# Uma ferramenta de realidade aumentada usando dispositivo móvel com sistema operacional Android

Douglas Samuel Rovadosky<sup>1</sup> Willingthon Pavan<sup>2</sup> Jaqson Dalbosco<sup>2</sup> Cristiano Roberto Cervi<sup>2</sup>

**Resumo:** Este trabalho apresenta uma ferramenta para dispositivos móveis usando realidade aumentada, que tem como objetivo proporcionar aos usuários maior acessibilidade na busca de informações adicionais em um ambiente real. Além de gerar informação visual, a ferramenta auxilia portadores de deficiência visual, reproduzindo áudio com a informação. Experimentos foram realizados para a busca de livros em uma biblioteca, onde a ferramenta apresentou grande utilidade.

*Palavras-chave:* Aplicações para Android. Dispositivos móveis. Realidade aumentada.

**Abstract:** This paper presents a tool for mobile devices using augmented reality that aims to provide greater accessibility to users in finding information in a real environment. Besides generating visual information, the tool helps blind people playing audio with the information. Experiments were conducted to search for books in a library, where the tool had very useful.

**Keywords:** Applications for Android phones. Augmented reality. Mobile devices.

# 1 Introdução

As novidades tecnológicas progridem rapidamente e o futuro da computação móvel está se tornando cada vez mais promissor. Os dispositivos móveis estão cada vez mais eficientes, especialmente com a chegada dos *Smartphones*, que possuem recursos inteligentes integrados. Um dos objetivos é melhorar a mobilidade e usabilidade sem perder o desempenho, pois esses aparelhos vêm adquirindo mais capacidade de processamento e de armazenamento. Além disso, com mais recursos de conectividade, são capazes de proporcionar altas taxas de transferência de dados, podendo transferir mídias em tempo real [1].

A capacidade da evolução tecnológica é interessante. Já ocorreu com os computadores, que nos primórdios eram lentos e pesados, hoje possuem vários núcleos de processamento, produzindo a agilidade necessária para os tempos modernos em que vivemos. Com os dispositivos móveis, o caminho da evolução é o mesmo [2].

Ultimamente, há uma propagação de novas tecnologias, que vêm surgindo para aprimorar o ensino e a obtenção de informações. Uma dessas tecnologias é a realidade aumentada (RA), que possibilita a visualização de um ambiente real com objetos virtuais.

O setor de publicidade e propaganda é um dos que mais tem utilizado essa tecnologia, pois pode oferecer a seus futuros clientes uma forma de conhecer produtos e serviços por meio de aplicações com o uso da RA.

http://dx.doi.org/10.5335/rbca.2012.1813

<sup>&</sup>lt;sup>1</sup> Especialização em Desenvolvimento de Software, UPF, Campus 1 - BR 285 - Passo Fundo (RS) - Brasil {Douglas.rovadosky@gmail.com}

<sup>&</sup>lt;sup>2</sup> Curso de Ciência da Computação, UPF, Campus 1 - BR 285 - Passo Fundo (RS) - Brasil {pavan, jaqson, cervi@upf.br}

Como exemplos podem ser citados a General Motors,<sup>3</sup> que lançou o automóvel Vectra GT, possibilitando a realização de *test drive* com o uso de realidade aumentada; e o Banco Itaú,<sup>4</sup> oferecendo diversos serviços.

A mobilidade revolucionou a área tecnológica nos últimos tempos, e, usada corretamente, pode proporcionar aos usuários a comunicação a qualquer momento e em qualquer lugar, provando que a acessibilidade e portabilidade dos dispositivos móveis são um diferencial. Dessa forma, observa-se que a mobilidade digital é uma realidade emergente, oferecendo diversos rumos na área da tecnologia da informação.

O objetivo deste trabalho consiste em apresentar uma ferramenta para a busca de informação sobre ambientes, fazendo uso da tecnologia de realidade aumentada. A ferramenta desenvolvida pode ser utilizada em diversos ambientes, como supermercados, lojas, bibliotecas, museus, oportunizando maior usabilidade e acessibilidade a usuários de dispositivos móveis.

O escopo de aplicação do trabalho foi uma ampla biblioteca, que recebe diariamente um considerável número de pessoas. Localizar-se dentro de suas instalações e encontrar informações sobre livros representa, muitas vezes, uma dificuldade. A ferramenta teve como premissa facilitar a localização de livros nas prateleiras da biblioteca. Um dos pontos trabalhados foi a facilidade de portadores de deficiência visual poderem realizar a busca. Para tanto, as sessões e categorias de livros são identificadas por marcadores. A ferramenta, ao identificar um marcador, emite um som para indicar a localização do livro pretendido.

O artigo está dividido como segue. Na seção 2 apresentamos as tecnologias utilizadas, na seção 3 abordamos o desenvolvimento da ferramenta, a seção 4 apresenta os experimentos e a discussão dos resultados obtidos, e, por fim, na seção 5, são apresentadas as considerações finais.

## 2 Tecnologia utilizada

A RA pode ser definida, de forma geral, como a inclusão de objetos virtuais tridimensionais gerados digitalmente com um ambiente real através de um dispositivo tecnológico (Fig. 1). Porém, sua inclusão fica mais notável em um contexto mais amplo, o da realidade misturada (RM).



Figura 1: Objetos virtuais inseridos em um ambiente real [3]

Milgram e Kishino [4] sugeriram uma taxonomia ao analisar que o termo "realidade virtual" vinha sendo aplicado a uma série de ambientes, nem sempre com imersão total. Foi identificado que esses ambientes se encontravam em algum ponto, do que foi chamado de "contínuo de virtualidade" (virtuality continuum), onde os extremos são o ambiente real e o ambiente virtual (Fig. 2). Entre uma extremidade e outra está a RM, que cria um ambiente tão real que não se percebe a diferença entre elementos virtuais e reais, abrangendo as seguintes

\_

www.gm.com

<sup>4</sup> www itau com br

possibilidades: a virtualidade aumentada (VA), em que predomina o mundo virtual, e a RA, onde o que prevalece é o mundo real. Portanto, entende-se que a RA é uma especificação da RM.



Figura 2: Contínuo de virtualidade. Adaptado de [4]

A realidade aumentada pode ser conceituada de diversas maneiras, com base em diferentes autores [3]:

- é uma melhoria do mundo real com textos, imagens e objetos virtuais gerados por computador [3];
- é a mistura de mundos reais e virtuais em algum ponto da realidade/virtualidade contínua que conecta ambientes completamente reais a ambientes completamente virtuais ([4] apud [3]);
- é um sistema que suplementa o mundo real com objetos virtuais gerados por computador, parecendo coexistir no mesmo espaço e apresentando as seguintes propriedades: combina objetos reais e virtuais no ambiente real; executa interativamente em tempo real; alinha objetos reais e virtuais entre si; aplica-se a todos os sentidos, incluindo audição, tato, força e cheiro ([5] apud [3]).

A RA, de acordo com Kirner e Zorzal [6], possui tipos de sistemas que podem ser classificados conforme o tipo de display utilizado, envolvendo visão ótica ou visão por vídeo, dando origem a quatro tipos de sistemas:

- sistema de visão ótica direta utiliza óculos ou capacetes com lentes que permitem o recebimento direto da imagem real, ao mesmo tempo em que possibilitam a projeção de imagens virtuais devidamente ajustadas com a cena real;
- sistema de visão direta por vídeo utiliza capacetes com microcâmeras de vídeo acopladas. A cena real, capturada pela microcâmera, é misturada com os elementos virtuais gerados por computador e apresentadas diretamente nos olhos do usuário, através de pequenos monitores montados no capacete;
- sistema de visão por vídeo baseado em monitor utiliza uma webcam para capturar a cena real. Depois de capturada, a cena real é misturada com os objetos virtuais gerados por computador e apresentada no monitor. O ponto de vista do usuário normalmente é fixo e depende do posicionamento da webcam;
- sistema de visão ótica por projeção utiliza superfícies do ambiente real, onde são projetadas imagens dos objetos virtuais, cujo conjunto é apresentado ao usuário que o visualiza sem a necessidade de nenhum equipamento auxiliar.

Para o desenvolvimento de um aplicativo com a tecnologia de RA é necessário fazer o uso de uma API (application programming interface) que utiliza a tecnologia de RA. A API mais conhecida é a ARToolKit, porém existem diversas outras ferramentas, tais como NyArtoolKit, Dart, ARTag, ou OSGART, todas baseadas na ARToolKit.

Para que uma aplicação seja desenvolvida para ser executada no sistema operacional Android utilizandose a linguagem Java, é necessário usar o ambiente de desenvolvimento Eclipse, que possui um *plugin* chamado "Android development tools" (ADT).

Para distribuir uma aplicação Android é necessário compilar o projeto Java em um arquivo de extensão .apk (*Android package file*), que é um arquivo binário compactado com as classes compiladas e os recursos do projeto. Para executar uma aplicação em um dispositivo móvel, também é necessário instalar o arquivo .apk.

O protótipo foi desenvolvido utilizando-se as seguintes tecnologias: linguagem de programação Java, C/C++, biblioteca Open GL, Java development Kit (JDK), Eclipse IDE, Android software development Kit (SDK), Cygwin, Android native development Kit (NDK), SDK Qualcomm augmented reality (QCAR), um Smartphone Nexus One Google com sistema operacional Android 2.1 e um computador com sistema operacional Windows 7 de 32 bits.

## 3 Ferramenta desenvolvida

Para o desenvolvimento da ferramenta foi utilizado o conceito de visão por vídeo baseado em monitor, que utiliza uma webcam para capturar a cena real. Depois de capturada, a cena real é misturada com objetos virtuais gerados por computador e mostrados na tela. O funcionamento consiste em capturar um marcador predefinido no sistema que reconheça essa imagem retornando o resultado da pesquisa. A ferramenta, ainda, dispara um áudio juntamente com a informação visual da tela do dispositivo móvel. Dessa forma, o aplicativo permite que portadores de necessidades especiais sejam auxiliados de forma dinâmica, oportunizando maior acessibilidade na busca de informações.

Foram identificadas algumas bibliotecas (API) para desenvolvimento em RA, conforme a Tab. 1.

Tabela 1: Ferramentas para desenvolvimento de software em RA

rabela 1. 1 erramentas para desenvolvimento de software em 101		
Ferramenta	Descrição	
ARToolKit	Biblioteca para desenvolvimento de interfaces para RA na linguagem C e C++.	
ARTag DART OSGART	Ferramentas para desenvolvimento em RA baseado na ARToolKit.	
NYARToolKit	Ferramenta para construção de software em RA, construída 100% em Java puro.	

A biblioteca NyArtoolKit foi desenvolvida no Japão e é baseada na ArtoolKit. Foi construída totalmente em Java e não usa código nativo. Foram realizados alguns testes com exemplos de aplicações que utilizaram a NyArtoolkit, onde foi constatado que a mesma não atenderia às necessidades do projeto, uma vez que não utiliza código nativo, o que torna a aplicação mais lenta, tanto na renderização da câmera quanto no resultado final, que é a imagem 3D para o usuário. As demais bibliotecas não foram testadas, pois não utilizavam código nativo.

## 3.1 Funcionamento da ferramenta

Na visão do usuário, a ferramenta desenvolvida funciona simplesmente com o reconhecimento do marcador, retornando o resultado visual e audível da pesquisa. Uma visão mais detalhada do funcionamento da ferramenta é apresentada na Figura 3.

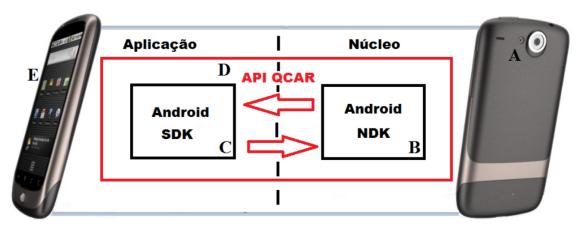


Figura 3: Funcionamento da ferramenta

O aplicativo inicia a câmera (A) do aparelho, capturando frames que são exibidas de acordo com a direção em que a câmera é apontada. No instante em que um marcador predefinido é encontrado no sistema, esses frames são identificados pelo Android NDK (B), que executa as funções nativas referentes à criação da imagem 3D. O NDK tem um papel fundamental no desempenho dessa ação, pois permite construir partes de desempenho crítico do sistema em código nativo, tendo melhor acessibilidade ao núcleo do processamento. Portanto, acelera e oferece a este um melhor desempenho. O NDK é projetado somente para o uso em conjunto com o Android SDK e é necessário que esses códigos nativos sejam compilados para que o Android SDK (C) os reconheca.

O Android SDK é um software utilizado para desenvolver aplicações no Android. Possui um emulador de celular, ferramentas utilitárias e uma API completa para a linguagem Java, com todas as classes necessárias para desenvolver as aplicações. A API (D) faz a comunicação entre o NDK e o SDK, possibilitando a troca das informações que são processadas onde o hardware é mais exigido, no qual atua o NDK. Portanto, a chamada das funções nativas ocorrem pela API, e a aplicação usa a tela (E) do dispositivo móvel como uma "lente mágica", aumentando o mundo real com objetos virtuais. O aplicativo processa a imagem ao vivo da câmera na tela para representar uma visão do mundo físico. Os objetos virtuais 3D são sobrepostos e parecem estar intimamente ligados, no mundo real, juntamente com o áudio que é disparado, alcançando o resultado final.

#### 3.2 Geração de imagem 3D e disparo de áudio

Para gerar a imagem 3D que o usuário recebe como resposta da pesquisa, é preciso trabalhar em código nativo (NDK), utilizando a linguagem de programação C/C++, a biblioteca OpenGL e a linguagem Java, que é tratada pela SDK do Android. Inicialmente, é necessário desenhar o fundo do vídeo e limpar o buffer de cores, que é responsável por armazenar informações de cor (RGBA) para cada pixel e o buffer de profundidade, que armazena a distância de cada pixel até o observador, para que a configuração da tela (quadro) seja visualizada em modo de perspectiva. Isso significa que quanto maior a distância do observador, menores os objetos ficarão, o que cria uma cena com aspecto muito realista. Para adicionar perspectiva à cena, é necessário obter a matriz de projeção. Sempre que uma cena for desenhada, os buffers devem ser inicializados, ou seja, habilitados (Fig. 4).

```
INTEXPORT world INTCALL
Java_com_qualcomm_QCARSamples_FrameMarkers_FrameMarkersRenderer_renderFrame(JNIEnv* env, jobject obj)
    //LOG("Java com qualcomm QCARSamples FrameMarkers GLRenderer renderFrame");
    // limpar a cor e o buffer de profundidade
   glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    // renderiza o fundo do video
   OCAR::State state = OCAR::Renderer::getInstance().begin();
    // obtem a matriz de projeção
    const OCAR::Tracker& tracker
        QCAR::Tracker::getInstance();
    const QCAR::CameraCalibration& cameraCalibration =
        tracker.getCameraCalibration();
    QCAR::Matrix44F projectionMatrix
        QCAR::Tool::getProjectionGL(cameraCalibration, 2.0f, 2000.0f);
    //habilita o buffer de profundidade
   glEnable (GL DEPTH TEST);
    //habilita a remocao das faces traseiras
   glEnable(GL CULL FACE);
```

Figura 4: Trecho do código identificando o texto

Com a ferramenta inicializada, a câmera permanece capturando imagens até encontrar o marcador. No momento em que o sistema identifica o marcador, já é realizada a indicação da textura com base no ID do marcador, e também é enviado o nome do marcador por meio do método displayMessage para a SDK (Fig. 5).

```
// laço para encontrar algum marcador
for(int tIdx = 0; tIdx < state.getNumActiveTrackables(); tIdx++)</pre>
    // pega o marcador
        st QCAR::Trackable* trackable = state.getActiveTrackable(tIdx);
    OCAR::Matrix44F modelViewMatrix =
        QCAR::Tool::convertPose2GLMatrix(trackable->getPose());
    // verifica o tipo do marcador
    // verifica of tape to markedor
assert(trackable-ygetType() == QCAR::Trackable::MARKER);
const QCAR::Marker* marker = static_cast<const QCAR::Marker*>(trackable);
    //se é um novo alvo, encontra o metodo displaymessage no java e chama com o nome do alvo
    if ( marker->getName() != lastTrackableId) {
         istring is = env->NewStringUTF( marker->getName());
         jclass javaClass = env->GetObjectClass(obj);
jmethodID method = env->GetMethodID(javaClass, "displayMessage", "(Ljava/lang/String;)V");
         env->CallObjectMethod(obj, method, js);
         lastTrackableId = marker->getName();
    // textura com base no id do marcador
    textureIndex = marker->getMarkerId();
    assert(textureIndex < textureCount);
    const Texture* const thisTexture = textures[textureIndex];
    // desenha a imagem
    const GLvoid* vertices = 0;
const GLvoid* normals = 0;
    const GLvoid* indices = 0;
    const GLvoid* texCoords = 0;
    int numIndices = 0;
    vertices = &ObjectVertices[0];
    normals = &ObjectNormals[0];
    indices = &ObjectIndices[0];
    texCoords = &ObjectTexCoords[0]:
    numIndices = NUM_OBJECT_INDEX;
```

Figura 5: Parte da programação citada no texto

Em seguida, o painel 3D é desenhado com base nas posições e dimensões indicadas nos vértices, coordenadas, normais e índices (Fig. 6). Isso ocorre em código nativo, portanto, no NDK.

Figura 6: Código que define as dimensões do Painel 3D

Na SDK, a linguagem Java manipula as chamadas dos métodos nativos do NDK, herdando as classes da biblioteca QCAR. Para chamar os métodos nativos é preciso estender as classes da API, pois esta estabelece a comunicação, tornando o trabalho mais produtivo. Para renderizar a câmera, é necessário estender a classe QCARSampleRenderer (Fig. 7) para que haja comunicação entre o SDK e o NDK.

```
//estende a classe
public class FrameMarkersRenderer extends OCARSampleRenderer
    //A função nativa que renderiza
    public native void renderFrame():
   // objeto que manipula envio de mensagens
    public static Handler mainActivityHandler;
    // chamado apartir da nativa para mostrar a mensagem
    public void displayMessage(String text)
        Message message = new Message();
        message.obj = text;
        mainActivityHandler.sendMessage(message);
    //chama para desenhar
    public void onDrawFrame (GL10 gl)
        if (!mIsActive)
            return;
        //chama a funcao nativa para processar
        renderFrame();
```

Figura 7: Trecho do código estendendo a classe

Para adicionar a textura, a classe QCARSampleActivity da API deve ser estendida, pois a função nativa possui a indicação da textura e somente é vinculada através da SDK do Android. O áudio é tratado somente no Java, pois recebe a identificação do nome do marcador por meio do método displayMessage da função nativa. A partir dessa identificação, o áudio correto é disparado. Para reproduzi-lo é preciso utilizar a classe android.media.MediaPlayer, onde abre o arquivo mp3. Para iniciá-lo é usado o método start, conforme a Figura 8.

```
public void configPlayer() {
    player.setOnCompletionListener(new OnCompletionListener() {
        public void onCompletion(MediaPlayer mp) {
            if(listaTelas.size()>1){
               Log.println(Log.DEBUG, "targetREMOVE", listaTelas.get(0));
                listaTelas.remove(0):
                Log.println(Log.DEBUG, "targetPLAY", listaTelas.get(0));
                playSound(listaTelas.get(0));
            }else{
               player = null;
    });
public void playSound(String opcao) {
    Log.println(Log.DEBUG, "targetoption", opcao);
    if(opcao.equalsIgnoreCase("mymarker1")){
        if(player != null){
            if(!player.isPlaying()){
                player.stop();
                player = MediaPlayer.create(this, R.raw.psicoterapia);
                configPlayer();
                player.start();
        }else{
            player = MediaPlayer.create(this, R.raw.psicoterapia);
            configPlayer();
            player.start();
    }else if(opcao.equalsIgnoreCase("mymarker2")){
        if(player != null){
            if(!player.isPlaying()){
                player.stop();
                player = MediaPlayer.create(this, R.raw.legislacao);
                configPlayer();
```

Figura 8: Trecho do código que reproduz o áudio

Como a ferramenta pode reconhecer mais de um marcador num mesmo quadro, foi necessária a criação de um *array* que armazenasse o reconhecimento de cada marcador, fazendo uma espécie de "fila de espera" para reproduzir o áudio. Portanto, identifica vários marcadores e reproduz o áudio conforme a fila.

## 3.3 Geração de marcador (alvo)

A API QCAR, além do recurso de detectar e monitorar marcadores, controla um tipo especial denominado "marcador de quadro" (*frame marker*). A identificação exclusiva de um marcador é codificada em um padrão binário ao longo da borda da imagem. A API exige o quadro e o padrão binário totalmente visíveis no reconhecimento da imagem na câmera.

Apesar da Qualcomm disponibilizar um sistema de gerenciamento de alvos (*Target Management System*) online, os marcadores de quadro utilizados na aplicação (512 marcadores) são distribuídos juntamente com a API. Para dimensioná-los em um tamanho adequado para impressão, basta escolher um editor de imagem, que pode ser detectado conforme a dimensão do marcador, sendo ativado e definido no arquivo config.xml, conforme a Figura 9.

Figura 9: Exemplo do arquivo config.xml

Em razão da necessidade de processamento relativamente baixo para decodificar o ID do marcador, todos os 512 marcadores podem ser usados em uma aplicação. Porém, no máximo, cinco marcadores podem ser detectados em um quadro.

Um marcador é composto por quatro áreas, e cada uma tem uma função específica, conforme a Figura 10.

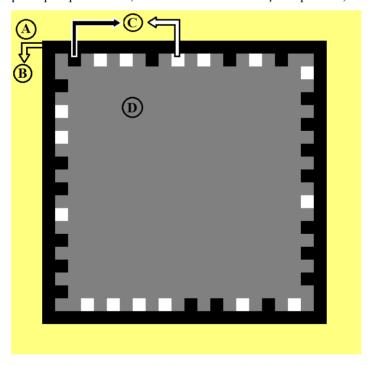


Figura 10: Exemplo de um marcador de quadro

As áreas foram identificadas na figura como A, B, C e D, e suas funções são:

- A: uma área ao redor da moldura preta do marcador deve ser deixada livre de elementos gráficos, ter um bom contraste da moldura, e é preferível ser duplamente mais larga que a moldura;
- B: a moldura preta é utilizada para reconhecer o marcador no ambiente visto pela câmera durante o tempo de execução;
- C: a área de identificação dentro do quadro codifica a identificação do marcador em um padrão binário;
- D: a área do design (cinza) é uma área livre, podendo ser inserida qualquer imagem ou informação, tornando, por exemplo, o marcador mais atraente. Essa área do marcador não é avaliada pela API.

A API também possibilita que imagens-alvo sejam criadas através de ferramenta online (sistema de gerenciamento de alvos). As características são extraídas a partir dessas imagens e são armazenadas em um banco de dados. A API permite, além das imagens de entrada múltipla no banco de dados, a detecção e rastreabilidade de até cinco alvos simultaneamente. O desempenho pode variar dependendo da carga sobre o processador. A API possibilita que até cinquenta imagens alvo sejam amazenadas em seu banco de dados. Pelo site <a href="https://ar.qualcomm.com/qdevnet/">https://ar.qualcomm.com/qdevnet/</a> pode-se adicionar essas imagens, onde serão mapeadas, tendo os alvos a elas adicionados, conforme a Figura 11.

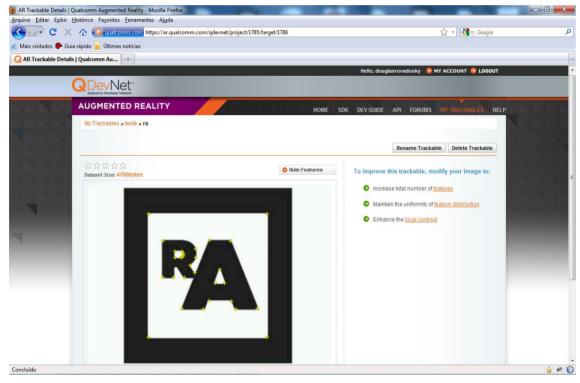


Figura 11: Imagem com os alvos

Como mencionado, para a identificação dessas imagens no programa é necessário a existência de um arquivo config.xml, sendo esse disponibilizado pela ferramenta.

# 4 Experimentos e análise de resultados

#### 4.1 Estudo de caso

Para validar a ferramenta proposta, foi desenvolvido um protótipo para a biblioteca da Universidade de Passo Fundo. O protótipo permite aos usuários maior acessibilidade na busca por informações, como a sessão e categoria dos livros, auxiliando, principalmente, aos portadores de deficiência visual, pois, além da informação

visual gerada na tela do dispositivo, o aplicativo também dispara um áudio no mesmo instante em que a resposta visual é gerada.

A ferramenta consiste em identificar um marcador que está fixado nas estantes e prateleiras no ambiente da biblioteca. Com o reconhecimento do marcador, o aplicativo gera as informações adicionais de uma determinada sessão e categoria. Esse reconhecimento pode ser com mais de um marcador simultaneamente (Fig. 12). Assim, são geradas mais de uma informação na tela do dispositivo, no entanto, o áudio é reproduzido um após o outro, seguindo a ordem da identificação dos marcadores.

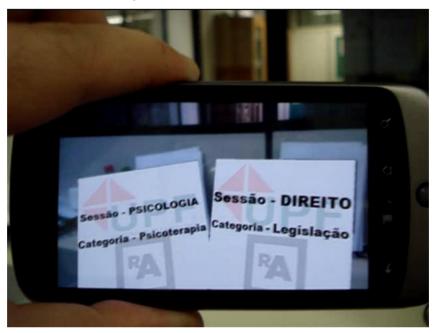


Figura 12: Reconhecimento de dois marcadores simultaneamente

A experiência foi realizada para o ambiente da biblioteca da UPF, que recebe diariamente um número expressivo de pessoas, dentre as quais alunos, professores e funcionários. Em razão de ser um ambiente amplo e com um grande acervo, há certa dificuldade em se localizar e encontrar informações sobre livros. A ferramenta também poderia ser aplicada em diversos outros ambientes, como em um supermercado, possibilitando informar dados adicionais de produtos ou em museus, informando dados de obras históricas.

Para o funcionamento do aplicativo, basta instalar o arquivo .apk no dispositivo móvel com Android, porém a API só possui suporte para alguns dispositivos móveis, conforme a Tabela 2.

## 4.2 Trabalho relacionado

As pesquisas sobre a tecnologia de RA se intensificaram no final da década de 1990. Pesquisadores do mundo inteiro tendem a aprimorar, cada vez mais, suas experiências com pesquisas mais promissoras, agregando aos usuários uma realidade notória.

Um trabalho na mesma linha foi desenvolvido por Eike J. Umlauf e Harald Piringer, na Universidade de Tecnologia de Viena, Áustria, no ano de 2002. O aplicativo denominado "ARLib" ou "ARLibrary" visava auxiliar na seleção e localização de livros em uma biblioteca através da RA. As principais funcionalidades desse aplicativo eram a procura por um livro específico através de uma interface intuitiva de busca tendo sua posição sobre as prateleiras, bem como obter informações adicionais sobre a publicação, podendo ser útil para o regresso do livro para a prateleira da biblioteca [7].

O sistema consiste em um notebook acoplado numa mochila, um capacete com uma câmera *firewire* e uma espécie de óculos com visor, além de um *wrist-worn*, que é um aparelho de pulso que controla a entrada de dados em um *touch pad*, conforme a Figura 13.

Tabela 2: Dispositivos que a ferramenta suporta

Marca	Modelo	SO
Google	Nexus One	Android 2.1, 2.2
HTC	Desire	Android 2.1, 2.2
HTC	Incredible	Android 2.1, 2.2
HTC	EVO 4G	Android 2.1, 2.2
HTC	EVO Shift 4G	Android 2.2
HTC	G2 (T-Mobile) / Desire Z	Android 2.2
HTC	Desire HD / Inspire 4G	Android 2.2
HTC	MyTouch 4G	Android 2.2
Dell	Streak	Android 2.2
Dell	Venue	Android 2.2
Motorola	Droid 2	Android 2.2
Motorola	Droid X	Android 2.2
Sony Ericsson	Xperia X10	Android 2.1
Samsung	Galaxy S SPH-D700	Android 2.1, 2.2
Samsung	Galaxy S SCH-I500	Android 2.1
Samsung	Galaxy S GT-I9000	Android 2.1, 2.2
Samsung	Galaxy S SGH-I897	Android 2.1, 2.2
Samsung	Galaxy S SGH-T959	Android 2.1, 2.2
Samsung	Nexus S	Android 2.3
Huawei	Ideos S7	Android 2.1
Huawei	Ideos x5 U8800	Android 2.2.1
Other Snapdragon devices	Any	Android 2.1, 2.2



Figura 13: Usuário trabalhando com a ARLib

Para o desenvolvimento do aplicativo ARLibrary foram utilizadas as ferramentas ARToolKit e Studierstube. A interface principal do ARLibrary para o usuário é o wrist-worn, que é totalmente dependente da

Studierstube, pois os widgets (componentes gráficos, como janelas e botões) visualizados no aparelho de pulso proporcionam uma interface amigável aos usuários.

#### 4.3 Discussão

Um aplicativo móvel, que atenda as necessidades dos usuários de forma prática e eficiente, deve oferecer alguns requisitos essenciais, como portabilidade, usabilidade e funcionalidade, sendo esses requisitos características da mobilidade. Em relação à computação móvel, a mobilidade pode ser definida como o uso de dispositivos móveis que possibilitam a capacidade de realizar facilmente um conjunto de funções de um aplicativo. Um dispositivo móvel deve apresentar algumas características, como ser portátil (capaz de ser transportado com relativa facilidade), utilizável e funcional, permitindo fácil conectividade e comunicação com os usuários e outros dispositivos.

A ferramenta aqui apresentada traz, em sua essência, o conceito de portabilidade, usabilidade e funcionalidade. A palavra smartphone entrou para o vocabulário dos usuários, assim como computação em nuvens, tablets, e-book readers, lojas de aplicativos e internet 3G. Todas são tecnologias emergentes e reais ao alcance de uma parte importante dos usuários, tornando o uso da mobilidade digital um fenômeno da atualidade. Com a ajuda de um smartphone com Android e a tecnologia de RA, foram obtidas essas características, permitindo aos usuários de forma simplificada e prática a procura por livros dentro da biblioteca de uma universidade.

O fator chave que diferencia esse trabalho do projeto do ARLibrary é a possibilidade de uma maior acessibilidade na localização de livros, especialmente por deficientes visuais, pois além de gerar a informação desejada visualmente, identifica também a pesquisa através de áudio. Outro fator importante é a facilidade de movimentação que o usuário pode ter, pois o dispositivo móvel cabe na palma da mão (Fig. 14), melhorando a qualidade visual, sendo mais confortável, leve, barato e discreto, ao passo que o outro utiliza uma série de equipamentos pesados e inadequados para o deslocamento do usuário (Fig. 13), sob pena de não ser utilizado.



Figura 14: Aplicativo em funcionamento

Este trabalho demonstra que as vantagens de aderir à tecnologia de realidade aumentada são evidentes, pois sua capacidade de gerar benefícios é muito grande.

## 5 Considerações finais

A conveniência e a poderosa funcionalidade oferecida pelos dispositivos móveis, como *Smartphones*, tiveram como resultado a opção de muitas pessoas pelo uso da mobilidade. Com o crescimento do mercado corporativo, muitas empresas estão incorporando aplicações móveis a seu dia a dia para agilizar seus negócios.

Num mundo onde a mobilidade está cada vez mais conhecida, este trabalho teve como objetivo a implementação de uma ferramenta para dispositivos móveis utilizando a tecnologia de realidade aumentada, tendo como estudo de caso a biblioteca da Universidade de Passo Fundo. Com o uso dessa ferramenta, os usuários podem obter maior acessibilidade a informações e localização de livros, especialmente os deficientes visuais, pois por meio da ferramenta desenvolvida ouvem áudio com a identificação da sessão e categoria dos livros

Este trabalho foi desenvolvido para verificar a possibilidade da implementação de um software com o uso da RA. Futuramente será possível aprimorá-lo, incluindo todas as sessões e categorias da biblioteca da universidade utilizada como estudo de caso, disponibilizando-o aos usuários finais.

É possível que em um futuro próximo a RA atinja todo o nosso meio, sendo que em algum momento as pessoas possam utilizar aparelhos, como uma espécie de óculos, que insiram objetos virtuais no meio real. Ao ir a um supermercado e visualizar um produto, por exemplo, o cliente poderá obter informações adicionais a respeito através da tecnologia de RA. Outros exemplos seriam a utilização em anúncios em outdoors, e uma nova forma de mídia social que ainda venha ser inventada. Tudo isso sendo integrado à visão do mundo físico, de tal forma a não ser uma distração, mas sim uma complementação.

Um exemplo que já existe e auxilia os motoristas no trânsito é um sistema de RA para Iphone com uso de GPS e de câmera, que deve estar fixado ao para-brisa. O aplicativo, denominado "Augumented driving", escaneia a área visível do motorista fornecendo informações de inclinações na pista, distância entre os veículos e velocidade do automóvel [8].

## Referências

- [1] LEE, Valentino; SCHNEIDER, Heather; SCHELL, Robbie. *Aplicações móveis*: arquitetura, projeto e desenvolvimento. São Paulo: Pearson Education do Brasil, 2005.
- [2] REVISTA DROPS. Realidade aumentada: das telas de ficção a sua webcam. Passo Fundo, v. 25, 2010.
- [3] KIRNER, Cláudio; SISCOUTTO, Robson. *Realidade virtual e aumentada*: conceitos, projeto e aplicações. Porto Alegre: Sociedade Brasileira de Computação, 2007.
- [4] MILGRAM, Paul; KISHINO, Fumio. A taxonomy of mixed reality visual displays. *IEICE Transactions on Information Systems*, v. E77-D, n. 12, Dec. 1994.
- [5] AZUMA, Ronald; BAILLOT, Yohan; BEHRINGER, Reinhold. Recent advances in augmented reality. *IEEE Computer Graphics and Applications*, v. 21, n. 6, p. 34-47, Nov. 2001.
- [6] KIRNER, Cláudio; ZORZAL, Ezequiel R. Aplicações educacionais em ambientes colaborativos com realidade aumentada. In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, XVI, 2005. UFJF, Juiz de Fora – MG, 2005.
- [7] UMLAUF, Eike; PIRINGER, Harald. ARLib: The augmented library. In Proc. of the ART02, p. 188/2-2002/10, 2002.
- [8] IMAGINYZE. Disponível em: < http://www.imaginyze.com/Site/Features.html >. Acesso em: 17 jan. 2011.