaca-se a interação em ambientes de Realidade Aumentada baseados no ARToolKit. A motivação para o desenvolvimento deste é a necessidade de pesquisas que visem o desenvolvimento de técnicas para implementação de interações no ARToolKit, e ainda, proporcionar uma boa fonte de consulta para os pesquisadores iniciantes na área de Realidade Aumentada. Assim, este texto tem por objetivo apresentar uma abordagem computacional/algóritmica para implementação de uma interface de Realidade Aumentada que possua um nível razoável de interação.

Espera-se com este Capítulo, contribuir para com os novos pesquisadores da área de Realidade Aumentada, oferecendo a descrição de uma abordagem (arquitetura) que possibilite a criação de uma interface baseada no ARToolKit, que, por meio de algoritmos pode ser criada interações do usuário com os objetos virtuais em cena. Essas interações podem ser realizadas por meio de transformações geométricas que podem desencadear colisões, oclusões e, ainda a troca de objetos virtuais na cena de Realidade Aumentada, proporcionando um nível satisfatório de interações.

7.2. Especificando a Framework ARToolKit

A biblioteca ARToolKit possui funções predefinidas que o programador deve chamar numa ordem específica para desenvolver programas de Realidade Aumentada. As diferentes partes do Toolkit também podem ser usadas separadamente. O ARToolKit suporta múltiplas plataformas, tentando minimizar as dependências entre as bibliotecas sem sacrificar a eficiência. Desta forma, o ARToolKit usa OpenGL¹ para a renderização, Glut para o controle de janelas e eventos, bibliotecas de vídeo e a API padrão de cada plataforma. A figura 7.1 ilustra a arquitetura do Framework de ARToolKit.

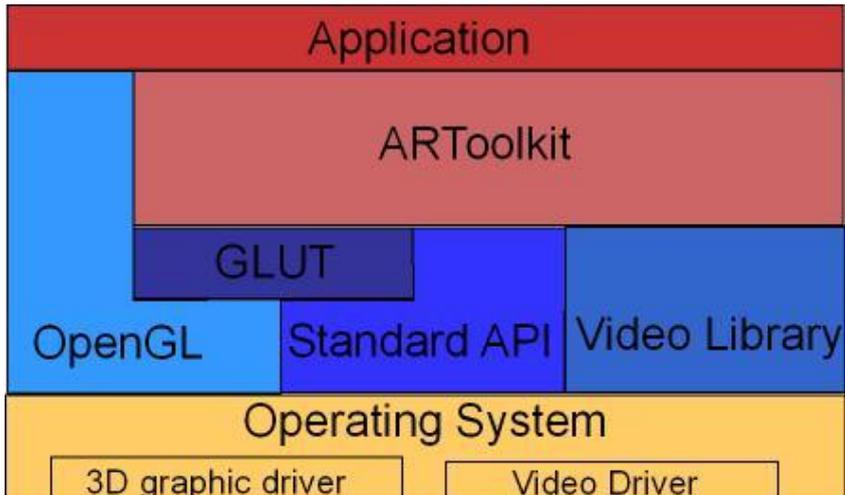


Figura 7.1 ARToolKit Framework

A biblioteca ARToolKit consiste de 3 pacotes:

- **libAR.lib**: biblioteca para rastreamento de marcadores, calibração e entrada de parâmetros. Contém as funções de suporte para a detecção dos marcadores.
- **libARvideo.lib**: biblioteca para capturar quadros da entrada de vídeo. Estas bibliotecas de vídeo variam a cada versão e, dependendo do suporte oferecido, disponibiliza bibliotecas para Windows WDM, Linux Vídeo 4 Linux, para câmeras Firewire ou então para placas de aquisição.
- **libARgsub.lib**: contém rotinas gráficas, baseadas nas bibliotecas OpenGL e Glut, para mapeamento do vídeo como textura em um ambiente 3D, bem como para o posicionamento dos objetos virtuais sobre o marcador.

A partir da versão 2.43 foram incluídas as bibliotecas de grafo de cena VRMLⁱⁱ, LibVRML97. Esta biblioteca de grafo de cena inclui outras 4 bibliotecas:

- **libARvrml.lib**: esta biblioteca implementa um visualizador VRML sobre o OpenGL. Na verdade ela implementa um visualizador utilizando as funções da biblioteca **LibVRML97** que atuam diretamente sobre o OpenGL. Estende ao ARToolkit a classe **OpenGLViewer** fornecida pela biblioteca **libvrml97gl**.
- **libvrml97core.lib**: é a implementação das funções VRML97. Contém todos os comandos para gerar uma cena a partir de um código VRML e, claro, o seu analisador léxico e sintático (*parser*).
- **libvrml97js.lib**: é a implementação do *script* (**javascript**) versão 1.1 do *engine* Mozilla. Estes *scripts* permitem gerar animações 3D, porém nem todos os aspectos são considerados.
- **libvrml97gl.lib**: é uma implementação de uma classe **OpenGLViewer** que renderiza as cenas VRML sobre uma janela OpenGL.

7.3. Compilando o ARToolkit

Para compilar um projeto em ARToolkit, é necessária a adição das bibliotecas Glut e OpenGL, as quais dão suporte aos objetos gráficos e outras aplicações ligadas ao teclado e ao mouse, caso sejam utilizadas, e a algoritmos de processamento de imagens, como thresholding e gradientes de visualização. Para compilar um projeto no Visual Studio 2005, basta incluir no arquivo glut.h no código fonte do projeto, e adicionar a biblioteca estática glut32.lib como additional library a ser linkada. A biblioteca dinâmica glut32.dll deve ser colocada na raiz do projeto. O OpenGL já compõe o pacote de bibliotecas do visual studio 2005, portanto não precisa ser adicionando explicitamente no projeto. Caso a sua aplicação utilize objetos 3D do tipo openVRML, deve-se adicionar as bibliotecas adicionais, openVRML.lib, libpng.lib, lipjpeg.lib, zlib.lib, artln.lib além da biblioteca libarvrml.lib e das demais bibliotecas do ArtoolKit.

7.4. Módulo de interação proposto para o ARToolkit

De acordo com a figura 7.2, ilustra-se o funcionamento de um Pipeline de funcionamento do algoritmo apresentado neste Capítulo e adaptado à arquitetura do ARToolkit, onde pode-se relatar:

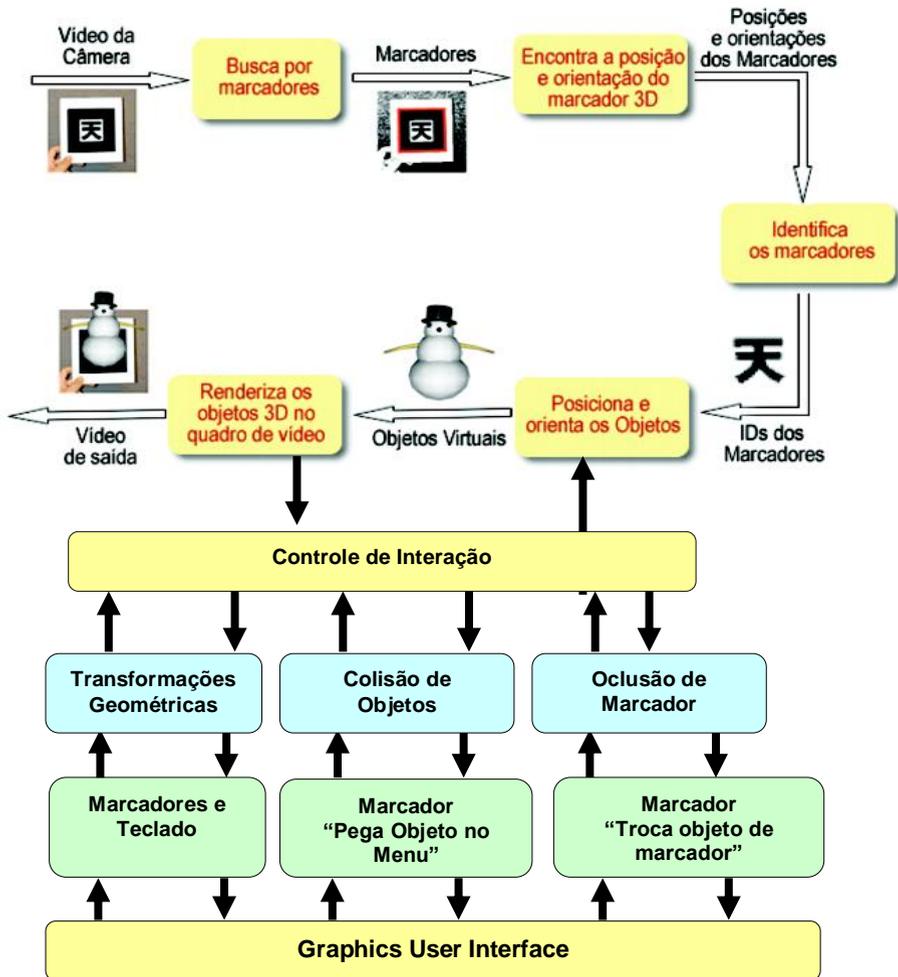


Figura 7.2. Pipeline da Interface R.A adaptada à Arquitetura do ARToolkit

- **Controle de Interação:** O controle realizado nas interações da Interface de Realidade Aumentada é realizado por meio de várias técnicas para criar interações que permitem obter um ambiente com um nível satisfatório de interatividade. Assim, pode-se destacar as interações na interface do ARToolKit: a) Transformações Geométricas: o ARToolKit a realiza em tempo de execução, todas as transformações geométricas possíveis. b) Colisão de Objetos Virtuais: o ARToolKit realiza a colisão de objetos virtuais, esses objetos estão dispostos na interface em forma de menus suspensos, onde o usuário ao colidir o marcador em qualquer um dos objetos disponíveis, faz a sua captura, ou seja, o objeto virtual que está sendo visualizado na tela é instanciado para o marcador que realizou a colisão. c) Oclusão de Marcadores: Ao realizar o item “b” deste parágrafo, o objeto virtual com seu respectivo marcador ao ser posicionado à frente de um outro marcador, onde o sistema de Realidade Aumentada verifica se o marcador que esta sendo sobreposto está ocluído ou não, caso a oclusão tenha ocorrido, o marcador que foi ocluído recebe o objeto virtual do marcador que realizou a oclusão.

- **Graphics User Interface (GUI):** Este bloco permite ao usuário visualizar graficamente, de modo interativo e em tempo real, a entrada de dados e a saída de informações. A GUI exibe então o cenário, apresentando os ambientes virtuais em Realidade Aumentada em uma janela gráfica por meio de dispositivo de captura de vídeo e monitor. Essa interface gráfica com o usuário permite realizar interações em tempo real. Essas interações ocorrem por meio das transformações geométricas, troca de objetos virtuais na cena, colisão de objetos virtuais e oclusão de marcadores.

7.5. Implementação da Interface de Realidade Aumentada

A interface de Realidade Aumentada foi implementada, como já visto anteriormente, utilizando-se a base do ARToolKit (KATO & BILLINGHURST, 2005). Desta forma, o código do ARToolKit

alterado foi o *simpleVRML.cpp* que fica na pasta *examples*. Este código foi escolhido devido ao fato de trabalhar com objetos em formato VRML.

7.5.1 Implementação das Interações via teclado

Para a implementação das interações via teclado, utilizou-se a função *keyEvent* implementada no ARToolkit. Por padrão esta função vem apenas com o evento da tecla “escape” para fechar a aplicação. As interações na interface de Realidade Aumentada do sistema de Realidade Aumentada baseiam-se nas transformações geométricas: escala, translação e rotação. Essas transformações ocorrem em tempo real de execução. Assim, destacam-se as interações implementadas: redução e aumento da escala dos objetos, translação de objetos para todos os eixos, rotação dos objetos para todos os eixos e uma função de retorno ao ponto inicial de *default* do aplicativo. Desta forma, descreve-se em pseudocódigo a implementação para redução de escala.

Tecla de subtração (teclado numérico) - Reduz a escala do objeto selecionado no marcador em 0.2 pixel a cada toque no teclado.

```
se ( tecla == 0x2d) {  
    px = px - 0.2  
    py = py - 0.2  
    pz = pz - 0.2 }
```

7.5.2. Implementação dos Menus da Interface de Realidade Aumentada

Para a implementação das interações realizadas por meio de marcadores tornou-se necessário a criação de menus na interface de Realidade Aumentada, esses menus de interações ficam visíveis na cena gráfica da aplicação sem a utilização de marcadores. A Figura 7.3 ilustra os objetos virtuais e seus respectivos menus.

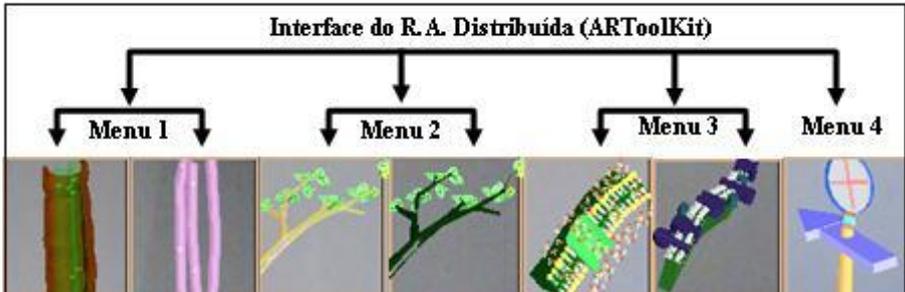


Figura 7.3 Esquema dos menus interativos e seus objetos virtuais

Os menus interativos de um a três podem conter n objetos, porém para este texto, optou-se por dois objetos por menu. Já o menu interativo de número quatro funciona como um painel de controle para a realização das transformações geométricas por meio de marcadores. Desta forma, dentro da função *mainLoop* na implementação do ARToolKit “pede-se” para visualizar os objetos na cena gráfica do vídeo. Para proporcionar a opção de dois objetos virtuais por menu, trabalha-se com estruturas de condição. É importante ressaltar que todos os objetos virtuais relacionados devem estar previamente cadastrados. Desta forma, ao incorporar os objetos virtuais, o ARToolKit por meio das bibliotecas *libvrm197gl.lib* e *libARvrm1.lib* cria na função *arVrm197Draw* uma lista de objetos virtuais do tipo VRML e, assim por meio desta lista de objetos virtuais pode-se criar várias possibilidades de interatividade. A figura 7.4 ilustra parte do pseudocódigo da implementação dos menus interativos.

```
// Primeiro Menu
Se (variável == 0) {
    objeto_vrml(visualiza objeto 1);
}
Senão se (variável == 1) {
    objeto_vrml(visualiza objeto 2);
}
...
```

Figura 7.4 Parte da implementação dos menus interativos

Para a disposição dos menus interativos na cena gráfica do ARToolkit foi necessário mapear toda a tela gráfica gerada pela biblioteca *libARgsub.lib*, para isso foi realizado vários testes durante o desenvolvimento da aplicação de Realidade Aumentada.

7.5.3 Implementação da opção de troca de objeto virtual por meio do menu suspenso na Interface de Realidade Aumentada (R.A)

Para a implementação da opção de troca de objeto virtual por menu, necessitou-se utilizar as propriedades de janelas da biblioteca *glut32.lib* para criação de um menu suspenso. (menu do Windows). Assim, a figura 7.5 ilustra o menu suspenso.

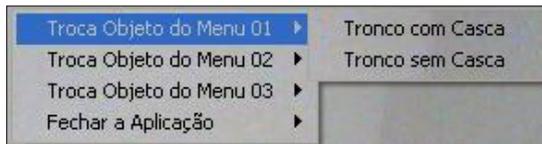


Figura 7.5. Menu suspenso na Interface de R.A

O menu suspenso possibilita a troca de objetos virtuais na cena gráfica, desta forma utiliza-se estruturas de condição para verificar qual objeto virtual será visualizado na cena. A figura 7.6 ilustra parte do pseudocódigo do gerenciamento do menu suspenso.

<pre>//Gerenciamento do Menu 01 função Menu01(opção) { Escolha (opção) { caso 0: variável = 1; pare; caso 1: variável = 0; pare; } } ...</pre>	<pre>//Gerenciamento do Menu 02 função Menu02(opção) { Escolha (opção) { caso 0: variável = 0; pare; caso 1: variável = 1; pare; } } ...</pre>
--	--

Figura 7.6. Parte do pseudocódigo do menu suspenso na interface de Realidade Aumentada

Ainda, na utilização do menu suspenso é necessário criar uma função para gerenciar o mouse. Esta função verifica se o botão direito do mouse foi pressionado ou não. Caso o botão direito do mouse tenha sido pressionado, a função *gerenciamouse* chama a função *criamenu*, conforme pode ser observado na figura 7.7.

```
função gerenciamouse( Qual botão foi pressionado?) {  
    se (botão == Botão_Direito)  
        se (estado == Foi_Pressionado)  
            CriaMenu();  
} ...
```

Figura 7.7 Parte do pseudocódigo da função *gerenciamouse*

7.5.4. Implementação das Interações por meio de Marcadores

A implementação das interações por meio de marcadores segue as mesmas características do item “*Implementação das Interações via teclado*”, porém é feita por meio da colisão de objetos virtuais com o *menu de interação quatro* (Menu 04). Assim, neste sentido, tem-se um marcador para cada transformada geométrica.

A colisão trabalhada aqui se trata de uma colisão por aproximação, onde verifica-se a posição do objeto virtual visualizado na cena gráfica, e ao aproximar-se de um valor determinado efetua-se uma função qualquer. A cena gráfica o ARToolKit foi mapeada, então, quando tem-se um objeto virtual que, neste caso fica invisível, atrelado a um marcador que estiver entre dois valores no eixo “X” e dois valores no eixo “Y”, tem-se efetuada uma colisão aproximada.

Assim, entende-se que, ao verificar tal condição, pode-se implementar qualquer tipo de ação e, neste caso, foi implementado o mesmo código do item “*Implementação das Interações via teclado*”. Neste sentido ilustra-se apenas parte do pseudocódigo da colisão aproximada, conforme demonstrado na figura 7.8.

```
//Colisão aproximada para realizar transformações geométricas com
Marcadores
se (marcaA > 130 E marcaA < 220 E marcaA < 190 E marcaA > 130 )
    {
        funçãoaumentaescala();
    }
senão
se (marcaR > 130 E marcaR < 220 E marcaR < 190 E marcaR > 130 )
    {
        funçãoreduzescala();
    }
senão
se (marcaD > 130 E marcaD < 220 E marcaD < 190 E marcaD > 130 )
    {
        funçãotransladadireita();
    }
    funçãotransladabaixo();
}
senão
se (marcaZM > 130 E marcaZM < 220 E marcaZM < 190 E marcaZM >
130 )
    {
        funcaotransladazmais();
    }
}
senão ...
```

Figura 7.8. Parte da implementação das transformações geométricas por meio de colisão aproximada

7.5.5. Interação com os Menus da Interface de Realidade Aumentada

Para interação com os menus da interface de Realidade Aumentada, como já descrito no item anterior “*Implementação das Interações por meio de Marcadores*” neste capítulo, utiliza-se colisão aproximada de objetos virtuais. Assim, trabalha-se com apenas um marcador para

interagir com todos os três menus. Este marcador faz a interação apenas com os três primeiros menus de cima para baixo na interface de Realidade Aumentada. O marcador em formato de uma “*pá*” ao fazer a colisão com algum objeto virtual que esteja entre dois valores no eixo “*X*” e dois valores no eixo “*Y*”, faz a captura do objeto virtual, ou seja, cria uma instância do objeto que está sendo visualizado na cena gráfica no marcador “*pá*” que realizou a colisão.

É importante ressaltar que existem apenas três menus de interação, porém são seis objetos virtuais para serem visualizados, ou seja, são seis objetos virtuais que podem sofrer colisão do marcador “*pá*”. Neste aspecto, utilizou-se de estruturas de condição para verificar qual objeto virtual está sendo visualizado no momento da colisão do marcador “*pá*”. Este assunto já foi comentado no item “*Implementação dos Menus da Interface de Realidade Aumentada*” deste capítulo. Desta forma, a figura 7.9 ilustra o pseudocódigo da implementação.

```
//Verifica o objeto que deverá ser instanciado no marcador “pá”
se (variável == 0) {
se (marca0 >130 E marca0 < 220 E marca0 < -50 E marca0 > -110 )
{
    marca0 = objetovirtualdomenu1;
    objetoatual = '1';
} senão
se (variável == 1) {
se (marca0 >130 E marca0 < 220 E marca0 < -50 E marca0 > -110 )
{
    marca0 = objetovirtualdomenu1;
    objetoatual = '2';
} senão ...
```

Figura 7.9 Parte da implementação das interações por colisão com os menus interativos

7.6. Funcionamento do Sistema

A aplicação de interface de R.A. possui três menus visualizados na cena gráfica (sem a utilização de marcadores), cada um desses menus pode, nesta aplicação, possuir dois objetos virtuais. A figura 7.10 ilustra a interface de Realidade Aumentada.

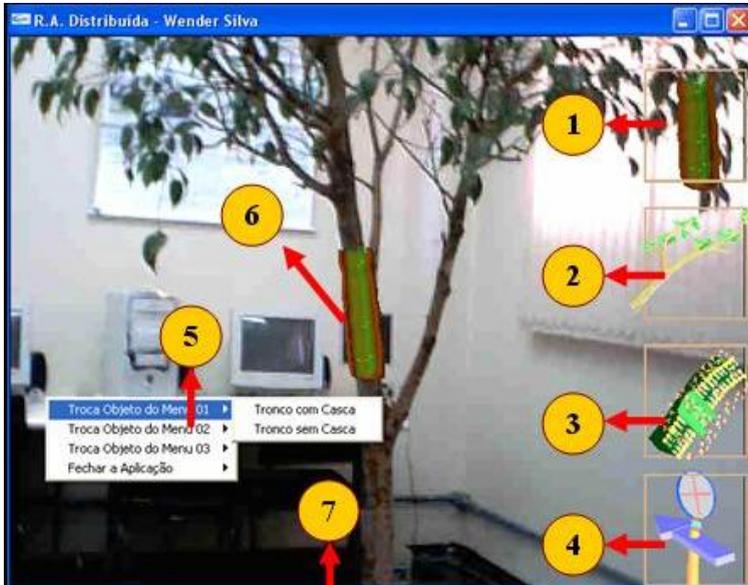


Figura 7.10. Aplicação de Interface de R.A

Onde: **(1)** Primeiro Menu: possui dois objetos virtuais, um tronco cortado com vasos de transporte de solutos no seu interior e, um objeto ilustrando apenas os vasos de transporte; **(2)** Segundo Menu: possui dois objetos virtuais, um galho transparente com os vasos de transporte de solutos no seu interior e, um galho sem os vasos e também sem transparência; **(3)** Terceiro Menu: possui um estroma com detalhamento do processo da fotossíntese e da evaporação de água no processo do transporte de solutos e, um estroma sem detalhamento; **(4)** Quarto Menu: possui interatividade por meio das transformações geométricas, trabalhando com colisão de marcadores.

Para tanto, têm-se um marcador para casa tipo de transformada geométrica; (5) Menu Suspenso: possui as rotinas para trocar o objeto virtual de qualquer um dos três menus visualizados na cena gráfica em tempo real; (6) Objeto virtual tronco cortado sobreposto sobre a árvore real: esse tronco cortado ilustra o transporte de seiva (solutos) no interior de uma árvore. Esse transporte de solutos é baseado em um estudo de caso baseado em fisiologia vegetal, apenas para ilustrar os objetos virtuais na cena; (7) Árvore real para ilustrar o protótipo de Realidade Aumentada.

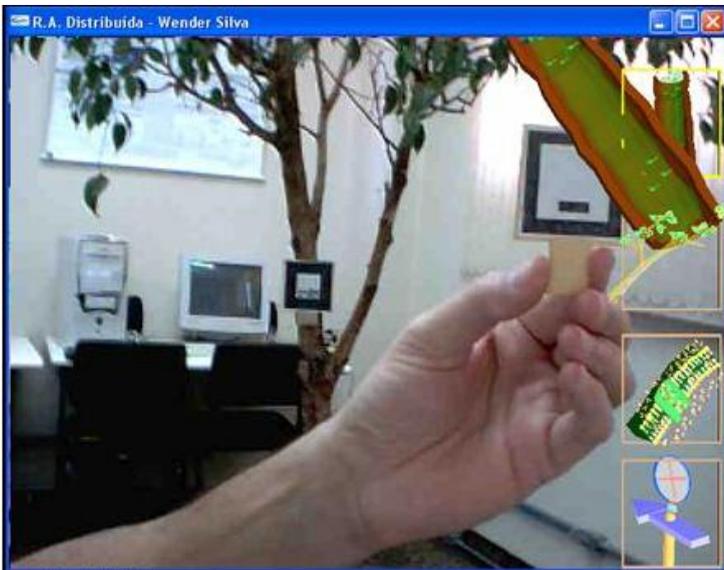


Figura 7.11. “pegando” objeto virtual no menu

A interface de Realidade Aumentada também permite a realização de transformações geométricas dos objetos virtuais selecionados na cena gráfica. As transformadas geométricas podem ser realizadas por meio de teclado, ou ainda por meio do menu interativo utilizando-se para isso marcadores previamente cadastrados. Para a utilização do menu interativo, deve-se possuir um marcador para cada transformada geométrica.

7.7. Considerações Finais e Tendências Futuras

Durante a pesquisa para a construção deste Capítulo, constatou-se que existe pouco material que trata da interatividade em ambientes de Realidade Aumentada e ainda, constatou-se que um dos maiores entraves para a pesquisa e construção de ambientes de Realidade Aumentada baseados em ARToolKit é a dificuldade de configuração, compilação e exemplos de implementação.

Desta forma, este Capítulo apresentou uma proposta de Pipeline de Interação para o ARToolKit e ainda detalhou textualmente os passos para a criação de menus interativos e as técnicas e tecnologias para desenvolvê-los.

A interface de Realidade Aumentada implementada proporciona um bom nível de interatividade, porém, para implementação de Interfaces de Realidade Aumentada pode-se também utilizar o OSGART ou FLARToolKit.

Assim, entende-se que a principal contribuição deste Capítulo é um Pipeline de interações da interface de Realidade Aumentada. Assim, pode-se proporcionar um ambiente virtual de Realidade Aumentada com um bom nível de interatividade.

Referências

- Azuma, R, et al. “Recent Advances in Augmented Reality. IEEE Computer Graphics and Applications, november/december 2001, Vol. 21, P. 34-37.
- Azuma, R. T.. “A Survey of Augmented Reality”, Presence: Teleoperators and Virtual Environments, Vol. 6, P. 355- 385. 1997
- Cardoso, Alexandre, et al. “Conceitos de Realidade Virtual e Aumentada”. In: Tecnologias para o desenvolvimento de sistemas de Realidade Virtual e Aumentada. Recife: ed. Universitária da UFPE, 2007.

Kato, H.; Billinghurst, M. et. al. “Artoolkit Version 2.52: A Software Library for Augmented Reality Applications”. HITLAB. Junho de 2005. Disponível em: <<http://www.hitl.washington.edu/>>. Acesso em: 10 janeiro de 2010.

Kirner, C. ; Tori, R. “Introdução à Realidade Virtual, Realidade Misturada e Hiper-Realidade”. In: Cláudio Kirner; Romero Tori. (ed.). Realidade Virtual: conceitos, tecnologia e tendências. 1ª ed. São Paulo, 2004, V. 1, P. 3-20.

Kirner, Cláudio. “ARToolKit versão 2.52”. Realidade Virtual e Aumentada, Abril de 2008. Disponível em: <<http://www.realidadevirtual.com.br>>. Acesso: maio 2008.

Milgram, P. et. al. “Augmented Reality: a class of displays on the Reality-Virtuality Continuum”. Telemanipulator and Telepresence Technologies, SPIE, V.2351, 1994.

Tori, Romero; Kirner, Cláudio; Siscoutto, Robson. “Fundamentos e Tecnologia de Realidade Virtual e Aumentada”. Porto Alegre: SBC, 2006.

Zhou, Zhiying; et al. “Interactive Entertainment Systems Using Tangible Cubes”. IEEE Computer Society. Virtual Reality, 2004.

ⁱ OpenGL é uma biblioteca de rotinas gráficas e de modelagem, bi (2D) e tridimensional (3D), extremamente portátil e rápida. Para maiores informações sobre OpenGL e suas rotinas gráficas acesse o endereço: <http://www.inf.pucrs.br/~pinho/CG/Aulas/OpenGL/OpenGL.html>

ⁱⁱ VRML (Virtual Reality Modeling Language) linguagem de programação para criação de Ambientes Tridimensionais voltados para a internet. Para maiores informações sobre o VRML acesse o endereço <http://www.realidadevirtual.com.br>