

## Capítulo

# 5

## ARToolKit: Biblioteca para Desenvolvimento de Aplicações de Realidade Aumentada

Rafael Santin e Claudio Kirner

Curso de Mestrado em Ciência da Computação  
Universidade Metodista de Piracicaba - UNIMEP  
Rodovia do Açúcar, Km 156, 13400-911 Piracicaba - SP, Brasil  
{rafael, ckirner}@unimep.br

### *Abstract*

*ARToolKit is a free and open library used in the development of augmented reality applications. The augmented environment is obtained by mixing video with registered virtual objects, showed on head mounted display or monitor. This chapter presents the concepts, structure and behavior of ARToolKit, as well as configuration examples and programming of functions.*

### *Resumo*

*ARToolKit é uma biblioteca gratuita e aberta usada para o desenvolvimento de aplicações de realidade aumentada. O ambiente aumentado é obtido pela mistura de vídeo com objetos virtuais, sendo mostrado em capacete ou monitor. Este capítulo apresenta conceitos, estrutura e funcionamento do ARToolKit e exemplos de configuração e programação de funcionalidades.*

### **6.1. Realidade Aumentada e ARToolKit**

A Realidade Aumentada (RA) consiste numa interface avançada de computador, que promove em tempo real a exibição de elementos virtuais sobre a visualização de determinadas cenas do mundo real, oferecendo um forte potencial a aplicações industriais e educacionais, devido o alto grau de interatividade [Kirner e Tori 2004]. Para desenvolver aplicações de RA, são utilizadas bibliotecas computacionais que implementam a captura de vídeo, técnicas de rastreamento, interação em tempo real e os ajustes visuais das cenas do mundo real e virtual.

O ARToolKit é uma biblioteca de programação multi-plataforma considerada um kit de ferramenta de RA, bastante utilizada e discutida por desenvolvedores e pesquisadores da comunidade de Realidade Aumentada. O ARToolKit possui o seu código livre para modificações e uso no desenvolvimento de aplicações não comerciais sob licença GPL [GNU 2007], enquanto que

a versão proprietária para a comercialização é oferecida pela incorporação ARToolworks [ARToolworks 2007].

### **6.1.1. Fundamentos do ARToolKit**

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

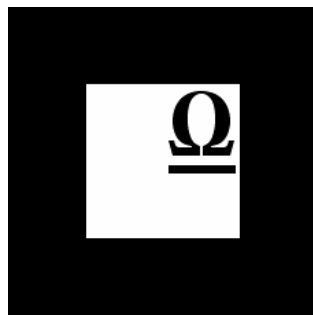
A saída de aplicações desenvolvidas com ARToolKit pode ser visualizada através de dispositivos de visão indireta, não imersivos, ou visão direta, imersivos. A diferença de imersão entre as duas formas de visão, está relacionada à autenticidade impelida ao sentido visual do usuário. Os dispositivos de visão indireta, como os monitores, não imersivos, promovem a visualização conjugada das cenas virtual e real fora do espaço alvo de atuação. Já a visualização direta, imersiva, pode ser obtida por meio dos dispositivos como o HMD (Head Mounted Display), que fornecem a visualização combinada das cenas diretamente no espaço alvo de atuação da aplicação [Kirner e Tori 2006].

Os objetos virtuais visualizados em aplicações desenvolvidas com as distribuições do ARToolKit podem ser implementados com OpenGL e/ou com VRML. A visualização desses objetos virtuais é realizada no momento da inserção de seus respectivos marcadores no campo de captura da câmera de vídeo.

#### **6.1.1.1 Marcadores**

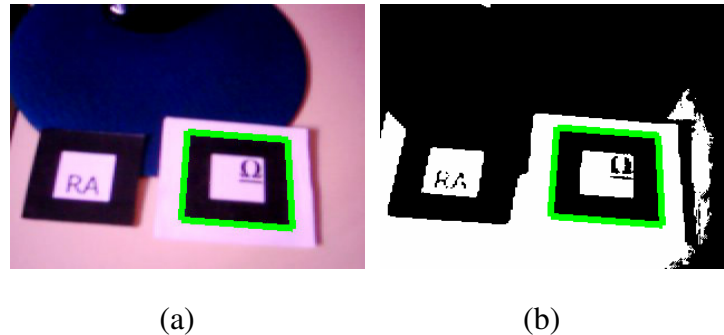
O rastreamento implementado no ARToolKit estima a pose de marcadores, tornando possível desenvolver aplicações que necessitem conhecer a posição e orientação de elementos ou ações reais, que são representados na cena por marcadores. Como exemplo, tem-se as aplicações de RA, que utilizam o marcador para posicionar e orientar elementos virtuais na cena do mundo real, tornando um meio de interação do usuário com essas aplicações.

Os marcadores reconhecidos pelo ARToolKit consistem em figuras geométricas quadradas, que contém no seu interior símbolos para identificá-los. A figura 6.1 mostra um exemplo de marcador, com símbolos para a sua identificação.



**Figura 6.1. Exemplo de marcador.**

Como o ARToolKit extrai da imagem de vídeo limiarizada (em preto e branco) as bordas do quadrado em preto, utiliza-se uma moldura em branco antecedendo esse quadrado para promover o contraste no próprio marcador, viabilizando o seu reconhecimento sobre superfícies de cores escuras. A Figura 6.2a demonstra dois marcadores dispostos sobre uma superfície escura. A diferença entre os marcadores consiste no fato do marcador com o símbolo RA não possuir a moldura em branco e outro conter essa moldura. O marcador com o símbolo RA não pode ser identificado sobre a superfície escura, pois não é possível extrair as suas bordas na imagem limiarizada (Figura 6.2b).

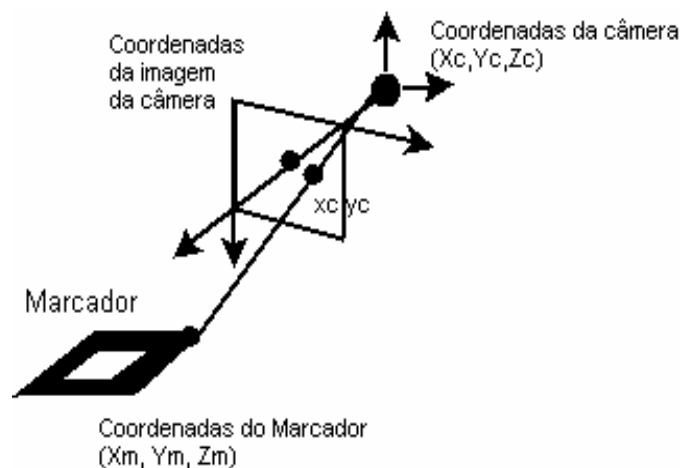


**Figura 6.2 Utilidade da moldura branca do marcador**

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

#### 6.1.1.2 Rastreamento.

O rastreamento no ARToolKit é responsável pelo processamento da imagem, que extrai algumas informações com relação a detecção, e pela identificação de características dos marcadores, além de estimar sua posição e orientação. Nesse caso, a obtenção da posição e orientação do marcador é realizada através da análise da imagem de vídeo, que estabelece o relacionamento entre as coordenadas do marcador e as coordenadas da câmera, como demonstrado na figura 6.3 [Kato et al 1999].



**Figura 6.3. Relacionamento entre os sistemas de coordenadas do marcador e da câmera.**

O relacionamento entre as coordenadas do marcador e as coordenadas da câmera é realizado por intermédio de uma matriz 3x4, denominada “matriz transformação”. A Figura 6.4 mostra a multiplicação de uma matriz transformação "T" por um ponto 3D no marcador ( $X_m, Y_m, Z_m$ ), obtendo o ponto correspondente no sistema de coordenadas da câmera ( $X_c, Y_c, Z_c$ )

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_1 \\ R_{21} & R_{22} & R_{23} & T_2 \\ R_{31} & R_{32} & R_{33} & T_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix} = \mathbf{T} \begin{bmatrix} X_m \\ Y_m \\ Z_m \\ 1 \end{bmatrix}$$

**Figura 6.4. Multiplicação de matrizes**

Para estimar a posição e orientação do marcador, através da análise da imagem de câmera, torna-se necessário utilizar os parâmetros de câmera, a fim de corrigir as distorções inerentes a câmera [Lepetit Fua 2005]. Assim, é possível estimar, com certa precisão, o relacionamento entre as coordenadas 3D do mundo e as coordenadas 2D da imagem [Abdullah e Martinez 2002]. O ARToolKit utiliza esses parâmetros de câmera no cálculo da matriz transformação [Kato Billinghamurst 1999].

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

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

### 6.1.1.3 Funcionamento do ARToolKit

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

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

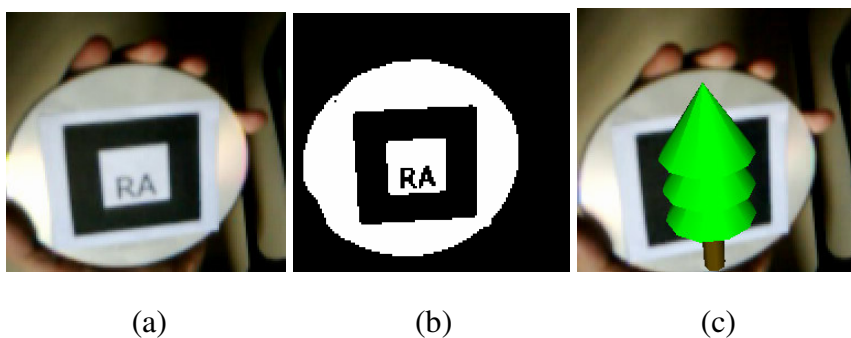
Os passos de número 2 a 5 são repetidos continuamente até a aplicação ser finalizada, enquanto os passos 1 e 6 fazem respectivamente a inicialização e o término da aplicação [Kato et al 2000].

O ARToolKit é distribuído com aplicações exemplos, que implementam os passos citados anteriormente, servindo de modelo aos programadores, tanto para o conhecimento de funções da biblioteca, quanto para o auxílio no desenvolvimento de novas aplicações de RA. A tabela 6.1 mostra os passos e as respectivas funções executadas na aplicação "simple" disponibilizada nas distribuições do ARToolKit.

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

**Tabela 6.1 Passos e as funções implementadas num exemplo de aplicação distribuída com o ARToolKit .**

O funcionamento de aplicações de RA, como alguns dos exemplos incluídos na distribuição do ARToolKit, executa várias etapas relacionadas as ilustrações da Figura 6.5. Na primeira etapa, a imagem de vídeo capturada, conforme a figura 6.5a, é convertida em uma imagem binária (preta e branca), de acordo com o valor do limiar ou *threshold*, resultando numa imagem como a da figura 6.5b. Por conseguinte, as regiões quadradas, nessa imagem binária, são detectadas e comparadas com gabaritos de marcadores cadastrados no sistema pelo usuário. Caso haja a identidade entre supostos marcadores e os marcadores conhecidos pelo sistema, a aplicação considera que encontrou um marcador na imagem. A próxima etapa, então, consiste na obtenção da posição e orientação de marcadores [Kato et al 2000]. Assim, é possível desenhar o objeto virtual sobreposto a seu respectivo marcador, como mostra a figura 6.5c.



**Figura 6.5 Etapas envolvidas no funcionamento de uma aplicação de RA : a) imagem da cena com o marcador, b) a imagem limiarizada e c) objeto virtual sobrepondo o marcador.**

A Figura 6.6 mostra um diagrama, detalhando as principais etapas realizadas no funcionamento da aplicação.

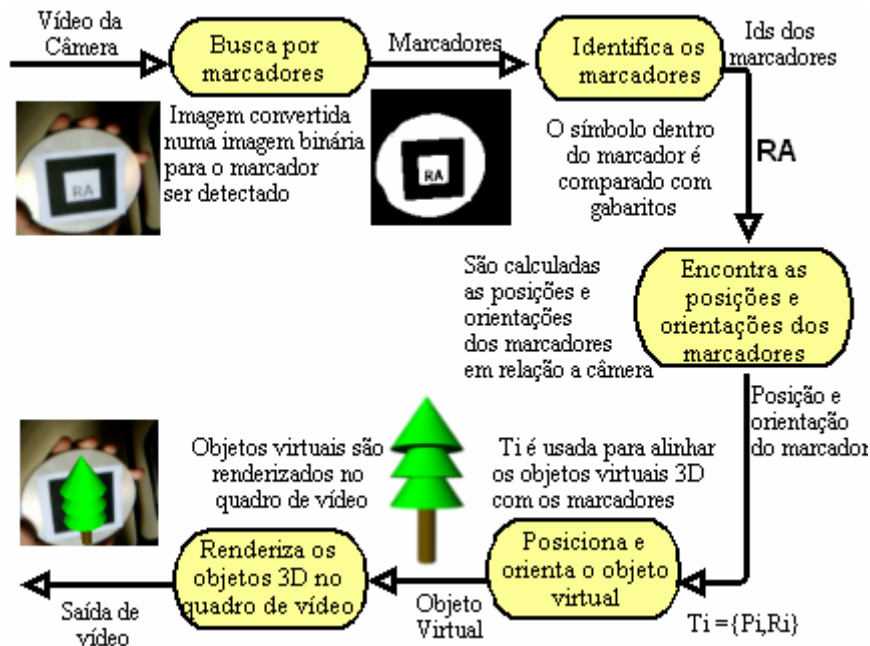


Figura 6.6. Funcionamento de uma aplicação de RA

## 6.2. Instalação e configuração do ARToolKit

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

### 6.2.1. Instalação

Para a instalação do ARToolKit deve-se, primeiramente, baixar a versão desejada. O próximo passo é descompactar o arquivo no local de conveniência. Esse local será referenciado abaixo como {ARToolKit}. A Figura 6.7 mostra a estrutura de diretórios da versão 2.72.1 do ARToolKit, após descompactação.

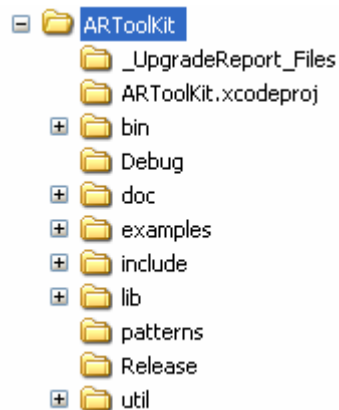


Figura 6.7. Estrutura do diretório da versão 2.72.1 do ARToolKit após a instalação.

### 6.2.2. Configurações na plataforma Windows

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

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

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

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

1. Instalar um dos compiladores da Microsoft (Visual Studio .NET 2003 ou Visual Studio 6) ou o CygWin.
2. Baixar e descompactar a biblioteca “DSVideoLib-0.0.8b-win32” [DSVideoLib 2007] dentro do diretório raiz “{ARToolKit}”, certificando-se que o diretório seja denominado “DSVL”. Copiar os arquivos “DSVL.dll” e “DSVLd.dll” para a pasta “bin” do ARToolKit (“{ARToolKit}\bin”).
3. Baixar o GLUT [GLUT 2007], descompactar e copiar o arquivo “glut32.dll” para pasta “system” do Windows, copiar o arquivo “glut32.lib” para a pasta “lib” do ARToolKit (“{ARToolKit}\lib”) e criar uma pasta “GL” no diretório “include” do ARToolKit (“{ARToolKit}\include\GL”), copiando para essa nova pasta o arquivo “glut.h”.
4. Executar o script “Configure.win32.bat” que está em “{ARToolKit}\Configure.win32.bat” para criar o arquivo “{ARToolKit}\include\AR\config.h”.
5. Abrir o arquivo “ARToolKit.sln” no VisualStudio .NET ou o arquivo “ARToolkit.dsw” no Visual Studio 6.
6. Compilar os projetos.
7. Executar as aplicações geradas em “{ARToolKit}\bin”

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

8. Baixar a biblioteca “OpenVRML” (OpenVRML-0.16.1-bin-win32 ou OpenVRML-0.14.3-win32 [OpenVRML 2007]) e descompactá-la no diretório raiz do ARToolKit.

9. Copiar o arquivo “js32.dll” da pasta “{ARToolKit} \OpenVRML\bin” para a pasta “{ARToolKit}\bin”.
10. Habilitar e compilar, no VisualStudio .NET, os projetos “libARvrm1” e “simpleVRML”. Já, no VisualStudio 6, é necessário criar um projeto para compilar a biblioteca estática gerada pelo “libARvrm1” e um projeto de aplicação win32, com as devidas dependências ajustadas para gerar o executável “simpleVRML.exe”.

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

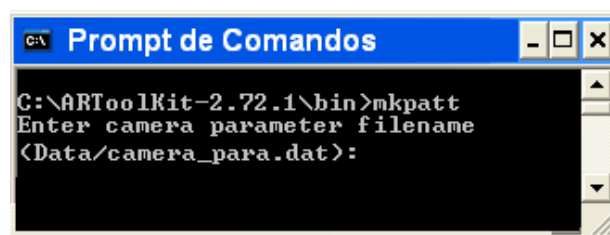
### 6.3.Aplicações inclusas no ARToolKit

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

#### 6.3.1.Aplicações Utilitárias

As aplicações utilitárias são responsáveis pela configuração e teste do sistema. Os utilitários de configuração são o “mkpatt, calib\_camera2, calib\_cparam e calib\_distortion”. Já os utilitários de teste do sistema são o “graphicTest” e “videoTest”. Os códigos desses programas estão em “{ARToolKit}\util”.

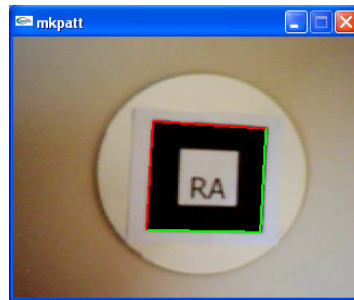
O “mkpatt” é um programa usado na geração dos arquivos bitmaps (mapa de bits), que relacionam os marcadores aos objetos virtuais. Cada arquivo bitmap contém um conjunto de exemplo de imagens do marcador. Esse conjunto é conhecido, como treinamento do marcador. Uma vez executado o “mkpatt”, será pedido para entrar com o arquivo contendo os parâmetros de câmera. Escreva o caminho para o arquivo, caso esse seja diferentes da configuração default (“Data/câmera\_para”), senão basta teclar “enter”. A Figura 6.8 demonstra o início do programa.



**Figura 6.8. Início da execução do mkpatt**

O programa exibirá uma janela com a imagem do vídeo. Enquadre o marcador nessa imagem, de modo a aparecer um retângulo com lados vermelhos, à esquerda e acima, e verdes, à direita e abaixo, nas bordas do marcador, como mostra Figura 6.9.





**Figura 6.9. Marcador identificado pelo “mkpatt”**

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

Os programas “calib\_camera2, calib\_cparam e calib\_distortion” são responsáveis pela calibração de câmera. As etapas relacionadas à calibração de câmera são detalhadas em [Consularo et al 2004].

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

### **6.3.2.Exemplo de aplicações de RA**

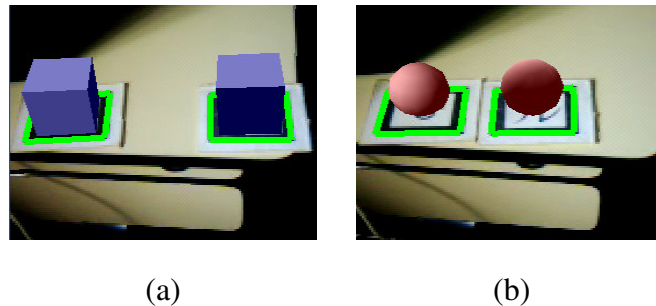
As aplicações de RA fornecidas junto ao ARToolKit são exemplos que não só viabilizam o entendimento no funcionamento das funções do ARToolKit, mas também servem como modelo para a produção de novas aplicações.

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

#### **6.3.2.1.Teste de Colisão**

A aplicação “collideTest” possui como característica principal a verificação de colisão entre dois marcadores. A função responsável em verificar a colisão é denominada “checkCollisions”. Essa função recebe as estruturas relacionadas às informações de cada um dos dois marcadores, além e um número representante do fator de colisão. O retorno dessa função é um número inteiro “1”, caso os marcadores esteja em colisão e “0”, caso contrário. A Figura 6.10 mostra a aplicação em execução, quando a distância entre os marcadores supera o limite do fator de colisão, sendo desenhado um cubo sobre os marcadores, conforme a Figura 6.10a. Quando a distância entre os

marcadores for inferior ao limite, será desenhada uma esfera sobre os marcadores, conforme a Figura 6.10b.

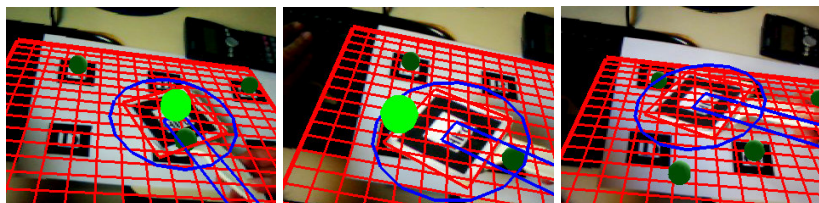


**Figura 6.10. Execução da aplicação “collideTest”: a) cubos desenhados sobre os marcadores distantes e b) esferas desenhadas sobre marcadores próximos.**

O exemplo “collide” utiliza os marcadores cadastrados no arquivo “object\_data2” contido em “{ARToolKit}\bin\Data”. Dessa forma, o usuário pode associar os seus próprios marcadores, utilizando para isso o “mk\_patt” na geração do arquivo bitmaps (mapa de bits) dos marcadores, os quais deverão substituir os marcadores configurados no “object\_data2”.

### 6.3.2.2. Uso da Pá

O “paddleDemo” (uso da pá) é um dos exemplos que insere um marcador com funcionalidades especializadas a técnicas tangíveis. Esse marcador será denominado pá. A pá é utilizada para interagir com objetos virtuais atrelados a um cartão contendo vários marcadores, denominado de cartão base. Essa aplicação disponibiliza técnicas que permitem identificar a inclinação da pá em relação ao cartão base e, assim, atribuir algumas ações, como no caso despejar uma esfera no cartão base ou a pegá-la. As funções que implementam essas técnicas são a “check\_incline” e a “check\_pickup”. Essas funções encontram-se no arquivo “command\_sub.c” em “{ARToolKit}\exemplo\paddleDemo”. A Figura 6.11 mostra a sequência de execução do “paddleDemo”.



**Figura 6.11. Execução do exemplo “paddleDemo”, mostrando a esfera sobre a pá, a inclinação para despejar a esfera na base e a esfera fixada na base.**

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

### 6.3.2.3. SimpleVRML

O “simpleVRML” é um programa que possibilita visualizar objetos virtuais, escritos na linguagem VRML, sobrepostos aos marcadores. Esse exemplo utiliza a biblioteca OpenVRML para a renderização dos objetos VRML. A figura 6.12 exibe o resultado da execução do “simpleVRML”.



**Figura 6.12. Resultado da execução do simpleVRML**

Essa aplicação utiliza o arquivo de configuração “object\_data\_vrml” para atrelar um marcador a um objeto virtual. Localizado em “{ARToolKit}\bin\Data”, esse arquivo armazena o relacionamento entre o arquivo bitmap do marcador, disposto no diretório “Data”, e o arquivo de referência ao objeto VRML, localizado em “{ARToolKit}\bin\wrl”. A Figura 6.13 mostra o arquivo “object\_data\_vrml”, que contém dois marcadores cadastrados: o “patt.hiro” e o “patt.kanji”. O marcador “patt.hiro” está associado a um arquivo de referência VRML, denominado “bud\_B.dat”, enquanto o marcador “patt.kanji” encontra-se associado ao arquivo “snowman.dat”.

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

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

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

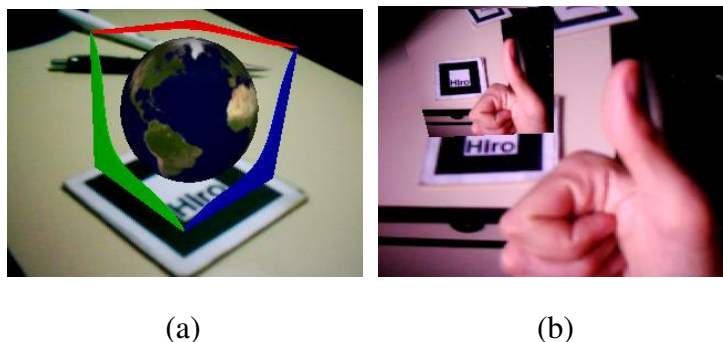
#pattern 2
VRML      Wrl/snowman.dat
Data/patt.kanji
80.0
0.0 0.0
```

**Figura 6.13. Referência de marcadores e objetos virtuais VRML**

## 6.4. Softwares baseados no ARToolKit.

Existem Softwares de Realidade Aumentada que estendem algumas técnicas de rastreamento do ARToolKit a diferentes abordagens funcionais e estruturais, no que tange os aspectos da diversidade computacional. Exemplos desses softwares são: O ARToolKitPlus considerado uma biblioteca de RA otimizada para uso em dispositivos portáteis como PDAs e alguns telefones celulares [Wagner e Schmalstieg 2003]; o jARToolKit que possibilita escrever aplicações de RA em java, acessando funções da biblioteca ARToolKit através da interface JNI

[GEIGER et al 2002]; o ARToolKit Python, um “bind” Python que encapsula as funções do ARToolKit, permitindo a exploração das vantagens do Python nas implementações, que podem ser feitas sem a compilação de código [Kirner 2007]; osgART framework baseado no ARToolKit que implementa a biblioteca gráfica OpenSceneGraph, apresentando alta qualidade na renderização de objetos virtuais e possibilidade de importar e exportar arquivos 3D gerados pelo 3D Studio Max e Maya [Looser et al 2006]. A Figura 6.14 mostra a execução de exemplos do osgART.



**Figura 6.14. Resultado da execução de exemplos do osgART: a) Exemplo osgARTsimpleNPR e b) Exemplo osgARTvideoPlane.**

## 6.5. Conclusões

O desenvolvimento de novas formas de interações em sistemas computacionais busca fornecer mecanismos adaptativos dinâmicos às necessidades dos usuários, aproximando as técnicas de manipulação do computador à intuição sensorial humana [ Kirner e Tori 2006]. Nesse contexto, está a Realidade Aumentada que promove a visualização e a manipulação de objetos modelados por computador no mundo real. Assim, é possível desenvolver aplicações altamente interativas e estimulantes associadas ao sentido visual do usuário.

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

Este capítulo apresentou uma abordagem sobre o ARToolKit, ressaltando os aspectos relacionados a seu funcionamento interno e detalhes para a sua instalação e configuração. Foram discutidos também, alguns exemplos inclusos nas versões mais recentes, que não só contribuem com o conhecimento prático de funções do ARToolKit, mas também, como modelo de partida para os usuários desenvolverem as suas próprias aplicações. Para finalizar, foram apresentados outros Softwares de RA, incluindo variações da biblioteca ARToolKit.

## Referências

- Abdullah, J.; Martinez, K. (2002) “Camera Self-Calibration for the ARToolkit”. In Proceedings of First International Augmented Reality Toolkit Workshop, pp. 84-88, Darmstadt, Germany.
- ARToolKit. Human Interface Technology Laboratory < <http://www.hitl.washington.edu/artoolkit/>> acesso em fev, 2007.

- ARToolworks, Inc.< <http://www.artoolworks.com/>>acesso em fev, 2007.
- Claus, David.; Fitzgibbon, A. W. "Reliable Automatic Calibration of a Marker-Based Position Tracking System," *wacv-motion*, pp. 300-305, Seventh IEEE Workshops on Application of Computer Vision (WACV/MOTION'05) - Volume 1, 2005.
- Consularo, L.A.; Calonego Jr, N.; Dainese, C.A.; Garbin, T. R.; Kirner, C.;Trindade, J.; Fiolhais, C.(2004) "ARToolKit: Aspectos Técnicos e Aplicações Educacionais". In: Cardoso, A.; Lamounier Jr, E. editores. Realidade Virtual: Uma Abordagem Prática. Livro dos Minicursos do SVR2004, SBC, São Paulo, 2004, p. 141-183.
- DSVideoLib,<[http://sourceforge.net/project/showfiles.php?group\\_id=75107&package\\_id=75491](http://sourceforge.net/project/showfiles.php?group_id=75107&package_id=75491)> acesso em fev, 2007.
- GEIGER, C.; REIMANN, C.; STICKLEIN, J.; PAELKE, V. (2002) "JARToolKit- A Java binding for ARToolKit." In: The First IEEE Augmented Reality ToolKit International Workshop, p124, 2002.
- GLUT, The OpenGL Utility Toolkit.< <http://www.xmission.com/~nate/glut.html>> acesso em fev, 2007
- GNU General Public License< <http://www.gnu.org/licenses/gpl.html#TOC1> >acesso em fev, 2007
- Kato, H.; Billinghurst, M.(1999) Marker Tracking and HMD Calibration for a Videobased Augmented Reality Conferencing System. In: Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality, San Francisco, CA,USA, p85-94, 1999.
- Kato, H.; Billinghurst, M.; Poupyrev, I. (2000) "ARToolKit Version 2.33", Human Interface Lab, Universidade de Washington.
- Kato, H. Inside ARToolKit.( 2002).< <http://www.hitl.washington.edu/artoolkit/Papers/ART02-Tutorial.pdf>> acesso em fev, 2007.
- Kirner, C.; Tori, R.(2004) "Introdução à Realidade Virtual, Realidade Misturada e Hiper-realidade". In: Claudio Kirner; Romero Tori. (Ed.). Realidade Virtual: Conceitos, Tecnologia e Tendências. São Paulo, 2004, v. 1, p. 3-20.
- Kirner, C.; Tori, R.(2006) "Fundamentos de Realidade Aumentada". In: Claudio Kirner; Romero Tori; Robson Siscoutto. (Ed.).Fundamentos e Tecnologia de Realidade Virtual e Aumentada. Pré Simpósio SVR 2006, SBC, Belém, 2006, pp. 22-37.
- Kirner, C.(2007) pyARToolKit <[www.ckirner.com](http://www.ckirner.com)>
- Lepetit, V.; Fua, P.(2005) "Monocular Model-Based 3D Tracking of Rigid Objects: A Survey". Foundations and Trends in Computer Graphics and Vision, Vol.1, N.1, pp. 1-89, Out 2005.
- Looser, J.; Grasset,R.; Seichter, H.; Billinghurst, B.(2006) "OSGART - A pragmatic approach to MR".Proceedings of the Fifth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR06).
- OpenVRML,< <http://www.openvrml.org/>> acesso em fev, 2007.
- Piekarski, W.; Thomas, B. H.(2002) Using ARToolKit for 3D Hand Position Tracking in Mobile Outdoor Environments, In 2nd International Augmented Reality Toolkit Workshop, Darmstadt, Set,2002.
- WAGNER, D.; SCHMALSTIEG, D.(2003) "ARToolKit on the PocketPC Platform". In:Proceedings of the 2nd Augmented Reality Toolkit Workshop, p14-15, 2003.