Mobile Phones as a Platform for Augmented Reality

Dieter Schmalstieg and Daniel Wagner

Graz University of Technology

Abstract: Handheld Augmented Reality (AR) running on self-contained handheld computers and smartphones, can leverage an extremely large potential user base of existing devices and users knowing how to operate them. In this paper we report on a platform for collaborative handheld AR applications, which employs specific efficient techniques from embedded development to push the limits of AR applications in terms of physical size, number of users and content intensity.

Keywords: Mobile phones, augmented reality, wearable computing

1 INTRODUCTION AND RELATED WORK

Mobile phones with embedded cameras make it possible to use a “magic lens” style of AR, using the live camera image both for computer vision tracking and for displaying augmented 3D images. Handheld devices combine CPU, graphics, camera, buttons or touchscreen, and wireless networking all in one conveniently designed package, making it very attractive as an off-the-shelf platform for AR. In the past five years we have built a number of handheld AR applications, in particular multi-user games, deployed in real world environments. In the course of this development, we have developed a complete application framework for deploying AR specifically on mobile phones.

A few other projects dealing with Augmented Reality on mobile phones or PDAs have been reported in literature. Early work used these devices as thin clients, outsourcing most processing tasks to PC-based servers via wireless connections [1][3].

Like the work reported here, later projects discarded the idea of outsourcing processing tasks in order to gain infrastructure independence. An early attempt on Symbian phones reported in [6] allowed only a very coarse estimation of the object's pose on the screen. VisualCodes [5] similarly allows only very coarse estimation. Later work ported ARToolKit to the Symbian platform and created the first two-player AR game for mobile phones [2]. ULTRA [4] uses PDAs for augmenting "snapshot" still images in non-real time.

None of the above approaches features a complete development platform for real-time AR on mobile phones. In this paper, we describe the software architecture of Studierstube ES, a framework for collaborative handheld AR applications, which employs specific efficient techniques from embedded development to push the limits of AR applications in terms of physical size, number of users and content intensity.

Figure 1: Nokia N95 running Studierstube ES superimposes geo-referenced content on a map of Graz

2 THE STUDIERSTUBE ES PLATFORM

As a foundation for Handheld Augmented Reality, we developed a software framework called *Studierstube ES* [7]. The framework is available for Windows CE and Windows XP, targeting small form factor devices such as shown in Figure 1. Experimental versions also exist for Symbian and Linux.

All processing is done natively on the handheld device, so that applications can run independently of any infrastructure and scale to an arbitrary number of simultaneous users. Typical frame rates on smartphones are in the order of 5-30 fps, depending on the content and device.

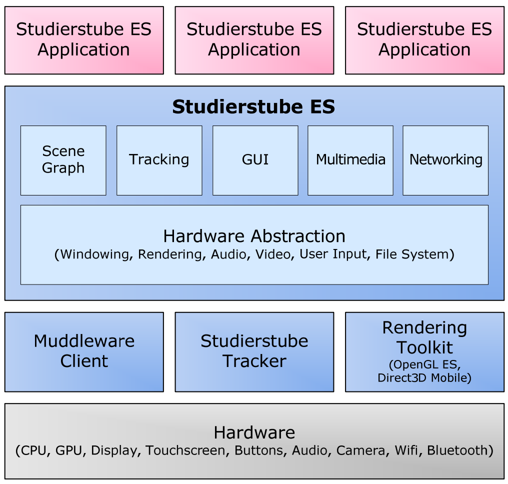

Figure 2: Software stack of the Handheld AR framework.

Figure 3: StbTracker supports a wide variety of markers: Template markers allow placing an image inside the rectangle; BCH markers directly encode 4096 IDs; DataMatrix markers can encode ~50 ASCII characters; Frame markers can have arbitrary or even no pattern at all; Split markers require only two bars on top and bottom, whereas the sides and content are free; Grid markers can span large areas of textured planar surfaces.

The client software framework is based on a component design (Figure 2), and allows customizing the runtime environment to the needs of the application and the capability of the handheld device. In particular, memory footprint can be optimized to as little as 500K for a basic system. However, an extensive set of components is available. The core components essential for AR are Studierstube Tracker, a real-time fiducial tracking component, and Studierstube Scene Graph, a rendering engine running on top of OpenGL ES or Direct3D Mobile.

Studierstube ES also offers scriptable components for networking, 2D user interfaces, Macromedia Flash playback, keyframe animation, audio, and video. Application code is managed through dynamically linked libraries, which simplifies memory management and downloading on demand.

A server component running on a PC was developed to address multi-user communication, content management and game-specific simulation [7]. Clients maintain a wireless networking connection to the server if available, but it can run fully stand-alone too. The server unifies blackboard-style communication with persistent storage of content. Data from transient events and persistent data objects are treated alike and are addressed using a hierarchical scheme. This design greatly simplifies rapid prototyping of multi-user applications over conventional relational database approaches.

Studierstube Tracker provides full 6 degree of freedom tracking in real time on mobile devices. It robustly tracks a variety of marker types (Figure 3) simultaneously and estimates the relative position and orientation between the marker and the phone's camera. The method searches for rectangles or corner points in a thresholded image, then decodes the contained barcode or pattern using one of several techniques. Finally, the camera pose is estimated from a homography. All computations are carried out with pure fixed point operations for optimal performance on embedded CPUs (~5ms on a 300Mhz ARM CPU).

3 Lessons Learned

The software foundation of Studierstube ES has evolved over several years to reflect the reality of developing for low-performance embedded devices. Similar to AR frameworks developed for a conventional PC platform, modularization, hardware abstraction and code reuse are essential for successful application development.

However, other aspects of software development differ significantly from larger platforms. Low memory footprint and memory bandwidth are essential requirements for embedded development. Consequently, features such as dynamic linking can be problematic. Moreover, many embedded devices can only perform fixed point computations, and have no or only very limited parallel execution. All code must be developed to meet these constraints and still perform efficiently. This means that not only the coding style but also the choice of algorithms can differ very much from conventional practices.

Overall, obtaining complete source code compatibility between a framework on PC and on a mobile phone is interfering with making optimal use of both platforms. We therefore suggest to achieve such compatibility not on the source code, but on the content and user interface level.

Acknowledgements

This project was funded in by part by Austrian Science Fund FWF under contract No. L32-N04 and Y193. Thank to Istvan Barakonyi, Matthias Stifter, Thomas Pintaric and Alessandro Mulloni.

References

[1] Gausemeier, J., Fruend, J., Matysczok, C., Bruederlin, B., Beier, D., Development of a real time image based object recognition method for mobile AR-devices, Proceedings of 2nd international conference on Computer graphics, Virtual Reality, Visualisation and Interaction in Africa, pp. 133-139, 2003

[2] Henrysson, A., Billinghurst, M., Ollila, M. Face to Face Collaborative AR on Mobile Phones. International Symposium on Augmented and Mixed Reality (ISMAR'05), pp. 80-89, 2005

[3] Ingram, D., Newman, J., Augmented Reality in a Wide Area Sentient Environment, Proceedings of the 2nd IEEE and ACM International Symposium on Augmented Reality (ISAR 2001), p. 77, 2001

[4] Makri, A., Arsenijevic, D., Weidenhausen, J., Eschler, P., Stricker, D., Machui, O., Fernandes, C., Maria, S., Voss, G., Ioannidis N., ULTRA: An Augmented Reality System for Handheld Platforms, Targeting Industrial Maintenance Applications, Proceedings of 11th International Conference on Virtual Systems and Multimedia (VSMM'05), 2005

[5] Michael Rohs Marker-Based Embodied Interaction for Handheld Augmented Reality Games Journal of Virtual Reality and Broadcasting (JVRB), Vol. 4, No. 5, March 2007

[6] Möhring, M., Lessig, C. and Bimber, O., Video See-Through AR on Consumer Cell Phones, International Symposium on Augmented and Mixed Reality (ISMAR'04), pp. 252-253, 2004

[7] Schmalstieg D., Wagner D., Experiences with Handheld Augmented Reality. Proc. IEEE International Symp. on Mixed and Augmented Reality, 2007.