

 Open access • Proceedings Article • DOI:10.1109/ISMAR.2005.32

Face to Face Collaborative AR on Mobile Phones — [Source link](#)

[Anders Henrysson](#), [Mark Billinghurst](#), [Mark Ollila](#)

Institutions: [Linköping University](#), [University of Canterbury](#)

Published on: 05 Oct 2005 - [International Symposium on Mixed and Augmented Reality](#)

Topics: [Mobile phone](#), [Mobile computing](#), [Augmented reality](#) and [Collaborative software](#)

Related papers:

- [A survey of augmented reality](#)
- [Video See-Through AR on Consumer Cell-Phones](#)
- [Marker tracking and HMD calibration for a video-based augmented reality conferencing system](#)
- [Recent advances in augmented reality](#)
- [First steps towards handheld augmented reality](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/face-to-face-collaborative-ar-on-mobile-phones-2srwitpizr>

Face to Face Collaborative AR on Mobile Phones

Anders Henrysson
NVIS
Linköping University
andhe@itn.liu.se

Mark Billingham
HIT Lab NZ
University of Canterbury
mark.billinghurst@hitlabnz.org

Mark Ollila
NVIS
Linköping University
marol@itn.liu.se

ABSTRACT

Mobile phones are an ideal platform for augmented reality. In this paper we describe how they can also be used to support face to face collaborative AR gaming. We have created a custom port of the ARToolKit library to the Symbian mobile phone operating system and then developed a sample collaborative AR game based on this. We describe the game in detail and user feedback from people who have played the game. We also provide general design guidelines that could be useful for others who are developing mobile phone collaborative AR applications.

Keywords

Mobile Camera Phones, Mobile Games, Human Computer Interaction, Tangible Interfaces, Augmented Reality

1. INTRODUCTION

In recent years mobile phones have developed into an ideal platform for augmented reality (AR). The current generation of phones have full colour displays, integrated cameras, fast processors and even dedicated 3D graphics chips. Henrysson [8] and Moehring [15] have shown how mobile phones can be used for simple single user AR applications. In their work they create custom computer vision libraries that allows developers to build video see through AR applications that run on a mobile phone.

Now that it is technically possible, it is important to conduct research on the types of AR applications that are ideally suited to mobile phones and user interface guidelines for developing these applications. This is significant because the widespread adoption of mobile phones means that this platform could be one of the dominant platforms for AR applications in the near future.

One particularly interesting area for mobile phone based AR is for supporting collaborative AR applications. Mobile phones are already designed to support local and remote communication and so provide a natural platform for collaborative AR. For example a blue-tooth enabled mobile phone can be used for face to face gaming or messaging, while the cellular network supports voice and video calls. In this paper we present the first example of a face to face collaborative AR application based on mobile phones.

In the next section we review related work on mobile AR and collaborative AR, and then talk about user interface aspects of mobile phone AR and the software platform we have developed to support phone based AR applications. Next we describe the collaborative AR game we have developed based on this platform and the user response to the game. Finally we conclude with some design guidelines for mobile phone based collaborative AR systems and directions for future research.

2. RELATED WORK

Our work draws on a rich legacy of previous work in mobile augmented reality, collaborative augmented reality, AR interaction techniques and mobile phone gaming.

From the early days of Feiner's Touring Machine [6] it was obvious that what was carried in a backpack would one day be held in the palm of the hand. Feiner showed the potential of mobile AR systems for outdoor context sensitive information overlay, while the ARQuake [24] showed how these same systems could be used for outdoor gaming.

At the same time as these early mobile systems were being developed, Schmalstieg [22], Billingham [2] and Rekimoto [20] were exploring early collaborative AR interfaces. Billingham's Shared Space work showed how AR can be used to seamlessly enhance face to face collaboration [3] and his AR Conferencing work [4] showed how AR can be used to creating the illusion that a remote collaborator is actually present in a local workspace, building a stronger sense of presence than traditional video conferencing. Schmalstieg's Studierstube [22] software architecture was ideally suited for building collaborative and distributed AR applications and his team also developed a number of interesting prototypes of collaborative AR systems. Finally Rekimoto's Transvision system explored how a tethered handheld display could provide shared object viewing in an AR setting [20]

Using Studierstube, Reitmayr [19] brought these two research directions together in a mobile collaborative augmented reality interface based again on a backpack configuration. Prior to this Hollerer [10] had added remote collaboration capabilities to the University of Columbia's

system, allowing a wearable AR user to collaborate with a remote user at a desktop computer. Piekarski and Thomas [18] also added similar remote collaboration capabilities to their Tinmith system, once again between a wearable AR user and a colleague at a desktop computer. However Reitmayr's work was the first that allowed multiple users with wearable AR systems to collaborate together in spontaneous ways, either face to face or in remote settings. These projects showed that the same benefits that tethered AR interfaces provided for collaboration could also extend to the mobile platform, and new application areas could be explored, such as location based gaming.

As significant computing and graphics power became available on the handheld platform, researchers have naturally begun to explore the use of personal digital assistants (PDA's) for AR applications as well. First there was work such as the AR-PDA project [9] and BatPortal [11] in which the PDA was used as a thin client for showing AR content generated on a remote server. This was necessary as the early PDA's did not have enough capability for stand-alone AR applications. Then in 2003 Wagner ported ARToolKit[1] to the PocketPC and developed the first self contained PDA AR application [26]. Since that time, Studierstube has been ported to the handheld platform and the first stand alone collaborative AR applications based on PDA's have been developed [27]. Unlike the backpack systems, handheld collaborative AR interfaces are unencumbering and ideal for lightweight social interactions.

Mobile phone based AR has followed a similar development path. Early phones did not have enough processing power so researchers also explored thin client approaches. For example, the AR-Phone project [5] used Bluetooth to send phone camera images to a remote sever for processing and graphics overlay, taking several seconds per image. However, Henrysson recently ported ARToolKit over to the Symbian phone platform [8], while Moehring developed an alternative custom computer vision and tracking library [15]. This work enables simple AR applications to be developed which run at 7-14 frames per second.

A third research thread that our work draws on is AR interaction techniques. As mobile AR applications have moved from a wearable backpack into the palm of the hand the interface has also changed. The first mobile AR systems used head mounted displays to show virtual graphics overlaid on the real world and developed a number of very innovative techniques for interacting with the virtual data. For example, in the Tinmith system [18] touch sensitive gloves were use to select menu options and move virtual objects in the real world. Kurata's handmouse system [13] allowed people to use natural gesture input in a wearable AR interface, while Reitmayr's work implemented a stylus based interaction method [19]. This

research showed that with mobile AR systems intuitive interfaces can be developed by considering the possible affordances of the system. For example, with a backpack system glove input or using a handheld device is natural because while wearing a head mounted display the user has both hands free.

PDA-based AR applications do not typically use head mounted displays, but are based instead around the LCD display on the PDA or handheld device. At least one of the user's hands is needed to hold the PDA so some of the earlier mobile interaction techniques are not suitable. It is natural in this setting to use stylus input but there are other possibilities as well. In the AR-PAD project [16], buttons and a trackball on the display are used as input in a face to face collaborative AR game. Träskbäck uses a tablet-PC and pen input for an AR-based refinery education tool [25], and markers in the environment are used as cues to load the correct virtual content. In Wagner's indoor navigation tool [26] user input is also a combination of stylus interaction and knowledge of display position from visual tracking of markers in the environment. These projects show that if the AR display is handheld the orientation and position of the display can be used as an important interaction tool.

Handheld AR applications such as the Invisible Train [27] also show an interesting combination of interacting with the AR content by interacting in the world and with the device itself. In this case, the user moves around in the real world to select the view of the virtual train set and then touches the screen with a stylus to change the position of tracks on the train set. Similarly in Wagner's AR-Kanji collaborative game [28] the user looks through the PDA screen to view real cards which have Kanji symbols printed on them. When the cards are seen through the screen virtual models are seen corresponding to the translation of the Kanji characters. These can be manipulated by hand and PDA shows the model from different viewpoints. There is very little stylus input required.

We can draw on this research to explore interaction techniques for mobile phone-based AR. In this case we have a display that is typically smaller than on a PDA platform and with even more limited input options. Later in this paper we describe our approach to AR interaction design for mobile phones.

Finally, work in mobile phone gaming has been used to inform our AR application design. Although there are thousands of games available for mobile phones, there are only a handful that use camera input. Two of the best known are "Mosquito Hunt" [17] and "Marble Revolution" [14]. In "Mosquito Hunt", virtual mosquitoes are superimposed over a live video image from the camera and simple motion flow techniques are used to allow the user to shoot the Mosquitos by moving the phone. Similarly, in the "Marble Revolution" game the player can steer a marble through a maze by moving the phone and using motion

flow techniques. Neither of these games are collaborative or true AR applications, but they do show that camera and phone motion can be used to create compelling game experiences.

The application most related to our work is Hakkarainen's "Symball" game [7]. This is a two person collaborative table tennis game which uses camera phones that are Bluetooth equipped. On their phone screen players can see a table tennis table and a virtual paddle. They select a real colour that they would like their phone to track and as they move the phone relative to this colour the paddle moves in the x-y direction on the screen. Players can either play alone or connect to another phone through Bluetooth and play again each other. Once again this is not a true AR experience, but it is the first example of a compelling collaborative game on phone that user camera input.

Other relevant phone applications are the Spotcode [23] and Q-Code pattern tracking systems. Spotcode is a two-dimensional ring like bar code that can be tracking in real time with a phone camera. The Spotcode software performs image processing techniques to extract the identity of the pattern and its angular orientation relative to the phone. This library can then be used to develop a number of interesting ubiquitous computing applications. Similarly Q-Code is a two-dimensional bar code developed in Japan that can also be recognized by mobile phones. Although our application is based on ARToolKit [1], the real time performance of these systems led us to believe that we should also be able to get good performance from our code.

As can be seen we have drawn on a significant of related work for this project. However our research is also different from what has been done before. We demonstrate the first face to face collaborative AR application running on a mobile phone and we provide user study results to evaluate this application. Unlike the other handheld AR applications described, we are focusing on applications that encourage and require multi-user input. This focus provides us with a challenging set of design challenges. Our game also uses multi-sensory output (audio, visual and tactile) to further engage with the player. Finally, we are developing an interaction design approach that is uniquely suited to the limited display and input requirements of mobile phones.

3. INTERFACE METAPHORS

Thus there are several key differences between using a mobile phone AR interface compared to a traditional head mounted display (HMD) based system, including:

- the display is handheld rather than headworn
- the phone affords a much greater peripheral view
- the phone the display and input device are connected

These differences mean that interface metaphors developed for HMD based systems may not be appropriate for

handheld phone based systems. For example, applications developed with a Tangible AR metaphor [12] often assume that the user has both hands free to manipulate physical input devices which will not be the case with mobile phones.

Compared to a PDA the mobile phone is operated using a one-handed button interface in contrast to the two-hand stylus interaction of the PDA. It is therefore possible to use the mobile phone as a tangible input object itself. In order to interact we can move the device relative to the world instead of moving the stylus relative a fairly static screen.

The approach that we are following is to assume the phone is like a handheld AR lens giving a small view into the AR scene. With this in mind we assume that the user will be more likely move the phone-display than change their viewpoint relative to the phone. Thus the small form factor of the mobile phone lets us go beyond the looking-glass metaphor to an object-based approach.

This metaphor can be applied to other AR applications that do not use a HMD, such as applications developed for projection screens, tablet-PC and PDAs. However the mobile phone is even more "object-like" than these other devices. This means that our input techniques are largely going to be based around motion of the phone itself, rather than keypad input into the phone.

In the next section we describe the software development necessary to build AR applications for the phone and then how this AR lens metaphor can be applied in AR application development.

4. SOFTWARE DEVELOPMENT

In order to develop collaborative AR applications for Symbian based mobile phones there were several key steps we needed to perform:

- port the ARToolKit tracking library to the Symbian operating system
- develop a peer to peer communications layer
- build a game application using 3D graphics
- provide support for audio and haptic feedback

In this section we review each of these steps in more detail.

Our collaborative AR platform is based on a custom port of the ARToolKit tracking library [1] to the Symbian operating system by Henrysson [8]. Wagner's work [26] on porting ARToolKit to the Windows CE PDA platform was also used for inspiration.

Henrysson was the first to implement ARToolKit for Symbian. To do this he wrote a C++ wrapper class in order to get rid of global variables which are prohibited by Symbian. However, both the mobile phones we are targeting and the PDA used by Wagner lack a floating point unit, making floating-point arithmetic orders of magnitude slower than integer arithmetic. To overcome

this, Wagner identified the most computational heavy functions and rewrote them to fixed point using Intels GPP library.

Since there is no equivalent fixed-point library featuring variable precision available for Symbian, we wrote our own. We did extensive performance tests to select the algorithms that ran fastest on the mobile phone. The average speed-up compared to corresponding floating-point functions was about 20 times. We started out by porting the functions rewritten by Wagner and continued backwards to cover most of functions needed for camera pose estimation. The resulting port runs several times faster than the original port. Some accuracy was lost when converting to fixed point but was perceived as acceptable.

Once ARToolkit was running on the phone we needed a way to transfer data between phones. Since our game is a face-to-face collaborative application we have chosen Bluetooth and we wrote a simple Bluetooth peer to peer communications layer. Our set-up consists of two mobile phones where one is a server that announces the game as a service and provides a channel for the client to connect to. The client makes an active search for the device and the service. There is thus no need for IP configuration. Once a connection is detected the game is ready to be played.

In addition to communication software we needed graphic application code. The OpenGL library is a powerful graphics API that was the natural starting point for the development of a graphics API for mobile devices. Our graphics application was developed using OpenGL ES which is a subset of OpenGL 1.3, suitable for low-power, embedded devices. To make it run on these limited devices some, members of the Khronos group removed redundant APIs and functions. Memory and processor demanding functions such as 3D texturing and double precision floating point values have been removed along with GLU. A 16:16 fixed-point data type has been added to increase performance while retain some of the floating-point precision. The most noticeable difference is the removal of the immediate mode in favor of vertex arrays. Since Symbian does not permit any global variables the vertex and normal arrays must be declared constant, which limits the dynamic properties of objects.

The phone we were developing for, the Nokia 6630, ships with a software implementation of OpenGL ES. While this takes care of the low level rendering there is still need for a higher-level game engine with ability to import models created with 3D animation software and organize the content into a scene graph. Though M3G (JSR 184) provides model loading features it does not allow us to invoke the ARToolkit tracking library written in C++ since there is no equivalent to Java Native Interfaces (JNI) for J2ME. There are a few commercial game engines written in C++ but they are not suited for AR research applications

that use calibration data and a tracking library to set the camera parameters.

To be able to import textured models from a 3D animation package we used the Deep Exploration tool from Right Hemisphere to convert the exported model to C++ code with OpenGL vertex arrays and then wrote a simple program that converted this into OpenGL ES compatible vertex arrays.

Finally, we needed to add support for audio and tactile feedback for our application. In addition to playing ring tones to alert the user of incoming calls and messages, mobile phones also provide a vibration mode used when audio is not appropriate for social reasons. Since we intend to use the mobile phone as an interaction device rather than a passive screen the vibration mode provide us with a means to give tactile feedback for user events such as collisions or rule violations. The Symbian API lets us set the duration and strength of the vibration to adapt it to various scenarios. The built-in media server allows us to set the frequency and volume of a specified audio sample. This allows us to adapt the audio feedback for various events.

5. SAMPLE APPLICATION

With the optimized AR libraries and architecture we have developed it would be possible to build a number of different AR applications. Our focus is on face to face collaborative AR and so our first application is a simple tennis game. Tennis was chosen because it could be played in either a competitive or cooperative fashion, awareness of the other player is helpful, it requires only simple graphics and it is a game that most people are familiar with.

Our tennis application uses a set of three ARToolkit markers arranged in a line (see figure 1). When the player points the camera phone at the markers they see a virtual tennis court model superimposed over the real world.



Figure 1: Playing AR tennis

As long as one or more of these markers are in the field of view then the virtual tennis court will appear. This marker set is used to establish a global coordinate frame and both of the phones are tracked in this coordinate frame.

There is a single ball that initially starts on the phone that is set up as the blue-tooth server. To serve the ball the player points their phone at the court and hits the '2' key on the keypad. The ball is served if the phone clients are connected to each other. Once the ball is in play there is no need to use the keypad any more. A simple physics engine is used to bounce the ball off the court and respond to when the player hits the ball with their camera phones (figure 2).



Figure 2: Hitting the ball over the net

The simulation takes place in marker space. To check for possible collision with the racket, the position of the ball is transformed into camera space. This transformation is

given by the ARToolKit tracking. The racket is defined as a circle with 4 cm radius centered on the z-axis in the xy-plane of the camera space. If there is an intersection between the racket plane and the ball (a cylinder in simulation space), the direction of the z-axis is transformed into marker space and used to initialize the simulation.

The direction and position vectors of the ball are sent over to the other phone using Bluetooth. By sending the position the simulations will be synchronized each round. When receiving data the device switches state from outgoing to incoming and starts to check for collision with the racket. Both devices check for collision with the net and if the ball is bounced outside the court. If an incoming ball is missed the user gets to serve since the other devices Bluetooth is in listening mode. The simulation will always be restarted when data is sent and received.

Each time the ball is hit there is a small sound played and the phone of the person that hits the ball vibrates, providing multi-sensory cues to help the players.

We have not implemented score keeping yet, relying on players to keep score themselves. However this could be added in the future.

As an extension to the basic application we also experimented with placing an ARToolKit marker on the back of each phone (see figure 1). In this way when one player caught sight of the others phone he or she would see a virtual tennis racquet superimposed over the phone (see figure 3). This allows the players to more easily perceive the whereabouts of their opponents in order to adjust their own behavior or to give guiding instructions in a collaborative task. It will however restrict the motion range since one of the court markers must visible as well.



Figure 3: The virtual tennis racquet.

The game was tested on both the Nokia 6600 and 6630 phones. Both phones have a screen resolution of 176x208 pixels. The video resolution provided by the camera is 160x120 pixels. The Nokia 6600 has a 104 Mhz ARM processor and was able to run the application at around 3-4 frames per second. In contrast the 6630 has a 210 Mhz ARM processor and achieved frame rates of up to 7 frames per second. This is fast enough to play the game without too much difficulty. Further optimization of the ARToolKit library and application code could improve this performance further, but the fastest that the camera on these phones can capture video is 15 frames per second. Note that this performance is for the entire game application and by turning off Bluetooth and audio and haptic feedback performance increases significantly.

6. USER FEEDBACK

In order to evaluate the usability of mobile phones for collaborative AR we conducted a small pilot user study. We were particularly interested in two questions:

1/ Does having an AR interface enhance the face to face gaming experience?

2/ Is multi-sensory feedback useful for the game playing experience?

To explore these questions we conducted two experiments, both using the AR tennis game we have developed.

6.1 Experiment One: The value of AR

In this first study we were interested in exploring how useful the AR view of the game was, especially in providing information about the other player's actions.

Pairs of subjects played the game in each of the following three conditions:

A: Face to Face AR – where they have virtual graphics superimposed over a live video view from the camera.

B: Face to Face non AR – where they could see the graphics only, not the live video input

C: Non Face to Face gaming – where the players could not see each other and also could see the graphics only. There was no live video background used.

Figure 4 shows a screen shot of the application running with and without the live video background.

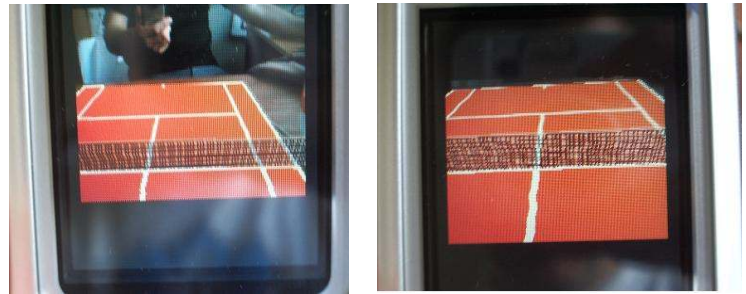


Figure 4: The application with and without live video

In the Face to Face conditions (A and B) players sat across a table facing each other sharing a single set of tracking markers (figure 5), while in condition C the players sat with a black cloth dividing them and each used their own tracking marker (figure 6).

Players were allowed to practice with the application until they felt proficient with the game and they were told to play for 3 minutes in each of the conditions. The goal was to work together to achieve the highest number of consecutive ball bounces over the net. This was to encourage the players to work together on a cooperative goal.

After each condition the number of ball bounces was recorded and also a simple survey was given asking the subjects how well they thought they could collaborate together.

Six pairs of subjects completed the pilot study, all of them male university staff and students aged between 21 and 40 years. Both of the players used a Nokia 6630 phone to ensure the highest frame rate possible. The experimental conditions were presented in different orders to reduce order effects on the outcomes.



Figure 5: Face to Face condition



Figure 7: Non-Face to Face condition

6.2 Experiment One Results

In general there was a large variability in the number of ball bounces counted for each condition. For each pair the number of ball bounces was normalized by dividing the results for conditions B and C by the number of bounces for condition A. However these normalized values still differed widely and there was no statistically significant difference across conditions.

This is not surprising because pairs used many different strategies for play the game. Some move the mobile phone about freely as if they were playing an actual game of tennis while others discovered they could get a very high score by placing the phones directly opposite each other and just bouncing the ball between them.

However we did get some significantly different results from the subjective user surveys. At the end of each condition subjects were asked the following four questions:

- 1/ How easy was it to work with your partner?
- 2/ How easily did your partner work with you?
- 3/ How easy was it to be aware of what your partner was doing?
- 4/ How enjoyable was the game?

Each questions was answered on a Likert scale from 1 to 7 where 1 = Not Very Easy and 7 = Very Easy. Table 1 below shows the average scores for each question across all conditions.

	A	B	C
Q1	4.83	4.13	3.75
Q2	4.50	4.25	3.83
Q3	5.92	3.17	2.50
Q4	5.17	5.25	5.33

Table 1: Subjective Survey Responses

As can be seen the responses to questions 4 are almost the same. An ANOVA test on these questions found no statistical difference, meaning that users found each condition equally enjoyable. Interestingly enough, despite simple graphics and limited interactivity this enjoyment score was relatively high.

However there was a significant difference in response to the first three questions. For question 1 (ANOVA $F(2,33) = 8.17, p < 0.05$) and for question 2 (ANOVA $F(2,33) = 3.97, p < 0.05$). The user felt that there was a difference between the conditions in terms of how easy it was to work with their partner and how easily their partner worked with them.

There is also a highly significant difference in response to question 3 (ANOVA $F(2,15) = 33.4, p < 0.0001$). Users felt that it was much easier to be aware of what their partner was doing in the face to face AR condition with the live video background than in the other two conditions which had no video background.

Subjects were also asked to rank the three conditions in order of how easy it was to work together where 1 = easiest and 3 = most difficult. Table 2 shows the average rankings.

A	B	C
1.08	2.25	2.67

Table 2: Conditions ranked by ease of collaboration (1 = easiest, 3 = most difficult)

Again there is a significant difference across ranking values (ANOVA $F(2,33) = 34.1, p < 0.001$). Remarkably, all but one of the users (11 out of 12) ranked the Face to Face AR condition as the easiest to work together in, and then split their opinion almost evenly between the remaining two conditions. This confirms the results from the earlier subjective survey questions.

After the experiment was completed subjects were also briefly interviewed about their experience. In general people overwhelmingly felt that seeing the AR view aided the face to face collaboration. Condition C was least favourite, because the collaborator was not visible – either in the phone or in peripheral vision. One subject even said that he didn't feel like he was playing with another person in condition C. Several people also commented on how adding graphics cues such as virtual shadows and a more realistic lit and shaded ball would help with the depth perception. Some people also had trouble with the ARToolKit marker tracking due to markers being covered up or shadows falling on the patterns. However once they practiced further they adopted behaviours that reduced the tracking problems.

In the experimental application we did not include the virtual tennis racquets described earlier. However at the end the experiment subjects were shown the virtual racquet and asked if they thought that would help with the collaboration further. The majority answered that they felt the visual cues provided by the live video of their collaborator was enough.

6.3 Discussion

It was interesting observing subject behaviour during the experiment. Subjects would often grasp the cell phone with both hands and start intently at the screen while playing, never looking across the table at their partner (see figure 5). Although they were collaborating in a face to face setting the focus of their attention was on the small screen.

Each of the three conditions provides less visual information about the player's partner. In the AR case the user can see their partner in the game space. Naturally, having a view of their collaborator on the screen allows users to feel connected to their partner, especially when they can see the phone of the other player and track their movements. In the non AR face to face condition the player can still get some understanding of what their partner is doing through the use of peripheral vision, but this is more difficult. Finally in the non face to face case they can just hear their partner, there are no visual cues at all. Thus it is not surprising that awareness dropped substantially between conditions. This shows one of the key benefits of AR interfaces for face to face collaboration - users can see their collaborators at the same time as the virtual information they are interacting with.

As subjects were playing the game their behaviour evolved over time. Initially many people tried playing tennis like they would in the real world, moving the phone from side to side to place shots over the virtual court. However the most successful players soon learned that holding the phone relatively still and sending the ball to the same location on the tennis court each time produced the best results. This is because fast camera motion can cause failure with the ARToolKit tracking and also increase the chances of missing a rebounding ball. The pairs that best adapted to each others style of play were those that were playing in the face to face AR condition where they had the best collaboration cues as to what their partner was doing.

These results seem to show that an AR interface does indeed enhance the face to face gaming experience with mobile phones.

6.4 Experiment Two: Multi-sensory Feedback

A second study was conducted to explore the value of having multi-sensory feedback in the collaborative AR application. In this case players played the game in the face to face AR condition used in experiment one, however they experienced the following variations in game feedback:

A: Face to Face AR with audio and haptic feedback

B: Face to Face AR with no audio feedback but with haptic

C: Face to Face AR with audio but no haptic feedback

D: Face to Face AR with no audio and no haptic feedback

These four conditions were used to explore which of the audio and tactile options the players found most valuable.

Each pair of players played in each condition for one minute, once again counting the highest number of consecutive ball bounces over the net and also completing a survey after each condition. Once again the order of conditions was varied to reduce order effects.

The same six pairs who completed experiment one also completed experiment two. After finishing the conditions for experiment one they would continue to complete the conditions for experiment two, so that they were trained on the system.

6.5 Experiment Two Results

As with the first experiment there was a wide variability in the average number of ball bounces counted and no statistical difference across conditions.

However we did get some significantly different results from the subjective user surveys. At the end of each condition subjects were asked the following three questions:

1/ How easy was it to be aware of when your had hit the ball?

2/ How easy what it to be aware of when your partner had hit the ball?

4/ How enjoyable was the game?

Once again each questions was answered on a Likert scale from 1 to 7 where 1 = Not Very Easy and 7 = Very Easy.

Table 3 below shows the average scores for each question across all conditions.

	A	B	C	D
Q1	5.75	5.00	5.33	3.17
Q2	5.42	4.17	5.17	3.33
Q3	5.83	4.83	4.83	4.17

Table 3: Subjective Survey Responses

There was a significant difference in response to all the questions. For question 1 about how easy the player felt it was to be aware of when they had hit the ball (ANOVA $F(3,44) = 11.1, p < 0.0001$). For question 2 about how easily they were aware of their partner hitting the ball (ANOVA $F(3,44) = 6.59, p < 0.001$). Finally for question 3 about how enjoyable the game was (ANOVA $F(3,44) = 6.53, p < 0.001$).

Subjects were also asked to rank the four conditions in order of how easy it was to work together where 1 = easiest and 4 = most difficult. Table 4 shows the average rankings.

A	B	C	D
1.17	3.00	1.92	3.91

Table 4: Conditions ranked by ease of collaboration (1 = easiest, 3 = most difficult)

Almost all of the subjects ranked condition A best (10 out of 12 responses), followed by condition C (audio but no haptic feedback), then condition B (haptic but no audio feedback) and finally condition D (no audio or haptic feedback). There is a very significant difference between these rankings (ANOVA $F(3, 44) = 102.6, p < 0.0001$).

After the experiment was completed subjects were also briefly interviewed about their experience. The majority thought that having both the audio and haptic feedback was the best choice for the application. They also felt that audio only was more valuable than haptic input only. This is partly because the audio provides a cue to both the person hitting the ball and the receiver, while the haptic vibration is only helpful for the person hitting the ball. Several users also requested that more audio feedback be added to the application, such as having sounds for the ball hitting the court or the net.

6.6 Discussion

These results show that users do feel that multi-sensory output is indeed important in face to face AR gaming. They almost unanimously rated the condition which provided the most sensory output (audio, visual, haptic) as easiest to work in and also as the most enjoyable. There also appears to be a clear preference for audio only output over haptic output. This appears is in part due to great awareness cue that audio provides for both the user and their partner when they hit the ball. With haptic only feedback, for the player that is not hitting the ball it is equivalent to having no feedback at all.

Interestingly enough the most successful players again adapted their behaviours to the interface conditions. In the non-audio cases they would often call out to the other player when they hit the ball, giving them the cue that they needed to perform better.

7. DESIGN RECOMMENDATIONS

In developing a collaborative AR game for mobile phones we have learned a little about design guidelines that can be applied to future collaborative games. The results above suggest that face to face mobile games could benefit from adding support for AR technology that would allow game graphics to be combined with views of the real world and the people that the user is playing with. The use of multi-

sensory feedback, especially audio and visual is important for increasing game enjoyment.

There are certain types of games that appear suitable for collaboration AR on mobile phones. If visual tracking is used then the ideal games have a focus on a single shared game space, such as with our tennis game. This enables the players to easily see each other at the same time as the virtual content.

The slow tracking performance of the current generation of phones means that the best games will also be those that don't really rely on quick reflexes or fast competition. Our tennis game worked because it was played in a cooperative manner. If players were competing against each other then it would have been too easy to score unanswered points.

The screens on mobile phones are very small so collaborative AR games need only use a limited amount of graphics and should mainly focus on enhancing the face to face interaction. For example in our tennis game a very simple ball, court and net model was used, but this was enough to keep users happily engaged.

The use of an appropriate tangible object metaphor is also important for the usability of mobile phone AR applications. In our case we wanted to player to feel like that the phone was a tennis racquet hitting balls over a virtual net. This is why the phone vibrated when a ball was hit and a racquet sound was made. Once they understood this metaphor it was easy for users to move the phone around the court space to hit the ball. Physical manipulation of a phone is very natural so provides and intuitive interaction approach for collaborative AR games. This also meant that apart from hitting a key to start serving there was no need to use keypad input while the game was underway.

8. CONCLUSIONS AND FUTURE WORK

In this paper we have described an early collaborative AR application for mobile phones. In order to develop this application we needed to create a highly optimized custom port of the ARToolKit library and then add application graphics and communication code.

Previous work has shown that AR technology can be used to naturally enhance face to face collaboration. The results from our user studies show that these same benefits can be found from using mobile phones. Even with a small screen and limited input capability, users felt that they were more aware of what their partner was doing in the face to face AR condition than in the other more traditional gaming conditions. Game enjoyment did not change over the different conditions, but awareness of their partner did which is important for effective collaboration. Providing support for multi-sensory output (visual, audio, haptic)

further increased the ease of collaboration and player enjoyment.

Part of the reason that users enjoyed playing the tennis game was the interaction metaphor used. In our work we consider the phone to be a tangible input device and use the motion of the phone as the primary interaction method. This is very different from traditional AR interfaces where the display and input devices are separated, but is ideal for small form factor phones. This interaction design approach and our preliminary design recommendations are two of the key contributions of the paper.

Mobile phones are becoming more and more advanced in processing power. With the addition of cameras with more detailed resolution AR is becoming a real possibility. Further features such as positioning (cell based and GPS) in conjunction with accelerometers in future handsets will provide a significant increase into the applications.

In the future we intend to develop a multimedia engine based on the built-in Symbian features and providing the developer with a simple interface to graphics, sound and communication besides tracking. We will also conduct more rigorous user studies to better understand the use of mobile phones as a platform for augmented reality and provide design guidelines back into the AR community.

9. ACKNOWLEDGMENTS

The authors would like to acknowledge the funding support of the Australasian CRC for Interaction Design (ACID). The first author is funded by the Department of Science and Technology at the University of Linköping as well receiving supervisory support from the Swedish National Graduate School in Computer Science. We would also like to thank Daniel Wagner for the coding advice he has given us and David Sickinger for assistance with the user studies.

10. REFERENCES

- [1] ARToolKit website: www.hitl.washington.edu/artoolkit/.
- [2] Billinghurst, M., Weghorst, S., Furness, T. (1996) Shared Space: Collaborative Augmented Reality. Workshop on Collaborative Virtual Environments 1996 (CVE 96), 19-20 Sept. 1996, Nottingham, UK.
- [3] Billinghurst, M., Poupyrev, I., Kato, H., May, R. Mixing Realities in Shared Space: An Augmented Reality Interface for Collaborative Computing. In Proceedings of the IEEE International Conference on Multimedia and Expo (ICME2000), July 30th - August 2, New York.
- [4] Billinghurst, M., Kato, H., (1999) Real World Teleconferencing. In Proceedings of the conference on Human Factors in Computing Systems (CHI 99). May 15th-20th, Pittsburgh, USA.
- [5] Cutting D. J. C. D., Assad M. and Hudson A. *AR phone: Accessible augmented reality in the intelligent environment*. In OZCHI2003, Brisbane, 2003.
- [6] Feiner T. H. S., MacIntyre B. and Webster T. *A touring machine: Prototyping 3d mobile augmented reality systems for exploring the urban environment*. In Proc. ISWC '97 (First IEEE Int. Symp. On Wearable Computers), Cambridge, MA, 1997.
- [7] Hakkarainen, M., Woodward, C., "SymBall - Camera driven table tennis for mobile phones", submitted to *ACM SIGCHI International Conference on Advances in Computer Entertainment Technology (ACE 2005)*, Valencia, Spain, 15-17 June, 2005 .
- [8] Henrysson A. and Ollila M. *UMAR - Ubiquitous Mobile Augmented Reality* In Proc. Third International Conference on Mobile and Ubiquitous Multimedia (MUM2004) pp 41-45. College Park, Maryland, U.S.A. October 27 - 29, 2004.
- [9] Geiger C, Kleinjohann B, Reimann C, Stichling D. *Mobile AR4ALL*, ISAR 2001, The Second IEEE and ACM International Symposium on Augmented Reality, New York, (2001).
- [10] Höllerer T., Feiner S., Terauchi T., Rashid G., and Hallaway D. Exploring MARS: developing indoor and outdoor user interfaces to a mobile augmented reality system. *Computer& Graphics*, 23(6): 779-785, 1999.
- [11] Ingram, D., Newman, J., *Augmented Reality in a WideArea Sentient Environment*. Proc. of the 2nd IEEE and ACM International Symposium on Augmented Reality (ISAR 2001), October 2001, New York.
- [12] Kato H., Billinghurst M., Poupyrev I., Tetsutani N. and Tachibana K. Tangible Augmented Reality for Human Computer Interaction. In *Proc. of Nicograph 2001*. (Nagoya, Japan, 2001).
- [13] Kurata, T., Takashi Okuma, T., Kourogi, M., and Sakaue, K.: "The Hand-mouse: A Human Interface Suitable for Augmented Reality Environments Enabled by Visual Wearables", In Proc. International Symposium on Mixed Reality (ISMR 2001) in Yokohama, Japan, pp.188-189 (2001)
- [14] Marble Revolution. <http://www.bit-side.com/entertainment/MOBILE%20GAMES/Marble>
- [15] Moehring, M., Lessig, C. and Bimber, O. *Video See-Through AR on Consumer Cell Phones*. In Proc. of International Symposium on Augmented and Mixed Reality (ISMAR'04), pp. 252-253, 2004.
- [16] Mogilev, D., Kiyokawa, K., Billinghurst, M., Pair, J. *AR Pad: An Interface for Face-to-face AR Collaboration*, Proc. of the ACM Conference on Human Factors in Computing Systems 2002 (CHI '02), Minneapolis, pp.654-655, 2002.
- [17] Mosquito Hunt. http://w4.siemens.de/en2/html/press/newsdesk_archive/2003/foe03111.html
- [18] Piekarski, W. and Thomas, B. H. Tinmith-Hand: Unified User Interface Technology for Mobile Outdoor Augmented Reality and Indoor Virtual Reality. In IEEE Virtual Reality Conference, Orlando, FL, Mar 2002.

- [19] G. Reitmayr, D. Schmalstieg Mobile Collaborative Augmented Reality In Proc. ISAR 2001, New York, USA, Oct. 2001.
- [20] Rekimoto J., *TransVision: A Hand-held Augmented Reality System for Collaborative Design*. Virtual Systems and Multi-Media (VSMM)'96, 1996.
- [21] Schmalstieg, D., Fuhrmann, A., Hesina, G., Szalavári, Z., Encarnação, L.M., Gervautz, M., Purgathofer, W.: *The studierstube augmented reality project*. Presence: Teleoperators and Virtual Environments 11 (2002) 33–54.
- [22] Schmalstieg, D., Fuhrmann, A., Szalavari, Z., Gervautz, M: "Studierstube" - An Environment for Collaboration in Augmented Reality Proceedings of Collaborative Virtual Environments '96, Nottingham, UK, Sep. 19-20, 1996
- [23] SpotCode. <http://www.highenergymagic.com>.
- [24] Thomas, B., Close, B., Donoghue, J., Squires, J., De Bondi, P., Morris, M., and Piekarski, W. *ARQuake: An Outdoor/Indoor Augmented Reality First Person Application*. Proc. 4th Int'l Symposium on Wearable Computers, pp 139-146, Atlanta, Ga, USA, Oct 2000.
- [25] Träskbäck M., Haller, M. Mixed reality training application for an oil refinery: user requirements., In ACM SIGGRAPH International Conference on Virtual Reality Continuum and its Applications in Industry, VRCAI 2004, pp. 324- 327, Singapore.
- [26] Wagner, D., Schmalstieg, D.: *First steps towards handheld augmented reality*. Proc. of the 7th International Symposium on Wearable Computers (ISWC2003), White Plains, NY, USA, IEEE Computer Society (2003) 127–137.
- [27] Wagner D., Pintaric T., Ledermann F., Schmalstieg D. *Towards Massively Multi-User Augmented Reality on Handheld Devices*. Proc. of the Third International Conference on Pervasive Computing (Pervasive 2005), Munich, Germany (to appear).
- [28] Wagner, D., Barakonyi, I.: *Augmented reality kanji learning*. Proc. of the 2003 IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR 2003), Tokyo, Japan, IEEE Computer Society (2003) 335–336.