Technical Note: Augmented Reality Software Kits for Smart Phones

Alan Craig[1,2], Robert E. McGrath[1,2], Alejandro Gutierrez[3]
1. National Center for Supercomputing Applications (NCSA)
2. Institute for Computing in Humanities, Arts, and Social Science (I-CHASS)
3. Graduate School for Library and Information Science
University of Illinois, Urbana-Champaign
January, 2011
Revised April, 2011
Updated May, 2011

1. Introduction

As part of the project entitled "Augmented Reality for Understanding Social and Environmental Science" (NSF ARC-1025298), we are investigating the use of Augmented Reality on smart phones, such as iPhones or Android phones. Our initial work has examined available software development kits. This note summarizes our key findings to date.

The reader must be aware that this technology is evolving rapidly, with new devices and software appearing every month. Many details in this report will be out of date in the near future.

Our goal is to investigate "Smart phones", by which mean iPhone, Android, and others, along with similar handhelds (e.g., the iTouch) and tablets. These devices are of interest because they are rapidly becoming ubiquitous, and therefore have the potential to transform science, engineering, and technical education at relatively low cost.

We are investigating "Augmented Reality" (AR), which we define as live view of a physical, real-world environment augmented by computer-generated graphics (e.g., see [18, 24]). We are particularly interested in AR where the computer generated graphics are registered with objects in the physical world, so they can be manipulated, picked up, turned around, etc.

The investigations have been performed by a mixed team of students, including undergraduates participating in a summer REU project, a graduate research assistant, and the Co-I's.

1.1. Key Technical questions

In our investigations, we have considered several questions, including:

What AR software is available that works on smart phones?
• Free or commercial
• Usability (e.g., easy to compile, run, etc.)
What kinds of applications can be implemented?

What is the target architecture?
• Smart phone/handheld, PC, web, other?
What performance can be achieved on a smart phone?
How is content created?
• What kind of 3D models can be used
• How is 3D content loaded, etc.

We have investigated these questions by obtaining specific software, trying it out on several platforms (Microsoft Windows, Apple Macintosh, Linux, iPhone, Android). These investigations focus on practical issues, we have not attempted any formal evaluation, benchmarks, or comparative studies.

1.2. Example Applications

This project is investigating the use of Augmented Reality for a variety of applications, primarily in Humanities, Arts, and Social Sciences. This section sketches two examples we are exploring.

(Virtual) Archaeology

We imagine a classroom exercise in which students are to study an archeological site, to discover remains and reconstruct the original configuration and uses of the site. The students must choose where to dig, and carefully uncover each grid square. They must record what they find systematically, step by step. Each find must be interpreted (and visualized), and finally, the whole site must be interpreted. Interpreting the objects requires reconstructing them from remains, and visualizing their original appearance. Interpreting the site requires hypothesizing the overall meaning of all the artifacts.

AR can be used to realize this story as follows:

A local computer has a model of the site with artifacts at various locations and depths. Imagine a grid square, where each square has a card or deck of cards with AR markers. Each card in the deck represents a depth in the matrix, with the top card representing the undisturbed geological surface. When viewed with the AR system, each card will show what the students have discovered by excavating to a specific depth. The displays are 3D, so items may stick up from the surface (at different angles and amounts). Also, there may be "nothing" at a given depth, in which case, the view revealed is excavated dirt. The computer system will enforce realistic constraints so that only the exposed material can be seen. Even though one could "cheat" and pull a card from the bottom, the display will be blank (and may record demerits) if this is attempted. In addition, excavating should require effort and time. We will experiment with mechanisms to enforce this, such as requiring activity to be specified ("We will excavate this square now, using the following technique.") and recording each step in the excavation log.

In this case, the AR provides an inexpensive immersive experience, much more like a real excavation than a movie or computer game provides.

Historic Houses

In collaboration with the Illinois Program for Research in the Humanities at the University of Illinois at Urbana-Champaign (IPHR [8]) and the Society of Architectural Historians (SAH [15]), we are considering several uses for AR to enhance the documentation, interpretation, and study of historic buildings.

Handheld AR can be used to enhance printed materials such as books and pamphlets, creating 3D, interactive "pop ups" visible through a smart phone. This is particularly attractive for the study and appreciation of historic architecture, which is inherently three dimensional and only available at specific locations.

Augmented Reality might also be used to enhance the spaces within a historic site, providing alternative views via a smart phone. For example, the AR might portray a no longer existing feature in its actual setting, or to reveal otherwise hidden details, such as the interior of a container.

2. Investigation of Software Toolkits

We have investigated available AR software development toolkits that can be used to create these and similar applications. The ARToolkit is the most widely used AR development environment. It is available in free and commercial versions, and has been ported to many platforms, including iPhone, Android, and web services. Other toolkits are emerging from commercial companies, and some companies provide consultation and studio services. We discuss these alternatives in this section.

2.1 ARToolkit: PC and Laptop

The ARToolkit is the most popular freely available AR software development kit [7]. The free version was effectively abandoned in 2007 (circa Version 2.71.1). The free version has been used in a number of other projects discussed below.

The original developers of the ARToolkit now work commercially at ARToolworks [3]. The commercial versions or the ARToolkit (version 4.x, etc.) are up to date, higher performing, and have many features not available in the free version.

The ARToolkit implements the logic necessary for Augmented Reality, including:

- Recognizing markers in the video frames
- Determining the geometry of the scene (i.e., the position and angle of the marker)
- Projecting the 3D model into the scene
- Creating the composite video frame (the view plus the 3D model in registration with the marker).

These steps are performed for each frame of the video, and must be performed fast enough for the live video to be acceptable.
The ARToolkit works with ordinary PCs and laptops equipped with inexpensive web cams.

Our project experimented with the free version of the ARToolkit on Microsoft Windows, Linux, and Apple Macintosh laptops. We developed several building blocks for interactive AR, such as a module that allows an end user to transfer a graphical object from one fiducial marker to another by placing the markers in proximity with each other. This is a model for scenarios in which students interact with virtual artifacts with virtual tools. E.g., they can use a virtual trowel (that is represented in 3D graphics when the camera recognizes its fiducial marker) to "pick up" a virtual object when the trowel is sufficiently near the object. The demonstrations are summarized in Table 1 and available at [17].

Table 1. Summary of exploratory AR demos (see [17]).

Demo	Description
Switch	Switch switches which object is displayed on which marker when the markers come within a specified distance of each other.
Change	Change is very similar to switch. It changes which objects are displayed when two markers come within a specified distance of each other.
Ordered	Ordered will only display the objects in a particular order. If the camera 'sees' a marker, but has yet to display an object that comes before the one associated with this marker, it will not display the object. Once an object has been displayed, it will always be displayed when the camera is 'seeing' that specific marker. The order that the objects are displayed in is determined by the order the markers are loaded in.
Ordered2	Ordered2 does the exact same thing that Ordered does except that it only displays the object at the current 'level'. Previously viewed objects cannot be viewed again once a new object is displayed. The order is determined by the order the markers are loaded.
Flip_book	Placing the hiro within a specified distance of the yani marker causes the object displayed on the yani marker to change. Once again, the user cannot change the distance or the order that the objects appear in.
Choice	Choice assigns markers to objects to markers depending on which key is pressed. It will assign the object chosen to all markers that it recognizes at the time the key is pressed.
Shovel	Shovel lets you use a marker as a shovel in order to pick up virtual objects and place them on other markers. Only one object is shown at any one time. It would be possible to add more objects, but there is a good chance that it will greatly decrease the frame rate.

2.2. AndAR

AndAR is a port and partial reimplementation of the ARToolkit (Version 2.x) to the Android phone [1]. The Android system captures and displays the video, code from the ARToolkit implements the recognition, geometric analysis and tracking.

We built and tested AndAR on several Android phones. AndAR performance is poor. The tracking is unstable (the 3D content drops in and out frequently) and the display does not keep up with even slow movements of the marker or the phone.

We investigated the source code, and believe that there are significant performance bottlenecks in the image capture and image processing. Note that this code is substantially the same as that running on a PC, but the phone has far less computing power, memory, and internal bandwidth to sustain the real time video capture and display.

2.3. ARToolkit: iPhone (iOS)

ARToolworks has created an implementation of the ARToolkit for the iPhone and related devices, which is distributed as a commercial product [2]. This product is a port and partial re-implementation of the ARToolkit to run on the iOS (the iPhone/iTouch etc. operating system). The company reports that the camera tracks "at full camera frame rate (up to 30 fps)" [2], and we have observed acceptable performance for simple examples.

This version of the ARToolkit uses the Apple iOS to control the application, to manage the camera, and to manage the screen. The geometric analysis is performed by ARToolkit libraries, the device is managed by calls to iOS.

We obtained the ARToolworks code, including the iOS toolkit. Using this toolkit, we implemented the concepts developed on the PC (Section 2.1, Table 1, above). Considerable additional programming was required to make the applications work on the Apple iOS operating system, mostly to deal with platform speciific details.

2.4. Qualcomm Augmented Reality (QCAR) SDK

Qualcomm has implemented AR for Android, not based on the ARToolkit [12]. The Qualcomm Augmented Reality Platform runs on any Android device, and the SDK is available for free.

We investigated the QCAR, obtaining, building, and loading on an Android phone. QCAR has several interesting features, and appears to perform very well, much better than the AndAR, and comparable with the iOS.

QCAR has an interesting model for application development [13]. A developer uploads their image for the target that they want to track to the QCAR "Target Management System", which applies their proprietary algorithms to extract features from the image and create an encoded description of the image. The developer downloads the target resources, which are bundled with the application. The QCAR SDK provides a binary library that implements the proprietary recognition and tracking algorithms for the marker images.

While the application can be freely distributed, it must contain QCAR's proprietary library. Also, it is not possible to create content without using the proprietary service to encode the tracking data.

QCAR has an extension to work with the Unity Game Development Tools [23]. The QComm extension makes it possible to use the Unity system to manage the 3D scene. We are investigating the use of Unity for our scenarios.

3. Other Similar Software

An interesting alternative to handheld AR uses a web browser or smart phone as a remote device for an AR application on a server. In general, these configurations use the handheld as a display for AR on a remote system.

With this approach, the 3D content can be viewed, there is no direct manipulation of the objects and point of view (those are manipulated at the remote source). For this reason, this approach is less desirable for the interactive scenarios we are considering.

We have not investigated these products in detail, though they are potentially interesting, so we mention them here.

3.1 NyARToolkit, FLARToolkit, SLARToolkit

The NyARToolkit ([11]) is based on the public ARToolkit (version 2.721), modified to use a smart phone or similar device as a remote interface to AR running on a server. Essentially, the handled captures video, sends it to the server, which ships the updated view to the hand held. The computation is done on the server rather than the local device.

The FLARtoolkit ([16]) is a version of the NyARToolkit that implements the AR using Flash, which may be run in a web browser or on certain smart phones (but not iPhone).

A similar product is available for Microsoft Silverlight, SLARToolkit [19] (also a port of the NyARToolkit).

These products are available for free, and ARToolworks provides commercial support for them. The commercial products use later, commercial, revisions of the ARToolkit.

3.2. Total Immersion

Total Immersion [21] has a suite of AR tools, and also sells development services. The company provides both development and delivery, and operates as a consultancy and development studio. Their software and services are commercial.

This company has quite a few impressive customers, including, for example, Avatar movie tie-ins [4, 22].

3.3. Other Smart Phones and Handheld Devices

There are many other similar platform, including BlackBerry [14], Nokia [10], and Microsoft [9]. These platforms either have or soon will have AR SDK's similar to the ones we discuss here, along with their own operating systems.

Unfortunately, these platforms are incompatible, so that many applications have to be partly or wholly rewritten for each platform [20]. Toolkits are emerging that enable some kinds of applications to be written once and executed on many platforms [5]. However, Augmented Reality applications require access to the camera and screen in real time, and cannot exploit these techniques.

However, we believe that Apple's iOS and Android are the dominant platforms at this time, so we have focused our attention on them.

4. Discussion and Conclusions

From our research to date, we conclude that AR is at the edge of feasibility for smart phones today. The basic techniques are known, the challenge is to achieve satisfactory performance. It appears that satisfactory solutions are emerging, but there is no convergence on a single platform or operating system.

4.1 Platforms

Two leading handheld platforms are iPhone and Android. Both of these platforms can achieve about the same performance. Note that the newer iPad and several Android tablets have cameras, so it should be possible to implement AR on them as well as phones.

The iPhone/iTouch/etc. (now termed 'iOS') devices are well designed and consistent. The SDK is available for modest cost, though a license is required. iOS applications only work on Apple products, and distribution is subject to Apple's permission. The ARToolkit iOS support is very good.

The Android is an open platform, the SDK and operating system are available for free. In general, an Android app can be used on any Android device, and they can potentially be distributed through many channels. The Android AR toolkits (especially Qcomm) are very good.

Our current view is that we should pursue Android applications, because:

- Android is an open platform, with many vendors and devices
- There are good SDKs available
- Android is well engineered and well documented, and we find it easier to program than iOS

4.2 Creating and Porting Applications to Smart Phones

At this time, and for the foreseeable future, there will be many competing platforms in the market, with very limited compatibility. Achieving universal access for a product will require multiple versions of the software, which could well be quite costly.

A smart phone is highly constrained, and the operating systems are complex. While simple phone applications may be easy to create, any application that requires significant computation or communication will be challenging. Also, porting existing code from other platforms may well require substantial rewriting to work on the phone—if it can be made to work at all.

In summary, programming smart phones should be considered a difficult and costly task.

4.3 The Importance of Content

For all of our candidate technologies, there are significant challenges obtaining and managing "content"—the three dimensional graphics that are overlaid on the view. The general process is to create the 3D content with one of many tools available (such as commercial drawing and CAD systesm, and free tools such as Google Sketchup [6]). The 3D modesl are then "exported" to a file, which is input to the AR system. This process may require several iterations to refine the graphics, adjust the scale (e.g., to fit the available screen), and otherwise create the desired effect.

There is little standardization (or rather, too many standards) for exchanging 3D models and the different AR toolkits support different formats. As a result, importing models into the Augmented Reality system is challenging. In general, commercial software supports many more import formats that free ware.

We note that most of the products seem to be moving toward a preference to importing data in wavefront (".obj") format [25]. This format is also widely supported by tools (though not by the free version of Google Sketchup).

Beyond the details of file formats and tools, developing 3D content for AR requires talent and skill. Even when an AR viewer is available, each application requires labor-intensive (and likely very costly) effort to create the content. When planning a project, it is important to include graphic artists, and to plan for multiple iterations of the graphics design processes.

Addressing this challenge, several companies provide commercial tools and services for developing AR content. To date, we have not evaluated these products.

4.3. Summary

Table 2. Summary of main features of AR toolkits (based on limited evaluation)

	License	Target	Usability	Performance	Content

		Architecture			created?
ARToolkit v2.x [7]	Free	PC, laptop	OK	OK	Any 3D graphics
ARToolkit v4.x from ARToolworks [3]	$$$	PC, laptop, www	Good	Good	Any 3D graphics
ARToolkit for iOS [2]	$$$	iPhone	Good	Good	Any 3D graphics
AndAR [1]	Free	Android	OK	poor	Any 3D graphics
QCAR [12]	Free	Android	Good	Good	Any 3D graphics, Interfaces with Unity
NyARToolkit [11]	Free	www	OK	?	Any 3D graphics
FLARToolkit [16]	Free	www	OK	?	Any 3D graphics
SLARtoolkit [19]	Free	www	OK	?	Any 3D graphics
D´Fusion from Total Immersion [21]	$$$	www,mobile?		Good	D´Fusion Studio

Acknowledgements

This work was supported in part by the National Science Foundation, ARC-1025298 "Augmented Reality for Understanding Social and Environmental Science" and by the Illinois Program for Research in the Humanities at the University of Illinois at Urbana-Champaign **(**http://www.iprh.illinois.edu/) and a grant from the Illinois Campus Research Board.

Significant portions of the work discussed here was done by students: Lauren Semararo, Joe Noonan, and Nick Zukoski. Lauren Semararo was supported in part by the Research Experiences for Undergraduate Program National Center for Supercomputing Applications, NSF SCI: Award #0504064.

References

1. AndAR. *AndAR - Android Augmented Reality*. 2010, http://code.google.com/p/andar/.
2. ARToolworks, Inc. *ARToolKit for iOS*. 2009, http://www.artoolworks.com/products/mobile/artoolkit-for-ios/.
3. ARToolworks, Inc. *Welcome to ARToolworks*. 2010, http://www.artoolworks.com/.

4. AVATAR. *AVATAR: Transmissions from Pandora*. 2009, http://avtr.com/.
5. Charland, Andre and Brian Leroux, *Mobile application development: web vs. native.* Commun. ACM, *54 (5):49-53*, 2011.
6. Google. *Sketchup*. 2011, http://sketchup.google.com/.
7. Human Interface Technology Laboratory. *ARToolkit*. 2009, http://www.hitl.washington.edu/artoolkit/.
8. iprh. *Illinois Program for Research in the Humanities*. 2011, http://www.iprh.illinois.edu/.
9. Microsoft. *App Hub*. 2011, http://create.msdn.com/en-US.
10. Nokia. *Forum Nokia*. 2011, http://www.forum.nokia.com/Develop/.
11. NyARToolkit. *NyARToolkit: ARToolKit Class Library for Java/C#/Android*. 2010, http://nyatla.jp/nyartoolkit/wiki/index.php.
12. Qualcomm. *Augmented Reality*. 2010, http://developer.qualcomm.com/ar.
13. Qualcomm. *QdevNet*. 2011, http://ar.qualcomm.com.
14. Research In Motion Limited. *BlackBerry(R) Developer Zone*. 2011, http://us.blackberry.com/developers/.
15. sah.org. *The Society of Architectural Historians (SAH)*. 2011, http://www.sah.org/.
16. Saqoosha. *What is FLARToolKit*. 2010, http://www.libspark.org/wiki/saqoosha/FLARToolKit/en.
17. Semararo, Lauren, Alan Craig, and Robert E. McGrath. *Augmented Reality Demonstrations (August 2010)*. 2010, http://hdl.handle.net/2142/18810.
18. Sherman, William R. and Alan B. Craig, *Understanding Virtual Reality: Interface Application and Design*, San Francisco, Morgan Kaufmann, 2003.
19. SLARToolkit. *SLARToolkit - Silverlight Augmented Reality Toolkit*. 2010, http://slartoolkit.codeplex.com/.
20. Tarkoma, S. and E. Lagerspetz, *Arching over the Mobile Computing Chasm: Platforms and Runtimes.* Computer, *44 (4):22-28*, 2011.
21. Total Immersion. *Total Immersion*. 2009, http://www.t-immersion.com/.
22. Twentieth Century Fox. *Avatar*. 2009, http://www.avataritag.com/.
23. Unity Technologies. *Unity Game Development Tool*. 2011, http://unity3d.com/.
24. wikipedia. *Augmented Reality*. 2011, http://en.wikipedia.org/wiki/Augmented_reality.
25. wikipedia. *Wavefront .obj file*. 2011, http://en.wikipedia.org/wiki/Wavefront_.obj_file.