Extracted from:

Augmented Reality

A Practical Guide

Chapter 1

About Augmented Reality

1.1 What Is Augmented Reality?

Wouldn't it be cool to walk around a level of your favorite video game and battle life-size game characters? For decades, video game developers have yearned to make their games more immersive, and when you consider the difference between Spacewar and Halo, it's obvious that a great deal of progress has been made. However, the typical video game is still played with the gamer sitting statically on a couch while staring at a screen.

Video games don't have to be so passive. Although the Nintendo Wii has helped turn video games into aerobic workouts, games don't even need to be played indoors. Freeing gamers from their living rooms is just one application of the burgeoning technology called *augmented reality* (AR).

The ambitious goal of AR is to create the sensation that virtual objects are present in the real world. To achieve the effect, software combines *virtual reality* (VR) elements with the real world. Obviously, AR is most effective when virtual elements are added in real time. Because of this, AR commonly involves augmenting 2D or 3D objects to a real-time digital video image. The simplest example of visual AR is overlaying a 2D image on digital video. However, it is also possible to add 3D objects—they can be rendered so that they appear to belong to a scene containing real 3D objects. Generally speaking, adding a 3D object to real-time video makes for a more impressive demonstration of AR technology.

When virtual objects are added to a scene, it is known as *visual AR*. By definition, AR elements are not visible to the naked eye, so visual AR relies upon some sort of display. This display can be as simple as a computer monitor or a television, or it could be something more advanced—such as a see-through eyepiece on a *head-mounted display* (HMD). New options are becoming available as many AR researchers are focusing their efforts on interfaces such as handheld devices, webcams, and more advanced HMDs.

This book introduces and explains AR to software hobbyists. It gives you all the information you will need to quickly start developing your own AR applications, and the final two chapters show you how to create your own AR video game. To get you up and running as fast as possible, sample code and compiled versions of the samples are provided. Furthermore, to make the examples as accessible as possible, code is provided in C++, and one sample is ported to C#. The compiled samples include versions for Linux, Mac OS X, and Windows. Want to dig deeper? You will also find detailed explanations of AR technology—even the math will be explained.

Using AR, it is possible to achieve a level of immersion that is beyond what most people associate with video games. However, this is not the only application of AR technology. Granted, few people these days have access to the HMDs that allow for truly immersive experiences, but it is possible to create AR applications using readily available hardware. In this book, we will show you how to create your own AR applications using nothing more than a PC and a USB 2.0 webcam.

1.2 The Origins

In 1968, Ivan Sutherland created a working prototype of what is widely considered to be the first VR system and the first AR system. Although it used simple wire-frame graphics, the project was the genesis of AR. Sutherland wrote about his work in a Harvard University paper entitled "A Head-Mounted Three-Dimensional Display" (http://www.aec.at/en/archiv_files/19902/E1990b_123.p Sutherland's system required that the user wear a cumbersome HMD that was so heavy it had to be suspended from the ceiling. Based on this, the device was nicknamed the Sword of Damocles.

Sutherland recognized that his interface was limited, so he continued to work on better systems. A few years after his first VR system, Sutherland wrote a paper called "The Ultimate Display." Sutherland's dream interface was described in this paper, and to any Star Trek fan his description would conjure up images of the famous holodeck. He wrote, "The ultimate display would, of course, be a room within

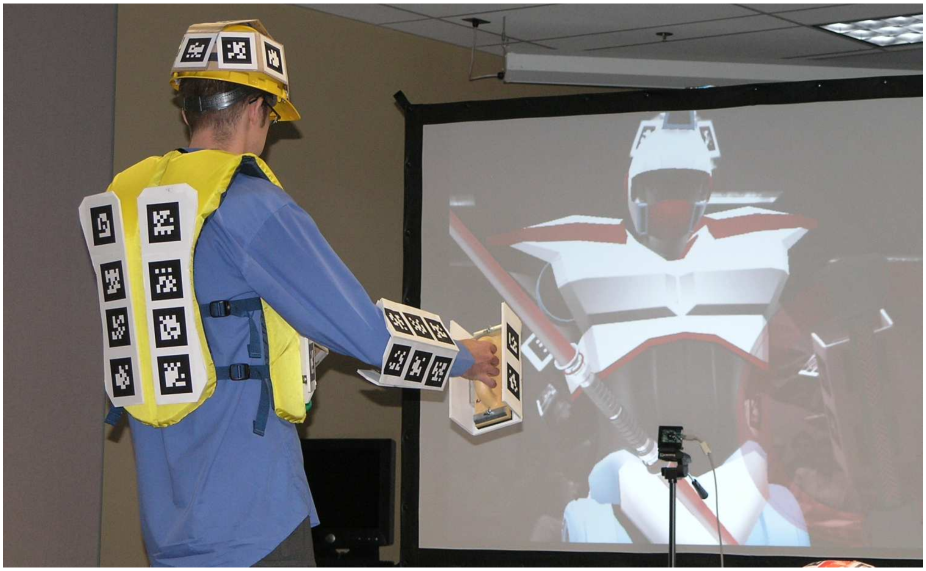


Figure 1.1: A large magic mirror

which the computer can control the existence of matter. A chair displayed in such a room would be good enough to sit in. Handcuffs displayed in such a room would be confining, and a bullet displayed in such a room would be fatal. With appropriate programming such a display could literally be the Wonderland into which Alice walked" (http://www.informatik.umu.se/~jwworth/TheUltimateDisplay.pdf).

1.3 Magic Lens vs. Magic Mirror

The two common paradigms for AR are the *magic mirror* and the *magic lens*. The magic mirror technique involves putting a computer monitor, or a television, behind the area that is being captured by an AR video camera. The display is the *mirror*, and it often displays the augmentations via real-time video. If you have a large display (for example, a projection screen), you can stand in front of a magic mirror and enjoy some full-scale augmentation (see Figure 1.1). For example, you might be a hockey goalie blocking shots from virtual shooters.

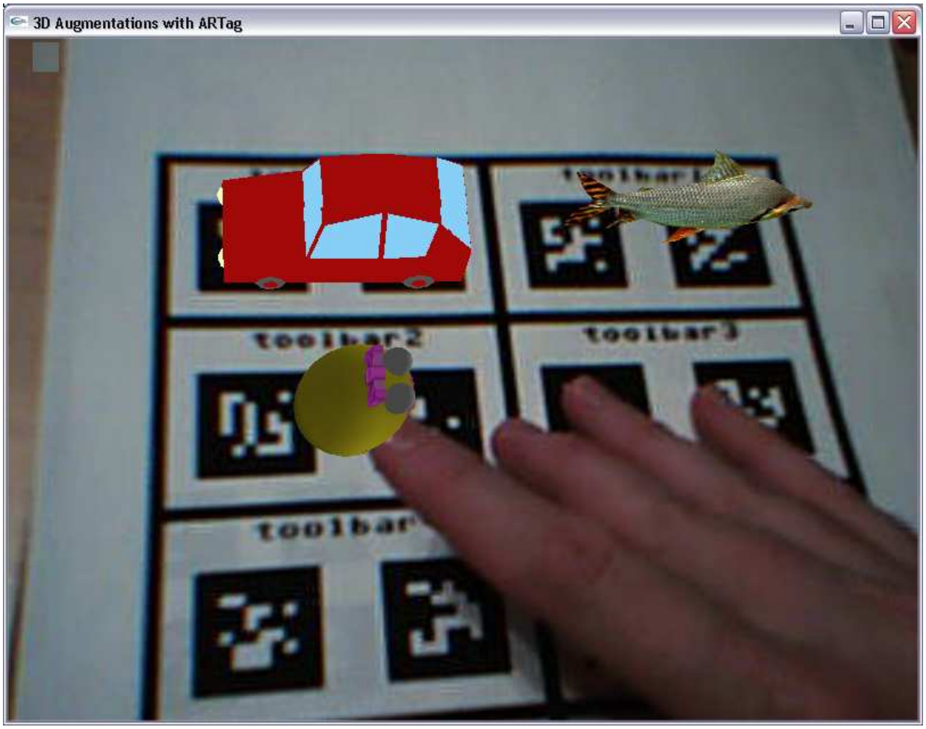


Figure 1.2: The magic lens perspective

The magic lens viewpoint is a completely different approach. Rather than offering the user a mirror to look at, the magic lens allows the user to *see through* to an image of the real world with added AR elements (see Figure 1.2). In this setup, the user has some sort of display in front of him; it could be as simple as a standard computer monitor or as complex as an HMD. In this book, the magic lens approach will be used for most of the examples.

1.4 Marker vs. Markerless AR

As you'll learn in Chapter 3, AR systems generally require some indication of where exactly they should augment a digital image. This is most commonly accomplished with *AR markers*. Various markers have been tried—everything from light-emitting diodes (LEDs) to a person's hand. But the simplest form of marker is a unique pattern that is visible to

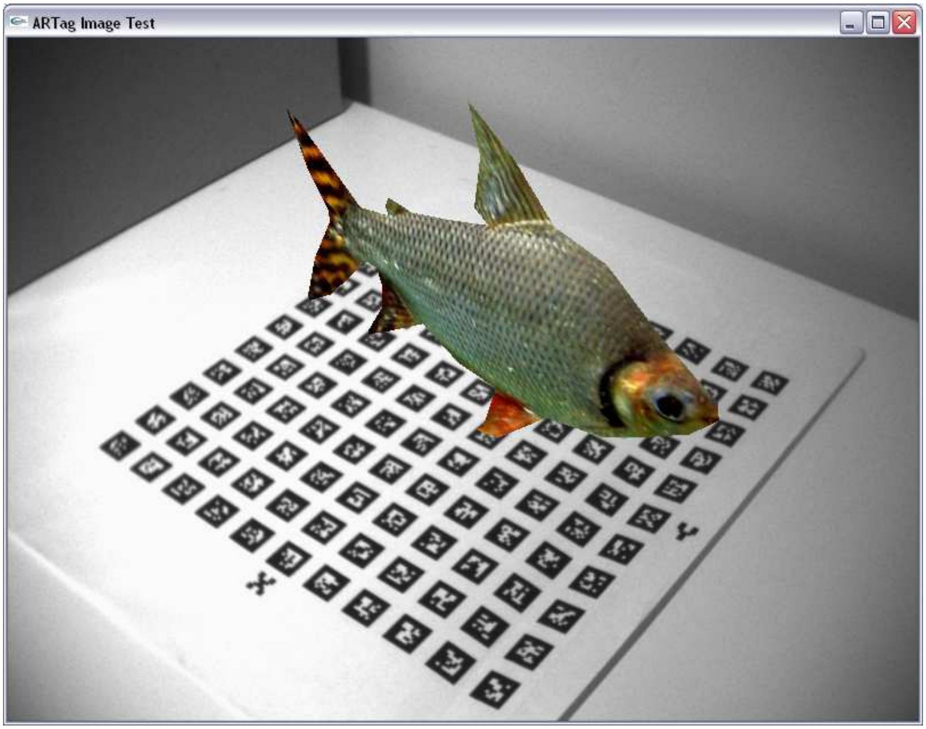


Figure 1.3: A 3D AR fish on a marker pattern

the AR camera and can be identified in the AR system software. These patterns are physically added to the real world. In Figure 1.3, the fish is a virtual 3D object, and the pattern below contains marker tags. The marker tags are used to determine the viewpoint of the real camera so that the virtual fish can be rendered appropriately.

To demonstrate AR technology, this book will use the ARTag marker system (http://www.artag.net). ARTag was developed by Dr. Mark Fiala, coauthor of this book and a researcher at the National Research Council of Canada (NRC) (http://www.iit-iti.nrc-cnrc.gc.ca/r-d/cv-vi_e.html). ARTag is a digitally generated and scientifically verified fiducial marker system for AR (see Figure 1.4, on the next page). *Fiducial* means the markers are used as a trusted reference.

ARTag is a computer vision software system that uses markers to align real and virtual cameras. ARTag recognizes special black-and-white square markers, finds the pose, and then sets the modelview matrix

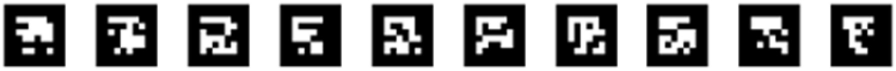


Figure 1.4: ARTag fiducial markers

so that subsequent rendering operations appear relative to the array and, therefore, relative to the real 3D world. In other words, the ARTag markers allow the software to calculate where to insert virtual elements so that they appear properly in the augmented image.

To use ARTag, marker patterns are printed and then placed where the user would like virtual elements to appear. ARTag has a *software development kit* (SDK) that includes a library of markers, the computer vision software to find them in images, and some code to help load and display 3D models. You will be learning more about how ARTag works later in the book. In fact, you'll probably find more information than you could ever want about AR markers.

Adding AR markers can be as simple as printing a marker pattern and then placing the printout in the field of a webcam image that's running through AR software. When the markers are recognized by AR software, the computer is able to determine the position and angle of the marker. The software can then use this information to extrapolate the correct position and orientation of a virtual object. When this computation is complete, the software overlays the virtual object on top of the original camera image, and the augmented version is shown to the user. If the AR software renders the object correctly, it will appear to be part of the real-world scene. Since the orientation is also calculated, the virtual object can be adjusted as the marker is moved around in 3D space.

It is possible, however, to create an AR effect without using markers; this is known as *markerless AR* and is ideal since it does not require the forethought of adding markers to a scene. A tracking system can instead track the position of LEDs, or reflective balls, with special equipment for placing virtual elements in a digital image. Such systems are often seen in university VR labs. Markerless tracking, however, is understood in the AR community to mean not adding anything to the environment. Instead, "natural" features such as the corner of a window or a distinctive painting on a wall would be used.

In the future, markerless AR will probably turn out to be the preferred method of AR. However, markerless AR has not yet advanced to the point where it's possible to provide a simple way for the public to use the technology, and markerless systems that perform as well as marker-based, or LED-based, systems are not yet available. For this reason, this book will use an AR marker system.

1.5 Examples of Applied AR

AR is rapidly gaining popularity, but it is still a fledgling technology. Most of the current applications are academic, but there are some commercial products on the market. These applications include everything from factory floor design to storytelling. Based on the wide breadth of these uses, it's clear that AR has a promising future. The next section highlights some examples of contemporary AR projects.

ARQuake

ARQuake (http://wearables.unisa.edu.au/arquake) is a virtual reality game using the game engine from id Software's classic first-person shooter, Quake. The game is an excellent example of how AR can be used to immerse a person within an augmented environment. ARQuake gamers are able to freely move around the physical space used for each level (see Figure 1.5, on the following page). To all those gamers who have been told to "go play outside," the ARQuake system may be the answer to their digital dreams.

ARQuake began in 1998 as a project at the University of South Australia. The original team members included Dr. Bruce Thomas, Dr. Wayne Piekarski, Benjamin Close, John Donoghue, John Squires, and Phil De Bondi. Over the years, it has been featured in many news stories and academic papers. Today, the project is still going and is currently maintained by Piekarski, Close, and Thomas.

To display the AR elements, track the player, and control the game, ARQuake uses an HMD, a notebook computer, a head tracker, and a GPS system. To create the illusion of Quake monsters attacking the player, ARQuake adds virtual monsters to the player's viewpoint through the HMD. As the player moves his head, the game calculates which virtual monsters should appear in his HMD display.

ARQuake inspired commercially available AR game systems such as a_rage—the Augmented Reality Active Game Engine. The a_rage engine

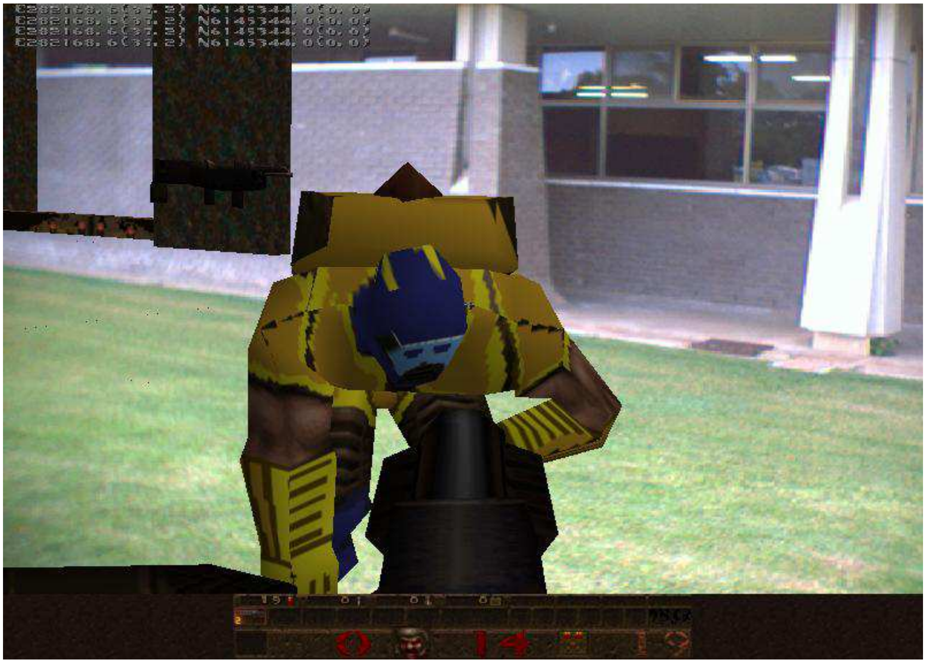


Figure 1.5: An ARQuake virtual monster

is still in its infancy, but games based on it have been featured at trade shows such as the Electronic Entertainment Expo (E3).

Looking toward the future, the company that makes a_rage held a contest to create a concept for its gaming hardware. The winning design is a sleeker version of the current ARQuake hardware (see Figure 1.6, on the next page). The days of cumbersome AR hardware are numbered.

BBC jam Storybooooks for Kids

In the past four years, the Creative Research and Development team (CR&D)—in the New Media division of the BBC—has funded a number of projects meant to explore the potential of AR. In 2006, roughly 500 people registered and took part in the first public trial.

This trial was run with *BBC jam*, which is a new online learning service developed to complement the school curriculums across the United Kingdom (http://www.bbc.co.uk/jam). The stated aim of the BBC jam trial was to use AR to create interactive stories suitable for children aged five

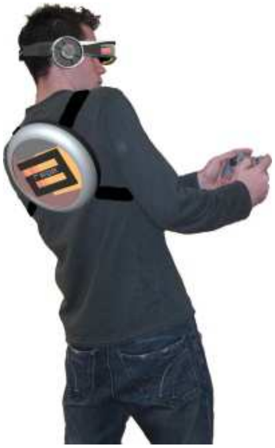


Figure 1.6: a_rage concept hardware

to seven. Three AR *storypacks* were offered for download. They were designed to be used with a standard PC and a USB webcam. Using the storypacks, young readers could use AR to explore and interact with scenes in a 3D *virtual pop-up book* (see Figure 1.7, on the following page). The jam AR storypacks employ a specially designed on-screen interface, narration, linked HTML help pages, and specially designed story scene printouts.

The jam project was evaluated by external researchers from London Knowledge Lab, and the CR&D team is now planning a second round of trials for other age groups and other subjects. Other AR activity within CR&D includes work with secondary-school teachers and pupils to develop interactive science visualizations in a user-centered and collaborative design process.

The Invisible Train

The Invisible Train (http://studierstube.icg.tu-graz.ac.at/invisible_train/) is a handheld multiplayer AR game developed by Daniel Wagner, Thomas Pintaric, and Dieter Schmalstieg at the Vienna University of Technology. In the game, the railroad is represented by a miniature wooden track, but the trains are entirely virtual (see Figure 1.8, on page 21).

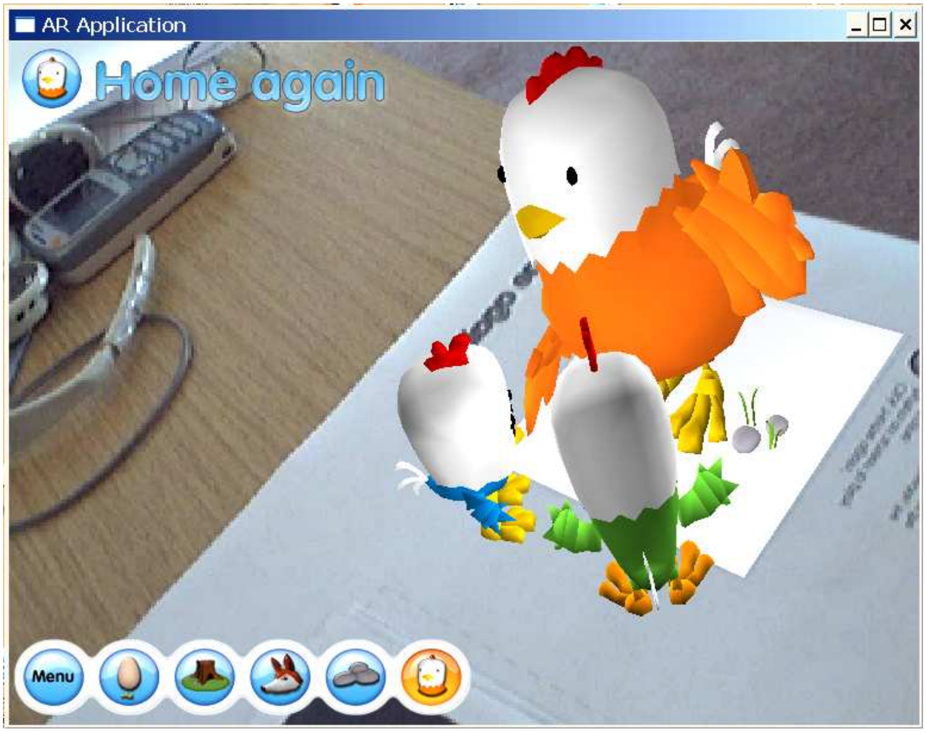


Figure 1.7: BBC jam characters make it home.

The goal of the Invisible Train game is for players to avoid crashing their trains into one another. The participants use personal digital assistants (PDAs) to view the virtual trains, control train speed, and change the position of railroad switches. Cameras on the PDAs are used to capture the table and view the virtual trains as they move around the track. Using PDAs in this manner is an example of the magic lens metaphor for AR.

Although it would be possible to use HMDs with this game, the creators thought PDAs would be better for their application. For example, PDAs are far less cumbersome, and they are also much cheaper. Furthermore, the players don't have to worry about adjusting to any sort of unusual hardware—they simply point the PDA toward the track, and they can see the virtual trains on the PDA's screen.

The Invisible Train game has won numerous awards and has been showcased at conferences such as SIGGRAPH 2004, Ars Electronica

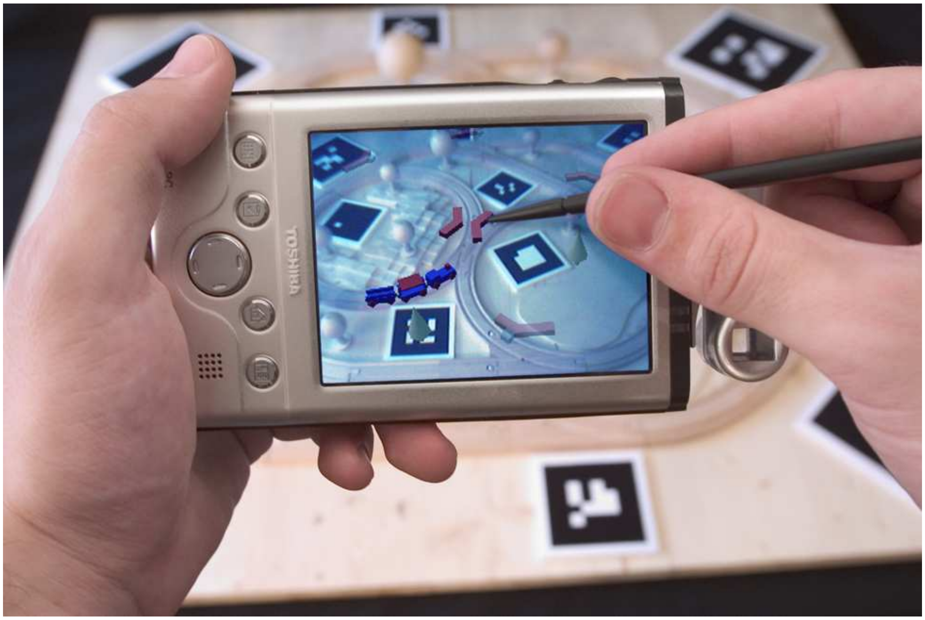


Figure 1.8: The Invisible Train game (image courtesy of Vienna University of Technology)

2004, IEEE Virtual Reality 2005, and Wired NextFest 2005. The AR marker system originally used for the Invisible Train was replaced by one inspired by ARTag—the marker system that will be discussed in great detail in this book.

AR Polygonal Modeling

AR polygonal modeling (ARpm) is a project by Peter Fiala at Purdue University. Using ARpm, the popular Autodesk 3D Studio Max (3DSMax) modeling software is converted into an augmented reality application. A user wears an HMD and works on a 3D object that appears to sit on the table in front of her. The usual working scene of 3D objects and the grids, labels, and other user interface items appear as virtual objects above an array of markers. Instead of the usual pull-down menus to access the 3DSMax tools, the menus are associated with tangible objects (*toolbar* marker arrays). The object shown in Figure 1.9, on the following page, was created entirely using the ARpm system.

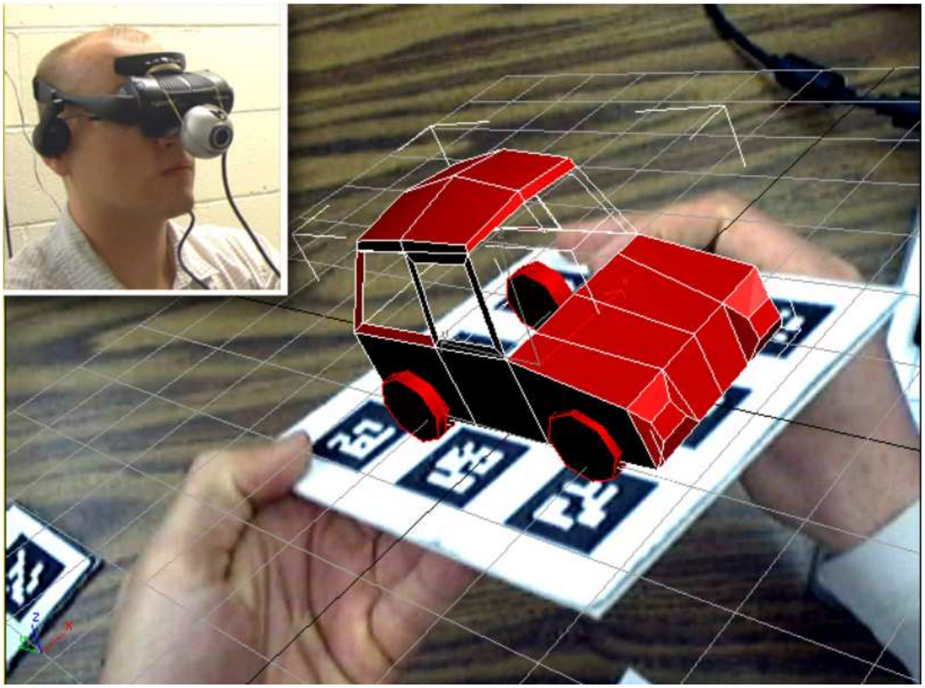


Figure 1.9: The ARpm system

The user operates ARpm with the following items: an HMD with a camera affixed (for video see-through AR), a 3D mouse, a 12" by 12" base array of markers and a stack of 2" by 4" cards. Each card has a small array of markers. The 3D mouse is not a specialized piece of hardware; it is a conventional wireless mouse attached to a rigid array of markers. In 3DSMax there are orthographic and perspective view panes (sections of the window) that can be resized. ARpm enlarges the perspective pane and mixes it with the video image to present it to the HMD. Through the HMD, the user can see the 3D object being created, and the standard grid lines appear to exist on top of the base array on the table in front of her. In 3DSMax, the actions of the mouse and mouse clicks depend on what mode the user is in. For example, there are modes such as "modifier," "sphere creation," and "vertex movement." Usually, in 3DSMax, one switches between modes by using hotkeys or pull-down menus. In ARpm, the user changes these modes by clicking her mouse in front of an icon hovering above each of the *toolbar* cards. The user can physically arrange the cards on her desktop, and thus she can place the controls where they are most convenient.

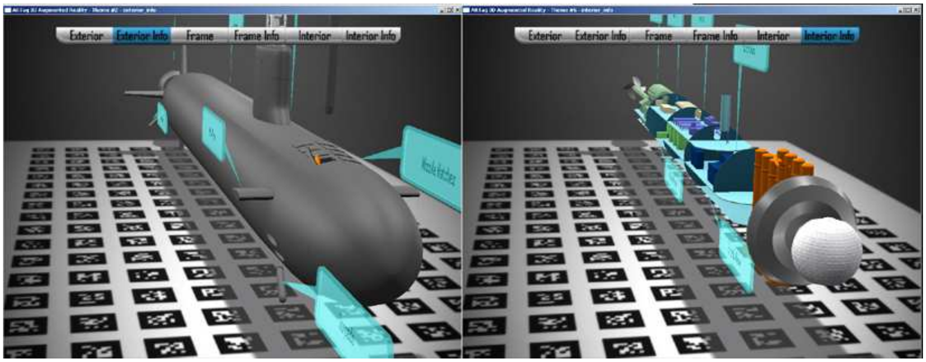


Figure 1.10: Two different views of a virtual sub

National Research Council of Canada Examples

AR technology can be used for a wide range of applications. The following sections include some AR examples that help present its potential for nongaming purposes. These programs are currently being tested by the NRC. By experimenting with various sorts of programs, researchers can explore potential applications of AR technology. You can download videos of some of these samples from http://www.artag.net/videos.html.

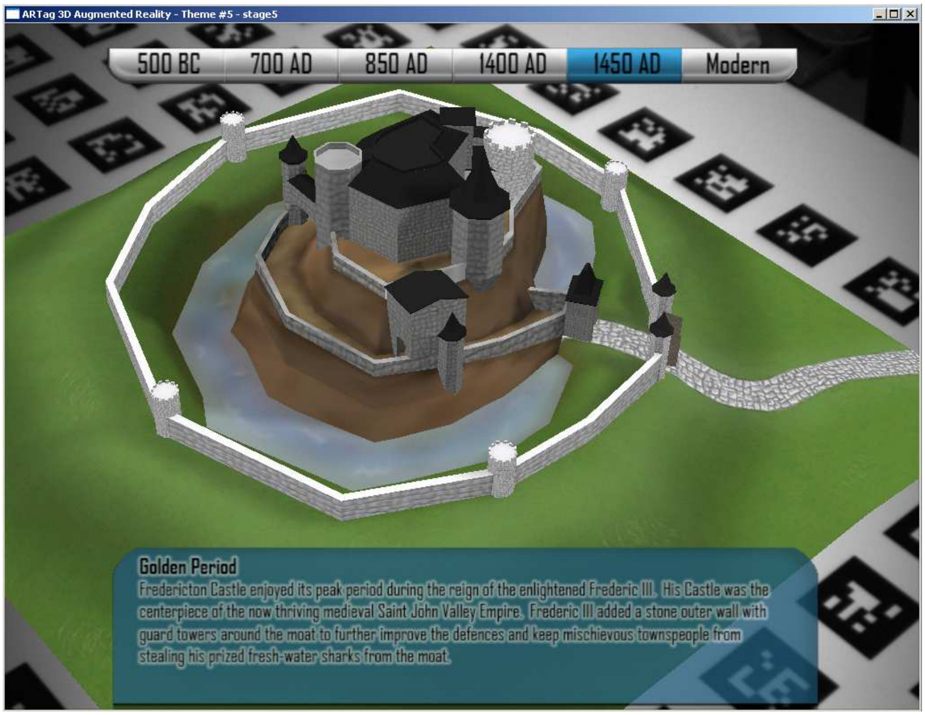
AR Submarine Views

This AR application shows various 3D submarine models. By using a mouse—or in this case, a table stylus—a user can switch between various views of the submarine. For example, the user can first look at an exterior view and then select Interior Info to see a detailed model of the inner components of the sub (see Figure 1.10).

This sort of application can be used as a powerful educational tool. You can imagine that this technology would be invaluable to anyone who needs to learn the inner workings of a machine. A similar sort of AR framework is used for the next two NRC examples.

Historical Education Example

In the submarine example, the user's viewpoint moved through space and allowed you to look at different slices of the sub. In this historical education example, you are able to move through time and see how a region might change.



Golden Period

Fredericton Castle enjoyed its peak period during the reign of the enlightened Frederic III. His Castle was the centerpiece of the now thriving medieval Saint John Valley Empire. Frederic III added a stone outer wall with guard towers around the moat to further improve the defences and keep mischievous townspeople from stealing his prized fresh-water sharks from the moat.

Figure 1.11: A castle circa 1450 AD

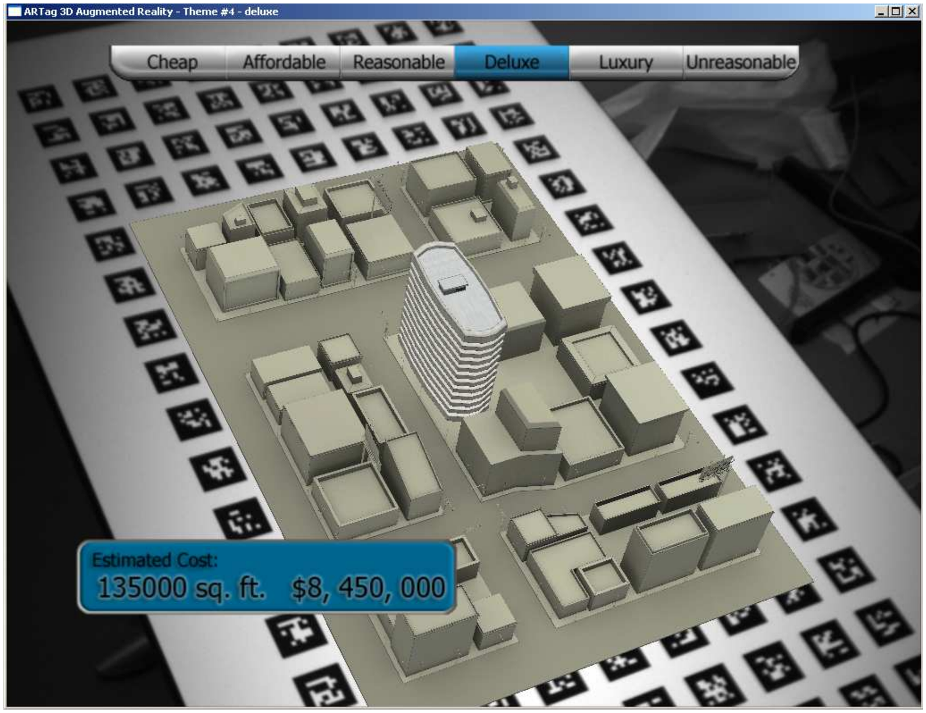
Specifically, you can see the same settlement as it was in different eras. The user can move around the table and see the various historical views from any angle.

You may begin with an early Celtic settlement (circa 500 BC), but with one click, you can progress 1,200 years to a wooden palisade (circa 700 AD). Another click takes you to about 1450 AD, when a castle is established at the same location (see Figure 1.11). Finally, a modern settlement is shown—complete with parking lot and fast food.

This sort of display is usually found in museums—using physical models. However, with AR, considerable space can be saved; for example, an entire museum's worth of displays could be presented on one table.

Architectural Demo Example

Not every demonstration has to be for educational purposes. A third example uses the same framework to show how an architectural firm



Estimated Cost:
135000 sq. ft. $8, 450, 000

Figure 1.12: Architectural mock-ups

might create mock-ups of various building designs for a client. Using this demonstration, the client can clearly see how each proposed high-rise would fit in with the existing structures. Amongst the options are Reasonable and Deluxe (see Figure 1.12). The final option is, comically, labeled Unreasonable.

Mobile AR

Technology Review (http://www.technologyreview.com) recently forecast that mobile AR will be one of the ten hottest technologies of 2008. This is based on the reasonable prediction that AR will appear more and more on handheld devices, especially camera cell phones. Not only are cameras becoming standard on cell phones, but they are also increasingly powerful. And everyone will have them; according to an article on Ars Technica (http://arstechnica.com/news.ars/post/20050720-5116.html), “Sales of cell phones will exceed 1 billion handsets each year in 2009, becoming the most common consumer electronics device in the world.”

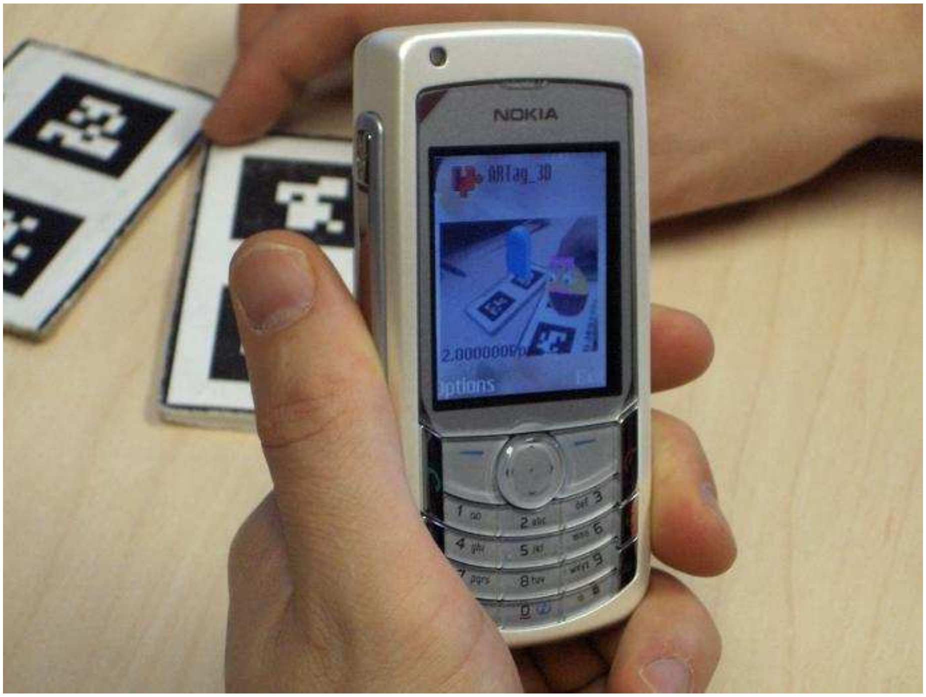


Figure 1.13: AR on a cell phone

What this means is that mobile handheld devices will have capabilities that make them ideal for AR. A visual AR platform needs at minimum a video input, a computer, and a display screen, but it can have increased potential with the addition of a network connection and other goodies such as a GPS receiver and orientation sensor. The Gizmondo was a short-lived handheld gaming console that had all of these; some high-end cell phones will soon have all these features.

One of the newest trials at the NRC is cell phone and PDA implementations of AR functionality. See Figure 1.13; it shows a cell phone running the same 3d_augmentations program that is used as a sample for this book.

The Pragmatic Bookshelf

The Pragmatic Bookshelf features books written by developers for developers. The titles continue the well-known Pragmatic Programmer style and continue to garner awards and rave reviews. As development gets more and more difficult, the Pragmatic Programmers will be there with more titles and products to help you stay on top of your game.

Visit Us Online

Augmented Reality's Home Page

http://pragprog.com/titles/cfar

Source code from this book, errata, and other resources. Come give us feedback, too!

Register for Updates

http://pragprog.com/updates

Be notified when updates and new books become available.

Join the Community

http://pragprog.com/community

Read our weblogs, join our online discussions, participate in our mailing list, interact with our wiki, and benefit from the experience of other Pragmatic Programmers.

New and Noteworthy

http://pragprog.com/news

Check out the latest pragmatic developments in the news.

Buy the Book

If you liked this PDF, perhaps you'd like to have a paper copy of the book. It's available for purchase at our store: pragprog.com/titles/cfar.

Contact Us

Phone Orders:	1-800-699-PROG (+1 919 847 3884)
Online Orders:	www.pragprog.com/catalog
Customer Service:	`orders@pragprog.com`
Non-English Versions:	`translations@pragprog.com`
Pragmatic Teaching:	`academic@pragprog.com`
Author Proposals:	`proposals@pragprog.com`