Mixed Reality Based Interaction Techniques Using Smartphones in Bulletin Board Applications

Ulrich Schwanecke, Jürgen Zeitz, Ralf Dörner

Wiesbaden University of Applied Sciences
Department for Design, Computer Science and Media
{schwanecke, doerner}@informatik.fh-wiesbaden.de

20. September 2007

Abstract

Smartphones, i.e. full featured mobile phones with integrated functionality for personal computing, become not only increasingly popular but also increasingly powerful. Faster speed, higher screen resolution and improved camera and video capabilities facilitate the implementation of sophisticated imaging algorithms on recent devices. Based on methodologies for mobile mixed reality, we present a bulletin board application for smartphones. Our system allows users not only to retrieve location based information from the bulletin board but also to post their own multimedia content at designated points of information (POIs). To identify a POI we combine data from GSM with an optical tracking procedure. In our application, we devised novel interaction techniques in order to cope with the limited size of handheld devices and the peculiarities of their mobile usage. We use a *magic lens* metaphor and take advantage of the additional information that is available from using the camera of the smartphone as an input device. This enables the user to interact with multimedia content by simply moving or rotating the smartphone.

1 Introduction

Modern mobile devices such as personal digital assitants (PDAs) with cell phone capabilities or smartphones do not restrict themselves to voice communication anymore. They contain comprehensive image capturing and video capabilities and the latest generation is nearly as powerful as early laptops. The sale of smartphones is the fastest growing segment in the handset market. Smartphones made the step from a niche product to a consumer product. Today, already 10 percent of all handsets throughout the world are smartphones. A fraction that is expected to increase up to 22 percent within the next four years [9].

Even with popular-priced devices, the increasing computing power and the availibility of cameras allow new application scenarios for smartphones employing mixed reality. Based on computer vision and computer graphics, mixed reality (MR) methodologies allow to combine computer-generated images with images from the real world in an elaborate way. As a result, a blend of reality with virtual worlds can be achieved. For the seamless integration of reality and virtuality, a variety of complex processes concerning tracking, simulation and rendering have to be combined. With the latest smartphone generation, these time-consuming computing steps can be performed in real-time.

As smartphones are becoming more and more prevalent, the demand for new applications is rising. One class of new applications is built upon location based services like distributed bulletin boards. We consider bulletin boards to be points of information (POI) distributed over a spacious area like a university campus or even a whole city. As smartphones make the usage of MR feasible, we enhance the presentation of the content of a bulletin board with MR and rich multimedia. Multimedia information which is associated with each POI can be retrieved from a server via a wireless network. In addition to the information retrieval, users can post their own content to selected bulletin boards. Thus, our virtual bulletin board aims at letting people exchange location-sensitive information. In this manner, people can use the system to communicate with each other whereby the context of communication is implicitly set by the location of the POI. For instance, hotel guests may want to leave a note at a certain location in their room (e.g. some object needs to be repaired) that can be read later by maintenence personnel.

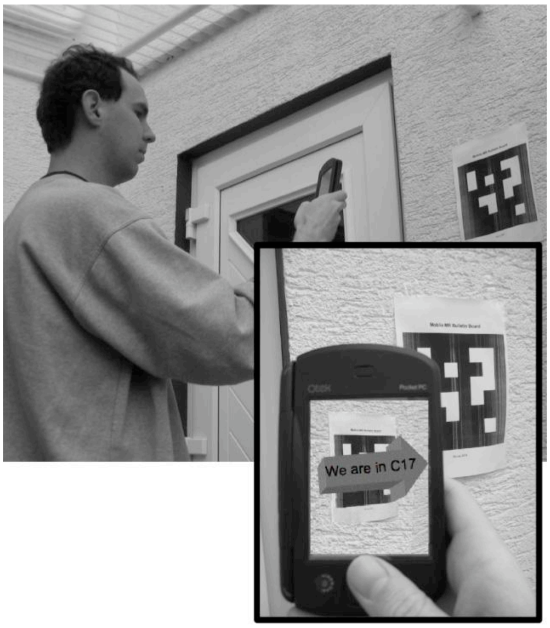

Figure 1: Accessing a virtual bulletin board using a *magical lens* metaphor. The smartphone's screen serves as a window into a mixed reality environment.

A typical use case starts with a user accessing a POI. For this, the user scans a fiducial marker (which has been attached to a real-world object) with the camera of his smartphone (see figure 1). The system identifies the POI and loads the associated multimedia content from a server. Finally, the content is displayed on the smartphone; it can be a simple text as well as audiovisual media or even complete 3D-modells augmenting the image taken from the POI. The visual markers fulfill two tasks. First, they signal the location of the virtual bulletin board to the user. Second, they allow the registration of the displayed objects with the real environment since they are easily recognizable by computer vision.

In order to access the content provided at a POI as well as to prepare own content for posting, the user needs to interact with the textual, aural or visual representations of the location based information. For this, the provision of appropriate interaction techniques is crucial because they significantly determine the usability and end user acceptance of the application. In this paper, we present novel means of interaction that take advantage of the specific features present in smartphones. For instance, we implemented algorithms to detect whether the smartphone is held in portrait or landscape mode. Thus, the system is able to adapt the user interface without user intervention or the need for special hardware. An optimized motion detection algorithm has been conceived that relies on an analysis of the captured video. It is able to determine how the user moves the device. This information is interpreted in the context of user interaction, and allows novel techniques for scrolling, zooming and rotating objects displayed on the smartphone. Thus, mixed reality methodologies are not only used for enhancing the presentation of information but also for improving user interaction.

The remainder of this paper is organized as follows. In section 2, we give a short overview of mobile applications using mixed reality technology and compare our system with existing ones. Section 3 describes the scenario and architecture of our system. The novel interaction techniques of our system are dicussed in Section 4. Section 5 briefly describes some important implementation details. In the last section, we conclude and give an outlook on future work.

2 Background

The term *mixed reality* describes the sophisticated combination of virtual environments with reality. One reason so far preventing the wide-spread use of mobile mixed reality applications has been the lack of adequate mobile

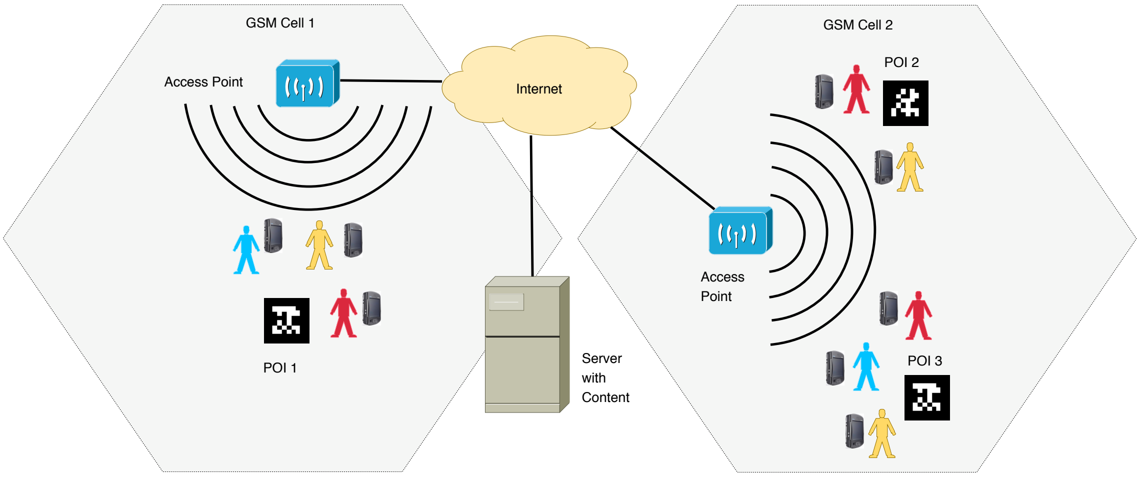

Figure 2: Scenario of our mobile mixed reality bulletin board. POIs are identified by using the GSM- Cell-ID together with information from an optical marker.

computers, displays and interfaces. Modern smartphones are able to mitigate this problem.

In the literature, there are several mixed reality projects reported which take advantage of mobile platforms. The *Studierstube* project ported *ARToolKit* [12] to the PocketPC platform and developed several PDA–based applications (see e.g. [17, 19, 20, 21]). These standalone-applications run without using a server infrastructure. Other projects like [4, 14] need this kind of infrastructure in order to shift some of the computational load to a server. In the field of mobile location based applications, the projects *MARS* [3, 7] and *NaviCam* [13, 16] are notable. The *MARS*–projects are designed for outdoor-scenarios. They use big backpack-computers and GPS–Tracking. The *NaviCam* application runs on a handheld device and uses a very simple optical tracking mechanism with colored markers which do not allow object-registration. Furthermore, there are applications in the field of cultural heritage like Archeoguide [6] and GEIST [10].

In contrast to most mobile MR–scenarios, our system's goal is not only to retrieve and display information in an MR scenario. It also aims at letting users create their own information and enrich the virtual world dynamically with user created content. The only known project which has a similar goal is the *Augment-able Reality* project [15]. This project is based on *NaviCam* but instead of handheld devices it makes use of backpack-computers. As a result, it has severe restrictions concerning mobility. A crucial difference to the *Augment-able Reality* project is that we use handheld devices with an integrated camera. Evaluating the video stream obtained by the camera, we determine the relative motion of the mobile device. This information facilitates the realisation of new interaction techniques. Controlling the application or manipulating the displayed content can be accomplished by moving the device instead of pushing buttons.

3 System Design

The *mobile mixed reality bulletin board* is a locally distributed information system with multimedia content. POIs which can be spread over a wide area are identified by a twodimensional printable marker and its ID. We use the *ARToolKitPlus* library [22] to identify a marker and acquire its position and orientation. Since the number of IDs is limited to 4096 by ARToolKitPlus, additional information has to be used in order to extend the number of possible POIs. The spatial dispersion of the system can be used for this purpose. The usage of a smartphone offers the opportunity to retrieve the current GSM–cell-ID of the mobile network. Therefore, the rough position of the user can be distinguished. The cell-ID in combination with the marker-ID increases the number of possible

POIs to 4096 per GSM–cell. This should be sufficient for most purposes especially due to the fact that the diameter of a GSM–cell varies from only a few meters up to 35 kilometer depending on the expected number of users inside a cell. Places like fairgrounds or campuses, for example, where the numer of desired POIs is high, are covered by a lot of GSM–cells and therefore a large number of POIs can be identified. Figure 2 depicts the scenario of our mobile mixed reality bulletin board. Different POIs possess unique IDs resulting from a combination of a visual marker and a GSM cell-ID. Thus, identical visual markers can be used for POIs in different GSM cells.

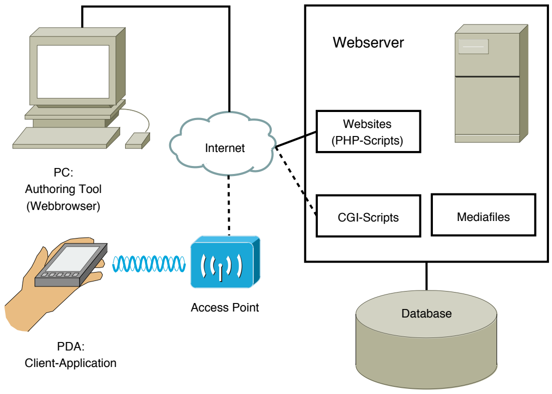

Figure 3: The architecture of the *mobile mixed reality bulletin board*. A smartphone makes wireless access of data possible. Alternatively, data can be accessed by a Web application on a PC for administration purposes.

Our system consists of different parts which are shown in figure 3. The multimedia-based content of the system is stored in a database and the file system of a *webserver*. There are two ways to access the information stored on the server:

- With the mobile client application for smartphones any user can retrieve data from the server. Thereby the smartphone connects to the wireless access point associated with the bulletin board the user is looking at.

- Administrators have the opportunity to manage the data directly on the server using a PHP–based frontend. Beside the maintenance and administration of the content, it is possible to configure the state of a bulletin board; a bulletin board can have the state *public*, i.e. every user can query the POI and post own content, or *private*, i.e. a user can only query the content of the POI he ist allowed to and only qualified users can submit content.

Compared to traditional bulletin boards, private POIs are especially advantageous because they are able to dynamically restrict reading and writing access to a specified subgroup of persons.

3.1 Media

Our application is able to present multimedia content in different ways. The different types of content and their representation are

- *Images* can be zoomed and scrolled in four directions. The image planes which contain the images can be registered with objects in the real world according to the visual markers representing the location of the bulletin board.

- Textured *threedimesional objects* can be registered with the real world (see figures 1 and 4).

- Long *texts* which can be scrolled vertically are drawn in a field on the display to guarantee good readability (see figure 4).

- *Audio–clips* of different formats can be played via the devices speakers.

Al type of media can be accessed consistently using interaction metaphors described in the following section.

4 User Interaction

There are multiple ways to control our system and interact with the media displayed. We use the *magic lens* metaphor to explore a POI. This means that the screen of our smartphone serves as a window into the mixed reality environment. The video stream of the smartphone's camera is analyzed in order to merge the real world imagery with virtual content. Thus, the user can perceive the real world that lies in the

Figure 4: Textual content can be displayed as a three-dimensional object combined with imagery of the real world – or as 2D text.

camera field of view through the video image displayed on the screen – but it is not just an image of the real world but augmented with additional virtual information.

Beside the typical input devices available on a smartphone like touchscreen and keyboard, we are using additional pieces of information extracted from the video stream in order to interact with the content presented in a more intuitive way. On the one hand, we use *marker–related information* namely the orientation of our physical markers for facilitating interaction. On the other hand, *motion–related information* obtained by a reliable real-time motion-detection algorithm for different content-dependent interaction metaphors is used.

Figure 5: If a user switches from portrait to landscape view (by rotating the mobile device), the 2D GUI is adjusted automatically.

4.1 Marker–related

The physical markers recognized by the system make it possible to identify a POI and retrieve the corresponding content from a server. For this purpose, the user holds the smartphone in a way that the physical marker is seen on the camera image. Having identified the current POI, the system automatically downloads the appropriate media and presents it in a proper way, e.g. by using mixed reality methods.

Another use of the 2D–markers is to determine the device orientation with respect to the tag. This orientation can be used to align the 2D elements of the GUI (see figure 5). So, the user can hold the device in a way he prefers and the GUI is adjusted automatically. Since the 2D GUI is rotated in 90 degree steps, there are only four possible orientations. The coarse angle between the device and the detected marker is sufficient to determine the desired GUI orientation. We calculate the angle α between the image of the x–axis of the world coordinate system attached to the marker and the x–axis of the image plane (see figure 6).

The threshold used to determine whether to switch from one mode to the other depends on the aspect ratio of the smartphones screen (see figure 7). For a ratio of $w : h$ we get

$$\omega = 2 \cdot \arcsin \frac{w}{\sqrt{w^2 + h^2}} \quad \text{and} \quad \phi = 180^\circ - \omega$$

for the corresponding angles. The aspect ratio of most smartphones is $3 : 4$. In this case we obtain $\omega \approx 74^\circ$ and $\phi \approx 106^\circ$. Therefore, we switch from portrait to landscape mode if $|\alpha|$ is greater than 37° and from landscape to portrait mode if $|\alpha|$ is greater than 53°.

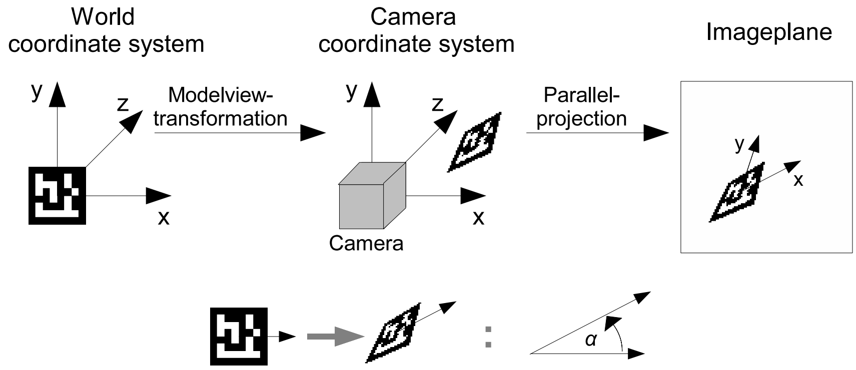

Figure 6: To determine the orientation of the image of a marker we project the x–axis of its (world) coordinate system into the imageplane and evaluate the angle α between the projected axis and the horizontal.

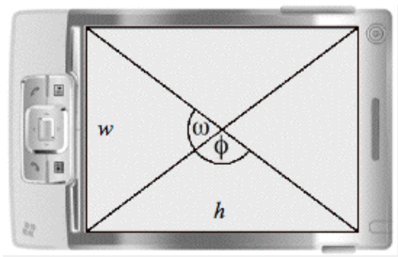

Figure 7: For $\omega < \Phi$ a user expects that he has to tilt the display less in order to switch from portrait to landscape mode than the other way around.

4.2 Motion–related

We detect the motion of the device and use it as a basic information for interaction metaphors. Depending on the content currently displayed, it is possible to manipulate twodimensional media as well as three-dimesional objects.

Motion detection is mostly based on optical flow analysis (for details see e.g. [11]). Many fast algorithms have been developed to determine the optical flow of an image sequence (see e.g. [1, 5, 8]). One crucial point is that the outcome obtained by most optical flow algorithms is strongly affected by illumination variations. In a real environment, the illumination is changing perpetually. Therefore, we have to expect that optical flow analysis will not provide satisfactory results. Moreover, we do not need the whole field of optical flow. We only want to determine one vector describing the relative motion of our mobile device. This vector should be fast to compute and as reliable as possible. As a result, we implemented a motion detection process based on *Projection Shift Analysis* described in [2] .

The main idea of *Projection Shift Analysis* is to analyze projections of images instead of the images themself. Therefore, in a first step the images are projected onto the x and y axis. For an image $I(x, y)$ of size (w, h) these projections are

$$I_x(x) = \sum_{i=0}^{h-1} I(x,i) \quad \text{and} \quad I_y(y) = \sum_{i=0}^{w-1} I(i,y)$$

with $x \in \{0, \dots, w-1\}$ and $y \in \{0, \dots, h-1\}$.

To determine a vector $\vec{v}$ describing the motion between the images $I^{(1)}(x, y)$ and $I^{(2)}(x, y)$ one can use the normalized sum of squared differences of projections shifted by s as a quality measurement for the shift s. For the x–direction this is

$$g_x(s) = \frac{1}{w - |s|} \sum_{i=0}^{w-1-|s|} \Delta I_x(i,s)^2$$

with

$$\Delta I_x(i,s) = \begin{cases} I_x^{(1)}(i+|s|) - I_x^{(2)}(i) & \text{for } s \geq 0 \\ I_x^{(2)}(i+|s|) - I_x^{(1)}(i) & \text{for } s < 0 \end{cases}$$

where $I_x^{(1)}(x), I_x^{(2)}(x)$ are the corresponding projections of $I^{(1)}(x, y)$ and $I^{(2)}(x, y)$ respectively. For $g_y(s)$ we obtain an analog formula.

The function $g_x(s)$ describes the difference between $I_x^{(1)}(x+s)$ and $I_x^{(2)}(x)$ i.e. the quality of the shift s of the two projections. The argument s with minimal value $g_x(s)$ describes the most likely shift in x–direction. The same can be done for the y–direction. Altogether, we obtain

$$\vec{v} = \left(\arg\left(\min_s(g_x(s))\right), \arg\left(\min_s(g_y(s))\right)\right)^T$$

as a vector characterising the motion between $I^{(1)}(x,y)$ and $I^{(2)}(x,y)$. This motion indicates the motion of our mobile device.

The acquired motion vector $\vec{v}$ can be used in different interaction techniques. In the actual implementation, we use it for scrolling, zooming and rotating purposes. By holding a device-button, the smartphone can be set into different navigation modes. When zoom-mode is active and the device is moved up, the displayed content is enlarged. By moving the device in the opposite direction the content is scaled down. This works in the same way for two-dimensional and three-dimensional items. In scroll-mode, the effect of the motion depends on the type of media which is currently displayed. In large images like for example city-maps where the image is only shown partly scrolling takes place in the opposite direction of the device-motion. This gives the impression of looking through a window or magnifying glass and fits well into our magic lens metaphor. The borders where parts of the larger image have been clipped are visualized as black lines (see figure 8).

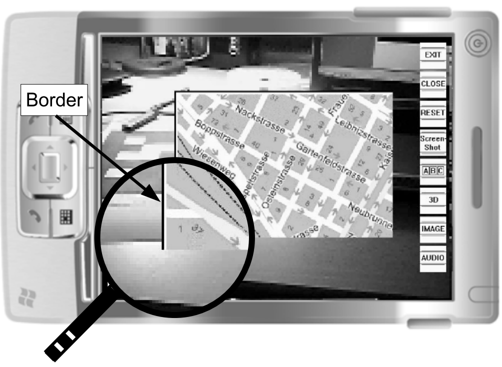

Figure 8: A scrollable image of a city map. Black lines marking the borders of the image. The map can be scrolled and zoomed by moving the smartphone.

With threedimensional objects, the behavior of the scroll-mode is slightly different. By tilting the device the displayed object is rotated with an adjusted *ArcBall Rotation Control* [18]. The ArcBall Control metaphor interprets the motion of a mouse as a motion along great circles on a sphere. The starting point and end point on this sphere are determined by the coordinates of the mouse while a button is pressed and held respectively. Unfortunately, the evaluated motion–vector $\vec{v}$ does not contain any information about its start– and end–coordinates. Therefore, we assume the start-coordinate of each motion to be the center of the screen (see figure 9). Thus, by switching to scroll–mode we initialize a motion–vector $\vec{v_a} = 0$. In scroll–mode, all vectors obtained in the motion detection step are added to $\vec{v_a}$ in order to calculate the end–coordinate of the 2D motion used by the ArcBall Rotation. With this process, rotating an object can be done in an intuitive way even if the object is not centered on the screen. This is because only relative movements are applied to the manipulated object. In order to keep user interaction consistent, the same mechanism is executed when rotating an object using a stylus on the touchscreen.

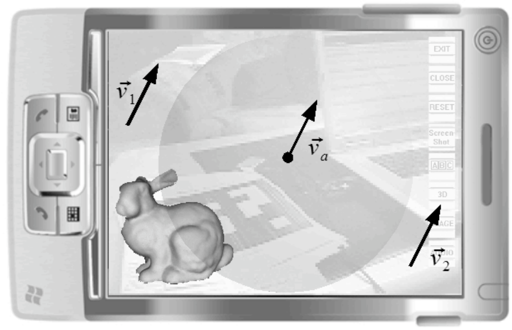

Figure 9: Modified ArcBall Rotation: The motions $\vec{v_1}$ and $\vec{v_2}$ are equivalent to the motion $\vec{v_a}$.

4.3 Posting Media

Our system distinguishes between *public* and *private* POIs. A public POI equals a physical bulletin board. Every user can see the complete content of a public POI and leave own content. Private POIs are more restricted. Every user can query the content. But only qualified users are able to submit content.

If a user is permitted to leave information at an appointed bulletin board the client application displays an icon showing an arrow and a globe (see lower center in figure 10). To post data to a POI, the user has to specify the content to be sent to the server and push the “arrow and globe”–icon on the touchscreen to confirm. There are different input options depending on the type of the data.

Textual content can be entered by using the keyboard of the device or a virtual keyboard on the touchscreen. We implemented a virtual keyboard covering the complete screen in order to make the use easy and reliable.

Images, 3D objects and audio files can be selected from different showcases depending on the type of the media file (see figure 10). Each showcase enables the user to browse through the content, to prepare a preview of the selected file, and to submit a selected item to a POI.

Figure 10: The showcases for image– and 3D-model–selection respectively.

Audio–clips can be created instantly by pushing a device button. The recorded audio–clip is added directly to the audio–showcase and therefore can be associated with a POI.

5 Implementation

The client application which runs on windows mobile based smartphones is implemented in C++ to achieve high performance. A complex runtime-environment like .NET would slow down the application considerably and waste the rare resources of the device. Moreover, some of the required APIs like DirectShow are only available native which makes the development in C++ mandatory. To gain maximum performance, Intels optimized compiler for XScale–processors is used [20].

To access the smartphone’s camera device–independently, Windows Mobile’s *DirectShow* API is used. Thus, our client application should work on all Windows Mobile 5 devices with an attached camera like the Qtek 9000 from HTC we have used in our experiments.

To visualize a threedimensional scene and superimpose it with the video of the real world, the *Direct3D Mobile* API is our choice because it is integrated directly in the operating system of the device. The alternatively available 3D–library *klimt* is not used because of performance issues even if it is not restricted to the Windows technology like our current client application is.

The client program uses two DirectShow filters which were developed in order to use the camera’s videostream with the application:

- A *grabber–filter* is used to extract images from the current live-videostream of the camera.
- A special *render–filter* (which does nothing) is used to terminate the filtergraph without displaying the video.

Due to the limited amount of memory available on current smartphones, there are some restrictions. Some Direct3D objects like large textures cannot be created. As a result, some content cannot be drawn in the MR–scene. This affects big images of e.g. maps as well as long texts which are internally represented as images. Furthermore, extra memory would permit to create an additional DirectShow filtergraph for the playback of a video–file or the live-video–stream of a webcam for example.

Audio playback is based on the external cross-platform library *FMOD*. In contrast to the DirectShow playback of audio content, no filtergraph has to be created. Therefore, using the external library saves precious ressources of the handheld device. In order to keep filesize small we are using a sample rate of 8kHz and 8 bit quantization which is sufficient for voice recording.

On the server-side, Python CGI–scripts handle the requests of the client application and provide the requested media. The frontend that manages the information stored on the server is implemented in PHP. The Python–scripts and the PHP–scripts access a PostgreSQL–database which stores information in the background.

6 Conclusion and Future Work

Mixed reality technology opens new vistas of applications for mobile devices. In the *mobile mixed reality bulletin board* application presented, smartphones are used to access and to submit varying multimedia information at different locations in an intuitive way. This is made possible through novel interaction techniques that exploit the additional information available in the video stream captured by the smartphone's camera.

The combination of more memory and a more powerful processor would allow using a higher camera-resolution for better pattern-detection and a better view on the virtual objects. With additional memory, the client application could instantly create video–files to be posted to a POI (similar to the way audio–files are treated).

Another major task which has to be solved is to run the client application in threads. In the current version, this could not be implemented correctly because the C–API of the ARToolKitPlus-module and the C++–APIs (e.g. DirectShow) of the application caused memory leaks when run in threaded mode.

In our actual implementation, the communication between the server application and the smartphone is based on WLAN technology. Since we are still using the GSM network to identify the POI of course a next step is to substitute the WLAN technology by other mobile data networks like GPRS or UMTS.

References

[1] Ted Camus. Real-Time Quantized Optical Flow. In *Proceedings of IEEE Conference on Computer Architectures for Machine Perception*, September 1995.

[2] Stephan A. Drab and Nicole M. Artner. Motion Detection as Interaction Technique for Games & Applications on Mobile Devices. In *Pervasive Mobile Interaction Devices (PERMID 2005) Workshop at the Pervasive 2005*, München, May 2005.

[3] Steven Feiner, Blair MacIntyre, Tobias Höllerer, and Anthony Webster. A touring machine: Prototyping 3d mobile augmented reality systems for exploring the urban environment. In *ISWC*, pages 74–81, 1997.

[4] Jürgen Fründ, Christian Geiger, Michael Grafe, and Bernd Kleinjohann. The augmented reality personal digital assistant. In *The 2nd International Symposium on Mixed Reality ISMR2001, The Virtual Reality Society of Japan*, Japan, 2001.

[5] Ben Galvin, Brendan McCane, Kevin Novins, David Mason, and Steven Mills. Recovering Motion Fields: An Evaluation of Eight Optical Flow Algorithms. In *Proceedings of the British Machine Vision Converence (BMVC)*, September 1998.

[6] A. Hildebrand, P. Dahne, F. Seibert, I. T. Christou, A. Demiris, N. Ioannidis M. Diorinos, L. Almeida, A. Diogo, and J. Weidenhausen. Archeoguide: An augmented reality based system for personalized tours in cultural heritage sites. *Cultivate Interactive*, 1, 2000.

[7] Tobias Höllerer, John V. Pavlik, and Steven Feiner. Situated documentaries: Embedding multimedia presentations in the real world. In *ISWC*, pages 79–86, 1999.

[8] Berthold K. P. Horn and Brian G. Schunck. Determining Optical Flow. *Artificial Intelligence*, 17:185–203, 1981.

[9] Berg Insight. Smartphone operating systems, vas research series 2007, May 2007.

[10] R. Jany and R. Leiner. Ein mobiles ar-informationssystem zum erleben historischer zusammenhänge im urbanen umfeld - das projekt geist. *HGG-Journal*, 16:262–265, 2001.

[11] Bernd Jähne. *Digital Image Processing. Concepts, Algorithms, and Scientific Applications*. Springer, 6 edition, 2005.

[12] Hirokazu Kato and Mark Billinghurst. Marker tracking and hmd calibration for a video-based augmented reality conferencing system. In *IWAR '99: Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality*, page 85, Washington, DC, USA, 1999. IEEE Computer Society.

[13] Jun Rekimoto. The magnifying glass approach to augmented reality systems. In *International Conference on Artificial Reality and Tele-Existence '95 / Conference on Virtual Reality Software and Technology '95 (ICAT/VRST'95)*, pages 123–132, November 1995.

[14] Jun Rekimoto. Transvision: A hand-held augmented reality system for collaborative design. In *Proceedings of Virtual Systems and Multimedia (VSMM) '96*, 1996.

[15] Jun Rekimoto, Yuji Ayatsuka, and Kazuteru Hayashi. Augment-able reality: Situated communication through physical and digital spaces. In *ISWC '98: Proceedings of the 2nd IEEE International Symposium on Wearable Computers*, page 68, Washington, DC, USA, 1998. IEEE Computer Society.

[16] Jun Rekimoto and Katashi Nagao. The world through the computer: Computer augmented interaction with real world environments. In *Symposium on User Interface Software and Technology (ACM)*, pages 29–36, Pittsburgh, Pennsylvania, United States, November 1995.

[17] Dieter Schmalstieg and Daniel Wagner. A handheld augmented reality museum guide. In *IADIS International Conference on Mobile Learning*, Malta, June 2005.

[18] Ken Shoemake. Arcball: A User Interface for Specifying Three-Dimensional Orientation Using a Mouse. In *Proceedings of the conference on Graphics interface '92*, pages 151–156, San Francisco, CA, USA, 1992. Morgan Kaufmann Publishers Inc.

[19] Daniel Wagner, Thomas Pintaric, Florian Ledermann, and Dieter Schmalstieg. Towards massively multi-user augmented reality on handheld devices. In *Third International Conference on Pervasive Computing (Pervasive 2005)*, Munich, Germany, May 2005.

[20] Daniel Wagner and Dieter Schmalstieg. Artoolkit on the pocketpc platform. In *Second IEEE Intl. Augmented Reality Toolkit Workshop (ART032)*, pages 14–15, Tokyo, Japan, October 2003.

[21] Daniel Wagner and Dieter Schmalstieg. First steps towards handheld augmented reality. *ISWC*, pages 127–135, 2003.

[22] Daniel Wagner and Dieter Schmalstieg. Artoolkitplus for pose tracking on mobile devices. In *Proceedings of 12th Computer Vision Winter Workshop (CVWW'07)*, February 2007.