

Enhance your virtual world designs with the
power of Android augmented reality



Pro
Android
Augmented Reality

Raghav Sood



Apress®

Pro Android Augmented Reality



Raghav Sood

Apress®

Pro Android Augmented Reality

Copyright © 2012 by Raghav Sood

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed. Exempted from this legal reservation are brief excerpts in connection with reviews or scholarly analysis or material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work. Duplication of this publication or parts thereof is permitted only under the provisions of the Copyright Law of the Publisher's location, in its current version, and permission for use must always be obtained from Springer. Permissions for use may be obtained through RightsLink at the Copyright Clearance Center. Violations are liable to prosecution under the respective Copyright Law.

ISBN 978-1-4302-3945-1

ISBN 978-1-4302-3946-8 (eBook)

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The images of the Android Robot (01 / Android Robot) are reproduced from work created and shared by Google and used according to terms described in the Creative Commons 3.0 Attribution License. Android and all Android and Google-based marks are trademarks or registered trademarks of Google, Inc., in the U.S. and other countries. Apress Media, L.L.C. is not affiliated with Google, Inc., and this book was written without endorsement from Google, Inc.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

President and Publisher: Paul Manning

Lead Editor: Steve Anglin

Technical Reviewers: Yosun Chang, Chad Darby

Editorial Board: Steve Anglin, Ewan Buckingham, Gary Cornell, Louise Corrigan, Morgan Ertel, Jonathan Gennick, Jonathan Hassell, Robert Hutchinson, Michelle Lowman, James Markham, Matthew Moodie, Jeff Olson, Jeffrey Pepper, Douglas Pundick, Ben Renow-Clarke, Dominic Shakeshaft, Gwenan Spearing, Matt Wade, Tom Welsh

Coordinating Editors: Corbin Collins, Christine Ricketts

Copy Editors: Vanessa Moore; Nancy Sixsmith, ConText Editorial Services

Compositor: Bytheway Publishing Services

Indexer: SPi Global

Artist: SPi Global

Cover Designer: Anna Ishchenko

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com.

For information on translations, please e-mail rights@apress.com, or visit www.apress.com.

Apress and friends of ED books may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Special Bulk Sales—eBook Licensing web page at www.apress.com/bulk-sales.

Any source code or other supplementary materials referenced by the author in this text is available to readers at www.apress.com. For detailed information about how to locate your book's source code, go to www.apress.com/source-code.

To my family and friends

-Raghav Sood

Contents at a Glance

■ About the Author	xi
■ About the Technical Reviewers	xii
■ Acknowledgments	xiii
■ Introduction	xiv
■ Chapter 1: Applications of Augmented Reality	1
■ Chapter 2: Basics of Augmented Reality on the Android Platform	13
■ Chapter 3: Adding Overlays	41
■ Chapter 4: Artificial Horizons.....	65
■ Chapter 5: Common and Uncommon Errors and Problems	95
■ Chapter 6: A Simple Location-Based App Using Augmented Reality... ..	107
■ Chapter 7: A Basic Navigational App Using Augmented Reality... ..	141
■ Chapter 8: A 3D Augmented Reality Model Viewer	159
■ Chapter 9: An Augmented Reality Browser.....	221
■ Index	319

Contents

■ About the Author	xi
■ About the Technical Reviewers	xii
■ Acknowledgments	xiii
■ Introduction	xiv
■ Chapter 1: Applications of Augmented Reality	1
Augmented Reality vs. Virtual Reality	1
Current Uses	1
Casual Users.....	2
Military and Law Enforcement.....	4
Vehicles	4
Medical	5
Trial Rooms.....	6
Tourism.....	6
Architecture.....	6
Assembly Lines.....	7
Cinema/Performance.....	7
Entertainment.....	7
Education.....	8
Art.....	8
Translation.....	8
Weather Forecasting	9
Television	9
Astronomy	9
Other.....	9
Future Uses	10
Virtual Experiences.....	10
Impossible Simulations	10

Holograms	11
Video Conferencing.....	11
Movies	11
Gesture Control.....	12
Summary.....	12
■ Chapter 2: Basics of Augmented Reality on the Android Platform	13
Basics of Augmented Reality on the Android Platform.....	13
Creating the App	13
Camera.....	14
Orientation Sensor	21
Accelerometer.....	24
Global Positioning System (GPS)	28
Latitude and Longitude)	29
ProAndroidAR2Activity.java.....	32
AndroidManifest.xml.....	36
main.xml.....	37
Sample LogCat Output	37
Summary	38
■ Chapter 3: Adding Overlays	41
Adding Overlays.....	41
Widget Overlays	41
Layout Options.....	43
Updating main.xml with a RelativeLayout	45
TextView Variable Declarations.....	49
Updated onCreate	49
Displaying the Sensors' Data.....	49
Updated AndroidManifest.xml	52
Testing the App	53
Markers.....	54
Activity.java.....	54
CustomObject Overlays	56
CustomRenderer	60
AndroidManifest	62
Summary	63
■ Chapter 4: Artificial Horizons.....	65
A Non-AR Demo App	65
The XML.....	66

The Java	67
The Android Manifest.....	77
Testing the Completed App.....	78
An AR Demo App	80
Setting Up the Project.....	81
Updating the XML	82
Updating the Java Files	86
Testing the Completed AR app.....	92
Summary.....	93
■ Chapter 5: Common and Uncommon Errors and Problems	95
Layout Errors	95
UI Alignment Issues	95
ClassCastException	96
Camera Errors.....	97
Failed to Connect to Camera Service	97
Camera.setParameters() failed.....	98
Exception in setPreviewDisplay().....	99
AndroidManifest Errors.....	100
Security Exceptions	100
<uses-library>	101
<uses-feature>	101
Errors Related to Maps	102
The Keys.....	102
Not Extending MapActivity.....	102
Debugging the App.....	103
LogCat	103
Black and White Squares When Using the Camera	104
Miscellaneous	105
Not Getting a Location Fix from the GPS	105
Compass Not Working	105
Summary.....	106
■ Chapter 6: A Simple Location-Based App Using Augmented Reality...	107
A Simple Location-Based App Using Augmented Reality and the Maps API .	107
Editing the XML.....	109
Creating Menu Resources	111
Layout Files	112
Getting API Keys	117
Getting the MD5 of Your Keys.....	118

Java Code	118
Main Activity	119
FlatBack.java	128
FixLocation.java.....	132
Running the App	135
Common errors	138
Summary.....	139
■ Chapter 7: A Basic Navigational App Using Augmented Reality... ..	141
The New App	141
Updated XML files.....	142
Updated Java files	145
Updated AndroidManifest	155
The Completed App	155
■ Chapter 8: A 3D Augmented Reality Model Viewer	159
Key Features of this App.....	160
The Manifest	162
Java Files.....	163
Main Activity	163
AssetsFileUtility.java.....	168
BaseFileUtil.java	169
CheckFileManagerActivity.java.....	170
Configuration File	175
Working with Numbers	175
Group.java	178
Instructions.java	180
Working with Light	181
Creating a Material	183
MemUtil.java.....	186
Model.java	187
Model3D.java	189
Viewing the Model.....	191
Parsing .mtl files.....	199
Parsing the .obj files.....	203
ParseException	207
Rendering	207
SDCardFileUtil.java	209
SimpleTokenizer.java	210
Util.java.....	211
3D Vectors	213
XML Files.....	214

Strings.xml	214
Layout for the Rows.....	215
instructions_layout.xml	215
List Header	216
main.xml.....	216
HTML Help File	217
Completed App.....	219
Summary	220
■ Chapter 9: An Augmented Reality Browser.....	221
The XML	222
strings.xml.....	222
menu.xml.....	222
The Java Code.....	223
The Activities and AugmentedView	223
Getting the Data	242
DataSource	242
LocalDataSource.....	242
NetworkDataSource	243
TwitterDataSource.....	246
WikipediaDataSource	250
Positioning Classes.....	252
ScreenPositionUtility	254
The UI Works	255
PaintableObject	255
PaintableBox.....	258
PaintableBoxedText.....	260
PaintableCircle.....	263
PaintableGps.....	264
PaintableIcon	265
PaintableLine	266
PaintablePoint.....	267
PaintablePosition.....	268
PaintableRadarPoints	270
PaintableText.....	271
Utility Classes	273
Vector	273
Utilities.....	277
PitchAzimuthCalculator	277
LowPassFilter	278
Matrix	280

Components 285
 Radar 285
 Marker 289
 IconMarker.java 302

Customized Widget 303
 VerticalSeekBar.java 304

Controlling the Camera 305
 CameraSurface.java 305
 CameraCompatibility 309
 CameraModel 310

The Global Class 311
 ARData.java 311
 AndroidManifest.xml 316
 Running the App 317

Summary 318

■ **Index 319**

About the Author



■ **Raghav Sood**, born on April 16, 1997, is a young Android developer. He started seriously working with computers after learning HTML, CSS, and JavaScript while making a website at the age of nine. Over the next three years, Raghav developed several websites and quite a few desktop applications. He has learned several programming languages, including PHP, Java, x86 assembly, PERL, and Python. In February 2011, Raghav received his first Android device, an LG Optimus One running Froyo. The next day, he began work on his first Android app. He is currently the owner of an Android tutorial site, an author on the Android Activist site and the developer of 12 Android apps. Raghav regularly takes part in the android-developers Google Group, trying to help whomever he can. Raghav also enjoys reading, photography and robotics. He currently resides in New Delhi, India. This is his first book.

About the Technical Reviewers

■ **Yosun Chang** has been creating apps for iOS and Android since early 2009, and is currently working on a next generation 3D and augmented reality mobile games startup called nusoy. Prior to that, since 1999 she did web development on the LAMP stack and Flash. She has also spoken at several virtual world, theater, and augmented reality conferences under her artist name of Ina Centaur. She has a graduate level background in physics and philosophy from UC San Diego and UC Berkeley. An avid reader who learned much of her coding chops from technical books like the current volume, she has taken care to read every single word of the chapters she reviewed — and vet the source. Contact her @yosunchang on Twitter.

■ **Chád Darby** is an author, instructor, and speaker in the Java development world. As a recognized authority on Java applications and architectures, he has presented technical sessions at software development conferences worldwide. In his 15 years as a professional software architect, Chád has had the opportunity to work for Blue Cross/Blue Shield, Merck, Boeing, Northrop Grumman, and a handful of startup companies.

Chád is a contributing author to several Java books, including *Professional Java E-Commerce* (Wrox Press), *Beginning Java Networking* (Wrox Press), and *XML and Web Services Unleashed* (Sams Publishing). Chád has Java certifications from Sun Microsystems and IBM. He holds a B.S. in Computer Science from Carnegie Mellon University.

You can read Chád's blog at www.luv2code.com and follow him on Twitter @darbyluvs2code.

Acknowledgments

Writing a book is a huge task. It's not the same as writing a blog or a review. It requires a lot of commitment right until the end. The difference in the time zones in which the team and I are located made it a little harder to communicate, but we managed quite well.

I was helped by several people in this project and would like to take the opportunity to thank them here.

First, I would like to thank Steve Anglin for having faith in me when he decided to sign me up for this book. I hope you feel that this faith was well placed. I would also like to thank Corbin Collins, Christine Ricketts, and Kate Blackham for putting up with the delays and giving me a gentle nudge to meet the deadlines, as well as their amazing work on this book.

On the more technical side, I would like to thank my tech reviewers Chád Darby and Yosun Chang for their invaluable input. I would also like to thank Tobias Domhan for writing the excellent AndAR library, the development of which will be continued by both of us from now on.

Finally, I would like to thank my family for their support, particularly for patience while I ignored them while working on this book.

Without all of these people, you would not be reading this book today.

–Raghav Sood

Introduction

Augmented reality is relatively recent development in the field of mobile computing. Despite its young age, it is already one of the fastest growing areas in this industry. Companies are investing lots of money in developing products that use augmented reality, the most notable of which is Google's Project Glass. Most people perceive augmented reality as hard to implement. That's a misconception. Like with any good app, good augmented reality apps will take some amount of effort to write. All you need to do is keep an open mind before diving in.

Who This Book Is For

This book is aimed at people who want to write apps employing augmented reality for the Android platform by Google. The book expects familiarity with the Java language and knowledge of the very basics of Android. However, an effort has been made to ensure that even people without such experience can understand the content and code. Hopefully, by the time you're done with this book, you'll know how to write amazing and rich Android apps that use the power of augmented reality.

How This Book Is Structured

This book is divided into nine chapters. We start with a basic introduction to augmented reality and move up through more and more complex features as we go. In Chapter 5, we take a look at dealing with the common errors that can happen in an augmented reality app. After that, we have four example apps that show use how to make increasingly complex augmented reality applications. A more detailed structure is given here:

- **Chapter 1:** This chapter gives you an idea of what augmented reality really is. It has several examples of how augmented reality has been used throughout the world, along with a short list of potential future applications.
- **Chapter 2:** This chapter guides you through writing a simple augmented reality app that consists of the four main features an augmented reality app usually uses. By the end of this chapter, you will have a skeleton structure that can be extended into any augmented reality application.

- **Chapter 3:** In this chapter, you are introduced to some of augmented reality's most important features: overlays and markers. In the span of two example apps, we cover using standard Android widgets as overlays as well as using the open source AndAR library to add marker recognition to our app.
- **Chapter 4:** The fourth chapter introduces the concept of artificial horizons by using a nonaugmented reality app. Then a second app is written that utilizes artificial horizons in an augmented reality app.
- **Chapter 5:** This chapter talks about the most common errors found while making an augmented reality app and also provides solutions for them. In addition to the errors, it also talks about other problems that don't result in an error, but still manage to stop your app from functioning as intended.
- **Chapter 6:** In this chapter, we write the first of our four example apps. It is an extremely simple AR app that provides basic information about the user's current location as well as plotting it on a map.
- **Chapter 7:** This chapter shows you how to extend the example app from Chapter 6 into a proper app that can be used to allow the user to navigate from his/her current location to one set on the map by the user.
- **Chapter 8:** This chapter shows you how to write an augmented reality model viewer using the AndAR library that allows you to display 3D models on a marker.
- **Chapter 9:** The last chapter of this book demonstrates how to write the most complex app of all: an augmented reality world browser that shows data from Wikipedia and Twitter all around you.

Prerequisites

This book contains some fairly advanced code, and it is assumed that you are familiar with the following:

- Java programming language
- Basic object-oriented concepts
- Android platform (moderate knowledge)
- Eclipse IDE basics

While it is not an absolute requirement to have all these prerequisites, it is highly recommended. You will absolutely need an Android device to test your apps on because many of the features used in the apps are not available on the Android emulator.

Downloading the Code

The code for the examples shown in this book is available on the Apress web site, www.apress.com/9781430239451. A link can be found on the book's information page under the Source Code/Downloads tab. This tab is located underneath the Related Titles section of the page.

You can also get the source code from this book's GitHub repository at <http://github.com/RaghavSood/ProAndroidAugmentedReality>.

In case you find a bug in our code, please file an issue for it at the GitHub repository, or directly contact the author via the means given below.

Contacting the Author

In case you have any questions, comments, or suggestions, or even find an error in this book, feel free to contact the author at raghavsood@appaholics.in via e-mail or via Twitter at [@Appaholics16](https://twitter.com/Appaholics16).

Applications of Augmented Reality

Augmented reality (AR) is a reasonably recent, but still large field. It does not have a very large market share, and most of its current applications are just out of prototyping. This makes AR a very anticipated and untapped niche. There are very few applications that implement AR technology in the Android Market right now. This chapter describes the real-world applications of AR, gives examples (along with images where possible), and discusses whether it is now possible to implement AR in the Android platform.

Augmented Reality vs. Virtual Reality

Augmented reality (AR) and virtual reality (VR) are fields in which the lines of distinction are kind of blurred. To put it another way, you can think of VR as the precursor to AR, with some parts overlapping in both. The main difference between the two technologies is that VR does not use a camera feed. All the things displayed in VR are either animations or prerecorded bits of film.

Current Uses

Despite being a relatively new field, there are enough AR apps available to allow us to make categories out of them. Here we take a look at what has already been implemented in the world of AR.

Casual Users

There are hundreds of apps that use AR that are meant to be used by the average person. They come in many types—for example, games, world browsers, and navigation apps. They are usually using the accelerometer and the GPS to obtain location and the physical state of the device. These apps are meant to be enjoyed and useful. One of the winning apps of the Android Developer Challenge 2 was an AR game: *SpecTrek*. The game uses your GPS to find your location and then prepares ghosts for you to hunt in surrounding areas. The game also has a map on which ghosts are displayed as markers on a Google map. During gameplay, the ghost is added as an overlay over the camera image.

On the other side of things, navigation apps have code to recognize roads and turnings, and mark out the route with arrows. This process is not as easy as it sounds, but is often done today.

In the end, world browsers are probably the most complex of all the casual apps that are widely used. They need several back-end databases and also need a lot of on-the-spot information from several sensors. After all, browsers still have to put everything together and display a set of icons on the screen. Almost every app you see on the market, whether AR or not, looks simple at first sight. But if you delve into the code and back ends, you will realize that most of them are in fact, very very complex and take a long time to create.

The best examples of casual AR apps are *SpecTrek* and *Wikitude*. Together, these apps make use of practically everything you can use to make an AR app on the Android platform. I highly recommend that you install them and become familiar with the features of AR on Android.

Most apps in this category can be implemented on the Android platform. In several cases, they do not even use all the sensors. Some of them can get quite complex. Figure 1-1 and Figure 1-2 show screenshots from *SpecTrek*.



Figure 1-1. Screenshot of SpecTrek

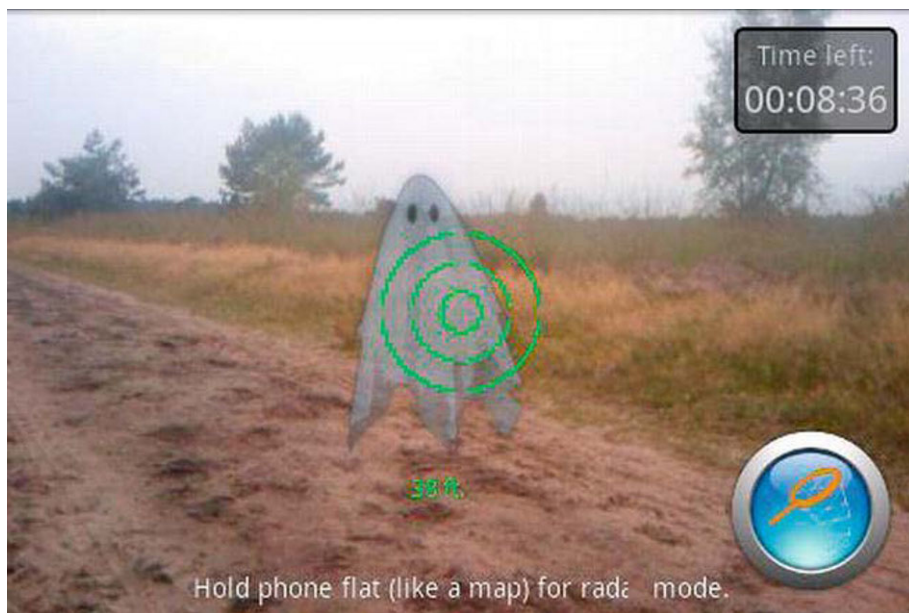


Figure 1-2. Another screenshot of SpecTrek

Military and Law Enforcement

Uses by military and law enforcement agencies are much more complex and technologically advanced. They range from AR goggles to full simulators designed to help in training. The military and some law enforcement agencies have simulators that make use of AR technology. A wide screen inside a room or a vehicle on which various scenarios is presented, and the trainee must decide the best course of action.

Some advanced Special Forces teams have basic AR goggles that, along with the land in sight, display information such as altitude, angle of viewing, light intensity, and so on. This information is calculated on the spot with mathematical formulas as these goggles do not come equipped with Internet connections.

Specialized night vision goggles come with AR technology as well. These goggles display location and other information, along with trying to fill in gaps that could not be illuminated by the night vision goggles themselves.

Almost all the unmanned vehicles implement AR as well. These vehicles, especially the aerial ones, can be thousands of kilometers away from their operators. These vehicles have one or more cameras mounted on their exterior, which transmit video to their operator. Most of these vehicles come equipped with several sensors as well. The sensor data is sent to the operator along with the video. This data is then processed and augmented over the video. Algorithms on the operator's system process the video and then pick out and mark buildings or objects of interest. All this is displayed as an overlay on the video.

These kinds of apps are quite difficult to implement on Android devices because of two main issues:

- Low processing power (Though with the recent release of the HTC One X and Samsung Galaxy S3, quad core phones released in May 2012, this is not so much of a problem.)
- Lack of more input devices and sensors

Vehicles

As of late, vehicles have started implementing AR technology. The windscreens have been replaced with large, wide, and high-definition displays. Often there are multiple screens in the vehicle, each showing a particular direction. If there is only one screen and multiple cameras, the vehicle will either switch the feed automatically or have the option for the user to do so. The exterior of the vehicle

has several cameras, facing multiple directions. The images on the screen are overlaid with useful data such as a small map, compass, direction arrows, alternate routes, weather forecast, and much more. This kind of technology is currently most visible in airplanes and trains at the moment. Smart cars with such technology are being tested out for the market. Submarines and ships are using this technology as well. The recently discontinued Space Shuttles had this kind of AR technology as well.

These apps can be implemented in a sort of hybrid way on the Android platform. Because most Android devices seem to be lacking in features that normal vehicles have, the same kind of features are not achieved. On the other hand, apps can be written that help with navigation by using the GPS to get the location; use direction APIs to get, well, the directions; and use the accelerometer to help with acquiring the speed of the vehicle. The Android device provides the AR power, and the vehicle provides the vehicle part.

Medical

AR-enabled surgeries are becoming more common these days. Surgeries done this way have a smaller error rate because the computer provides valuable inputs on the surgery and uses the information to control robots to perform some or all of the surgery. The computer can often provide alternatives and instructions on what can be done to improve the surgery in real time. The AR stream, along with other data, can also be sent to remote doctors, who can view the information of the patient as if the patient were in front of them.

There are also other medical applications of AR technology. AR machines can be used to monitor a large number of patients and make sure that their vital signs are under observation at all times.

This kind of AR technology has never been implemented on the Android platform because of several reasons:

- It would require an immense amount of information on the device because Internet connections are not yet reliable enough to risk a patient's life.
- The processing power required for some of these medical tasks is currently not available on the devices.
- There is not a very large market for Android devices in surgery and to help with medical tasks.

To top all this off, it is currently very difficult and expensive to design and build such an app. The AI algorithms needed to allow real-time AR work in the

medical field are yet to come into existence. Apart from that, you would require a team of very good developers, a team of highly skilled and experienced doctors, and a large amount of money.

Trial Rooms

In several shops, AR is being tried out as a virtual trial room. The user can stand in front of a screen with a camera mounted somewhere. The user will see himself displayed on the screen. The user then uses an input device such as a mouse or keyboard to select any of the available clothing options. The computer will then augment that item onto the user's image and display it on the screen. The user can turn to view himself from all angles.

These apps can be written for the Android platform in principle, but nobody has done it for lack of interest, and probably for lack of any idea as to why someone would want this. Actually apps in the genre have been made, but they are used for entertainment and modifying the facial features of people virtually.

Tourism

Tourism has received some part of the AR magic as well. At several famous spots around the world, organized tours now offer a head-mounted AR system that displays information about the current site and its buildings when you look at it. With AR, tourists can rebuild buildings, cities, landscapes, and terrains as they existed in the past. Tourism AR is also a built-in part of most world browsing applications because they provide markers to famous monuments. Tourism AR is not limited to historical places. It can be used to find parks, restaurants, hotels, and other tourist-related sites and attractions in an unfamiliar city. While not in very widespread use, it has grown exponentially over the past few years.

Features of these apps are already present in world browsers, but have a small back end of information to display. Nobody has yet implemented a complete version of any one city that can provide the required information.

Architecture

There are many camera-equipped machines that can generate a blueprint from an existing structure or display a virtual structure from the blueprints on the proposed site of construction. These speed up architectural work and help to design and check buildings. AR can also simulate natural disaster conditions and show how the building structure will react under that kind of pressure.

Apps in this segment can be written to an extent on Android. The ones that create blueprints out of the view of a room have already been written for the iOS platform and can be written for Android. The ones that display virtual models on a building scale are a little more difficult, but still feasible, as long as the models to be augmented can fit within the size constraints of the Android process and the device's RAM.

Assembly Lines

AR technology helps out a lot on various assembly lines, whether you are assembling cars, planes, mobiles, or anything else. Preprogrammed head goggles can provide step-by-step instructions on how to assemble it.

These apps can be written for Android, as long as the assembly process can incorporate markers at each step that requires instructions to be augmented. The information can be stored on a remote backend in this case.

Cinema/Performance

AR technology has been used to enhance movies and plays by having a static background and a screen with overlays on it to produce images and scenery that would otherwise require expensive and highly detailed sets.

This is a really feasible option. All you need to do is acquire the footage or background information for the performance, place markers at appropriate places, and augment the footage or background when needed.

Entertainment

In several amusement parks around the world, AR technology is being used to make rides that fit within a single room and manage to give you the experience of a whole ride. You will be made to sit in a car or some other vehicle that is mounted on hydraulics. You are surrounded on all sides by massive screens on which the whole scenery is displayed. Depending on whether the scenery is from a live camera or is animated, this could fall under both VR and AR. The vehicle moves in the air as the virtual track progresses. If the track is going down, the vehicle will tilt downward, and you will actually feel as if you are moving down. To provide a more realistic experience, the AR technology is coupled with some fans or water-spraying equipment.

It is possible to implement this on Android, but there are a few limitations. To have a completely immersive experience, you will need a large screen. Some of

the tablets might provide sufficient space to have a good experience, but implementing it for phones is a little too optimistic. Additionally, hydraulic mounted vehicles are used in the actual rides to provide the complete experience of movement. To compensate, some innovative thinking will be required on your part.

Education

AR technology has been successfully used in various educational institutes to act as add-ons to the textbook material or as a virtual, 3d textbook in itself. Normally done with head mounts the AR experience allows the students to “relive” events as they are known to have happened, while never leaving their class.

These apps can be implemented on the Android platform, but you need the backing of some course material provider. Apps like these also have the potential to push AR to the forefront because they have a very large potential user base.

Art

AR technology can and has been used to help create paintings, models and other forms of art. It has also helped disabled people realize their creative talent. AR is also used widely to try out a particular design, before actually putting it down in ink or carving it out of stone. Paintings can, for example, be painted virtually to see how they turn out, be refined until the artist is happy with them, and then be put down on the canvas finally.

These kinds of apps are possible as well. They will need to have several fine art-related features and will most likely make little use of the sensors available. The device should ideally have a high-resolution screen, coupled with a high-resolution camera.

Translation

AR-enabled devices are being used to translate text from multiple languages all over the world. These devices feature OCR and either have an entire cross-language dictionary on the device or translate the language over the Internet.

These apps are already in production. You would need to either write or use a ready-made optical character recognition (OCR) library to convert the images from the camera to text. After you have extracted the text from the images, you

can either use an on device translation dictionary, which would have to be bundled with the app, or translate it over the Internet and display the results.

Weather Forecasting

On practically every news channel a weather forecaster forecasts the weather on a map of the world behind him. In reality, most of these apps are augmented. The forecaster stands in front of a massive green backdrop. While recording, the green backdrop serves as a marker. After the recording is done, a computer is used to add the map and position it to match the forecaster's actions. If the forecast is being transmitted live to the viewers, the map is added as the forecast is transmitted.

Television

AR can be found in daily life as well. Many game shows, especially the ones with the questions, augment this information over the video of the players. Even in live sports matches, the score and other game-relevant information is augmented over the video and sent to the viewers. The slightly more annoying advertisements are augmented, too.

Many apps that provide live streams of sports matches currently implement this.

Astronomy

There are many apps that are useful to astronomers and good fun for everyone else. These apps can display the location of stars and constellations during the day or on a foggy night and do it in (more or less) real time.

Other

There are many, many more uses of AR that cannot be categorized so easily. They are mostly still in the designing and planning stages, but have the potential to forward AR technology to the forefront of daily gadgets.

Future Uses

As the previous section discussed, AR is quite well known and has enough apps available to make it noteworthy. However, there are some amazing uses for the technology that cannot be implemented right now due to limitations in hardware and algorithms.

Virtual Experiences

In the future, AR technology could be used to create virtual experiences. You could have a head mounted system that could transform your current location into something completely different. For example, you could live through movies by wearing such a system and seeing the movie happen around you. You could convert your house into a medieval castle or into the international space station. Coupled with aural AR and some smell-emitting technology, a whole experience could be made lifelike and feel completely real. In addition to this, wearing a body suit that can emulate the sense of touch will make it absolutely and undeniably real.

This would be quite difficult to implement on Android if and when it turns up because Android is lacking in the required sensors and input methods to implement such a thing. Its visual features could be implemented to an extent, but the sound and feeling ones would be out of reach unless someone creates a bodysuit with a head mounted display and sound on a ported version of Android.

Impossible Simulations

AR technology could do what real hardware cannot, at least as of now. You could have a screen on which you have an ordinary object such as a cube. You could then apply various scenarios and forces to this cube and see how it turns out. You would not be able to do this with real hardware because real hardware usually cannot change shape without being destroyed. You could also test theories using experiments that would otherwise be extremely expensive or completely impossible.

This may be possible to implement on Android by the time other real-world models are developed because the only hard requirement for high-end simulations is the data and a large amount of processing power. At the rate the power of mobile phones is increasing, they could become fast enough to run such apps.

Holograms

AR allows the user to have a live direct or indirect view of the world, which might enable users to have holograms in front of them. These holograms could be interactive or merely descriptive. They could be showing anything.

This could be done even today with a highly modified version of an app that uses markers to display models. Instead of static models, the app could be made to display an animation or recording or live transmission. However this would not provide a true hologram experience as it will be on the device's screen only.

Video Conferencing

AR could allow multiple people to appear in the same conference room if a video feed of a conference room is transmitted to them. The people could use a webcam to “appear” in the seats of the room, along with the others. This could create a collaborative environment, even if the collaborators were thousands of kilometers apart.

This app could be implemented with some advanced placement algorithms and a high-speed Internet connection. You would need the algorithms because it is unlikely that the people taking part in the conference will stay in exactly the same place throughout. You would need to keep positioning them again and again so that they do not overlap with the other people.

Movies

AR could be used to play entire movies. The theatre could be replaced with the background of the movie or the theatre could be replaced with the actors only. In the first way, the actors could be augmented onto the background and in the second method the background could be augmented behind the actors. These could provide for more realistic and fun movies, while keeping the cost of shooting down.

Apps like these are already in production, but not in the quality, popularity, and sophistication to have me drag this out of the future implementations. Although these apps are not that easy to make, they're not very difficult, either.

Gesture Control

AR could be used to implement many gesture controls such as eye dialing. The camera could track the user's eye movement to select the appropriate number key. After the desired key has been selected, the user could blink to press that number and then proceed to select the next key. This could similarly be implemented to control music players, mobile apps, computers, and other forms of technology.

These kinds of apps would require a few things:

- A front camera with a reasonable resolution
- Well written algorithms to detect fine eye movements and to be able to distinguish them from other movements, such as checking a side view mirror

AR has come a long way from its beginnings and has a long way to go. Its basic requirements of a camera, GPS, accelerometer, and compass are fulfilled by almost every Android device on the market. Although apps that use AR technology exist for the Android platform, they are few in number compared with the other kinds of apps. It is a great time to enter the Android platform by making AR apps because the competition is good enough to drive user interest to these apps, but not fierce enough to drive you out of business yet. Considering the relatively few AR apps on the market, there is also a good chance that if you come up with a good AR app it will have no more than 3–5 competing apps, giving you a great advantage. In the next chapter, the basics of AR apps on Android are explained, and a basic app is developed.

Summary

That concludes our look at the current and future uses of AR and their implementation (or likely implementation) on the Android platform. The next chapter looks at the basics of creating an AR app on Android.

Basics of Augmented Reality on the Android Platform

By now, you have a basic idea of what augmented reality (AR) is, what is being done with it around the world, and what you can do with it on an Android device. This chapter will launch you into the world of AR on Android and teach you the basics of it. To aid in your understanding of everything done here (and elsewhere) in this book, we will create apps that demonstrate what is being taught as we move along. This chapter will focus on making a basic app that contains the four main parts of any advanced AR app: the camera, GPS, accelerometer, and compass.

Creating the App

This is a really simple app. It has no overlays and no actual use for any of the data it is receiving from the GPS, compass, camera, and accelerometer. In the next chapter, we will build on this app and add overlays to it.

First, we need to create a new project. In the package name, I am using `com.paar.ch2`. You can use any name that suits you, but make sure to change any references in the code here to match your package name. The project should be set to support Android 2.1 as the minimum. I am building the project against Android 4.0 (Ice Cream Sandwich), but you can choose your own target.