Mobile AR Requirements for Location Based Social Networks

Tuukka Turunen[1], Tino Pyssysalo[2] and Juha Röning[3]

[1] Digia Plc, Oulu, Finland
[2] Digia Plc, Tampere, Finland
[3] Department of Electrical Engineering, University of Oulu, Finland

***Abstract*—Utilizing the novel User Interface (UI) technology of Augmented Reality (AR) in mobile phones provides significant advantages for Location Based Social Networks (LBSN) via powerful UI that allows the user to see the world through AR view rather than via a traditional map view. Compared to use of a map based interface, it is much easier for the user to understand where the nearby friends and points of interests are located when using a mobile AR interface to access the information of LBSN services. Recent development in the commercially available high end mobile phones has made it a viable device to use globally available AR services, but there still exist some limitations when it comes to LBSN services. Especially demanding is the need to include constantly moving friends reliably and accurately as annotated objects into the AR view of a user. In this article we show for the first time that mobile phones can be utilized to create mobile AR based LBSN services and create an experimental system to validate this. We present the most important use cases of the mobile AR based LBSN services, define the key requirements for the system, and analyze how the current high end mobile phones meet these. We point out the main challenges in position and orientation accuracy, data transfer and power consumption, as well as solutions to improve these. We present results from end-user studies and our experimental system we have created to study mobile AR interface for the LBSN services, and conclude that the mobile phones can be used for creation of these services when the key challenges are resolved.**

***Index Terms*—Augmented Reality; Registration; Context; Location; Mobile; Social Network; User Interface**

I. INTRODUCTION

Augmented Reality (AR) is one of the most promising new User Interface (UI) technologies for applications in mobile phones and other similar devices. AR has been proven to provide benefits over traditional mobile phone UI in user interaction and representation of visual content in mobile phones [1], [2], [3], [4], [5], [6], [7]. AR offers significantly higher degree of immersion than traditional UI and enables efficient completion of tasks with the system. Interacting with the real world is a task that naturally suits the mobile device usage scenarios. Additionally, AR typically has less side effect of nausea than fully virtual systems [8]. Creating a basic video see-through AR view requires less processing power than complete Virtual Reality (VR) view, due to only the annotated objects needing to be generated by the system, which makes it better suited for mobile devices. In the case of mobile AR, these annotations typically indicate either the location of an object, information related to an object or instructive data. The annotations are placed into the view of the user based on the location of the object of interest, location of the user and orientation of the user [9]. In the case of dynamic objects, such as friends in a LBSN service, the end-to-end transmission time needs to be fast enough to provide accurate registration. Because AR is mainly based on reality, the registration of annotations needs to be more accurate than corresponding fully virtual environment, which causes a challenge for implementation of mobile AR services.

In this article we show that mobile AR is suitable for creation of LBSN services using commercially available high end mobile phone devices. Our earlier research on mobile AR has concentrated on enabling mobile AR based personal navigation, global information services, and robot control, as well as evaluation of system level performance of these applications. We have shown that the most feasible path towards consumer mobile AR services is achieved via implementing the services using mobile phone devices, and determined the basic requirements for the devices [6], [7], [10]. Mobile phones benefit from better usability provided by AR, especially when creating services that require the user to map annotations to the surrounding environment, for example personal navigation, and finding the location of items [7], [11].

The basic service of being able to see a set of fixed points of interest as annotations in AR enabled device is already taking its first commercial steps in mobile phones [12, 13]. In the existing mobile AR implementations for mobile phones, such as Layar [12], the objects of interest are stationary, which makes their implementation easier, and requirements for data transmission less demanding than in mobile AR based LBSN services. In another type of AR application, the Google Goggles [13], mobile phone is used for taking a picture of a real world object, which is then used for searching further information about it. This type of mobile AR service does not have any real-time requirements – it is sufficient for the user that the perceived system performance is satisfactory, and the waiting times are not too long. In LBSN services the mobile AR use case becomes much more challenging, because the objects of interest (i.e.

Manuscript Received on 20 July 2010
E-mail:tuukka.turunen@digia.com

friends) are allowed to move, thus introducing the need to dynamic and real-time transfer of data for annotated objects. The use case of LBSN services is often finding out where your friends are, where they have been, what and where they have done (context based status updates), as well as what is near you. Thus the system is required to be able to provide accurate enough registration also for dynamic objects, as we show in this article.

Utilizing mobile AR for LBSN provides significant advantages, as the UI allows the user to see the world via AR view rather than traditional map view. AR makes it intuitive for the user to find friends and services by simply looking at the world via the AR view. Compared to using a map based interface, the direct benefit of mobile AR based interface is that the user no longer has to be able to understand the concept of a map, as he is able to see the virtual items annotated to the correct position in the view of the real world. The benefit of easier usability in mapping the surroundings and immersive experience is expected to be sufficient to drive wide adoption of AR in mobile services as soon as the technology is ready to provide rewarding end user experience, which is already starting to happen for the simple AR services such as being able to see fixed points of interest annotated to mobile AR view [14]. In order to create commercially successful services, the end user experience has to be good, which means that the annotations need to be placed accurately within a margin accepted by the user, annotations should not unnecessarily move around in the AR view, service has to be responsive, and not to consume too much system resources. If these characteristics are compromised, the end user experience degrades and renders the service less desired by end users.

In this article, we first have a look into LBSN and mobile AR at their current state, after that we discuss what mobile AR user interface means for the LBSN services, and determine basic requirements for creation of mobile AR services with good user experience. With these requirements, it is possible to benchmark new mobile AR services and to validate if their performance is satisfactory for providing the desired experience. We present our experimental system called LocTrac, that allows the users to use a mobile phone to view friends and services in their current location respective to the users' own position and orientation via true AR user interface. We benchmark our experimental system against the defined basic requirements and show that current high end mobile phones allow implementation of mobile AR LBSN services, but can not yet provide good user experience in all conditions.

II. LOCATION BASED SOCIAL NETWORKS AND MOBILE AR

Augmented Reality has been studied a lot during the past two decades and applied to several domains [9], [15], [16], [17]. Especially maintenance and service tasks have been commonly found as good application areas for mobile AR [18], [19], [20]. Similarly, other domains in which AR is used to point the user’s focus to a specific part of the real world, such as personal navigation [7], [21], [22] and guidance [23], [24], [25], [26] applications, are commonly seen to benefit significantly from mobile AR based user interfaces. All these applications are generally characterized by annotation of fixed points or items that are moving with the predicted area. In these systems tracking is achieved either via a specific limited area tracking system, utilization of machine vision and markers, or via global coordinates of the item. When the tracked items are stationary, there is no significant real-time requirement for transfer of location info, which is needed when the user or the object of interest are moving. In addition to the basic use cases, mobile AR based techniques can also be used in creation of entertainment applications, such as games [1], [27], [28]. Games have been found to be a good application area for AR research, as they typically do provide quite strict real-time requirement, but also allow the known system level inaccuracies and shortcomings to be hidden from the users, thus allowing to concentrate to the studied items. In the context of LBSN, the games are an important peer group, as many of the networked games are also social networks.

By definition a social network is a social structure made of individuals connected by one or more specific types of relationships [29]. Social network service is defined to be focused on building and reflecting of social networks or social relations among people [30]. Social Network Services are typically Internet based services for managing one’s social networks, that allow constructing a public or semi-public profile within a bounded system, articulating a list of other users with whom they share a connection, and viewing list of connections, and the connections made by others within the system [31]. Recently the Social Network Services have become widely utilized, as people find those valuable and feasible tools to keep in touch with friends and acquaintances. Using these internet based services, it has become much easier to be in contact with people that one does not physically meet. Being able to share experiences is important to individuals, and most persons find this user created and personal content more valuable than such content that has been created for public use.

Current Location Based Social Networks include a number of experimental and commercial services that merge the social circles of the user with context gained from the location of items within the service. Context is defined to mean any information used to characterize the situation of an entity [32]. For creating context, adding location and time typically completes the meaningful context for an item or incident [33], [34]. As context is very valuable to the end user, it is likely that essentially all social networks will contain it to some extent in the future. Currently, the most common examples of LBSN are 1. the possibility to see where your friends are at the moment, 2. receiving a notification when your friend or someone who fits the pre-defined profile is within a range from you, and 3. finding services. It is likely that all social networks will eventually become context aware – and often also LBSN systems.

Augmented Reality brings the value of a new UI to LBSN in a similar way as it enhances the user’s experience in the common

AR applications. For LBSN the main use case of mobile AR is the possibility to see the fixed and moving objects annotated to the AR view of the world. These annotated objects are gained from:

1. Point of interest repositories of different service providers (such as Google), that already contain either specific coordinates or an address that can be converted to coordinates
2. User created content inside the LBSN service, such as personal points of interest (i.e. landmarks)
3. User created content of friends inside the LBSN service
4. Friends' current location as known by the LBSN service

The three first categories are static, while friends are dynamic, i.e. the location of friends changes constantly inside the system. The mobile AR user interface makes it easier for the user to utilize all these categories, and enabling the user to see friends through the AR view offers significant benefits in the friend finding use case.

III. MOBILE AR REQUIREMENTS FOR LBSN

Creation of mobile AR based LBSN service is a lot more challenging than creation of such AR system that only shows fixed points of interest. Especially the possibility for the annotated items to move freely magnifies the challenge of creating the service. This causes the need for close to real-time transfer of information between two mobile users, which is challenging as there always is some level of delay caused by the data transmission, and frequent transmission consumes a lot of system resources. Additionally, AR greatly increases the needed accuracy of the location information compared to web and mobile web based LBSN applications, where it sometimes is enough to know the location based on the current cell where the user's phone is located, or on the internet address used by the home computer. Mobile AR requirement of being able to render both fixed and moving annotations correctly also when viewed by moving user, specifically mandates high accuracy for both position and orientation detection. The applications may use different approaches, especially in the visual appearance, to render the virtual objects of the service, but they all serve the typical use cases and requirements. Examples of different visualization of virtual objects are: using icons or pictures to mark the place or direction towards the annotated object, rendering information such as service details or notes, showing an arrow or similar marker to visualize the path towards an item, and adding animated avatar representation of friends to the user's AR view, just to name some possibilities.

Based on the typical mobile AR LBSN services we can define three categories of use cases:

1. Fixed objects annotated to mobile AR view
2. Moving objects annotated to mobile AR view
3. Supplementary LBSN tasks such as messaging, settings, content creation etc.

Key functionality of annotating objects to the user's view of the world is very similar in LBSN as it is in personal navigation, and we can align registration requirements with the previously well studied personal navigation use case. In mobile AR based personal navigation, the focus has been in showing annotations for specific location, and in guidance applications in showing the direction and path towards the specific point [7], [16], [22]. The main requirements are spatial accuracy and orientation accuracy. When extending the requirements towards annotation of friend locations, the demand for transfer of information with a known and small enough delay becomes significant. In the case of annotating friends both the service provider (i.e. the friend) and the user are allowed to move. Specifically, when implementing the mobile AR LBSN services, these form the basic functional requirements. For practical usability and feasibility with the available technology, there is very important non-functional requirement of power consumption, and the related battery life of the device. Security and privacy are important requirements for any networked system, especially one containing personal information of the user – all LBSN services have stringent requirements for security and privacy. Mobile AR does not, however, significantly add or remove any of these compared to the traditional LBSN services. Thus we do not concentrate on these, or any other important, but not mobile AR specific, requirements in this article. We define the mobile AR specific requirements of Location Based Social Networks in four main categories:

1. Spatial accuracy
2. Orientation accuracy
3. Data transfer
4. Power consumption

Many experimental AR systems can operate only within a fixed usage environment [1], [11], [23], [24], [25], [35], [36], [37]. There it is much simpler to implement accurate positioning, detection of head orientation, and to use markers and machine vision for accurate placement of annotations. Our goal for mobile AR is to take it out from the laboratory and to create globally usable services, which makes it impossible to use the marker based orientation tracking solutions used in many fixed area AR systems. Machine vision and intelligent indexing algorithms can assist in determining the orientation via detecting certain objects such as buildings, provided that there is a global database for objects [4]. Due to its use cases mobile AR LBSN services do not need so precise placement of annotations as, for example, medical applications [9]. Thus it is feasible to rely on less accurate position and orientation detection methods than markers or precision tracking equipment. As it is not possible to use markers, the best approach to create the mobile AR LBSN services is to have sensors for detecting the position and orientation. Most high end mobile phones already provide orientation and position sensors, and the services can rely on the electronic compass and the Assisted GPS (Global Positioning System) or similar, if they provide sufficient accuracy needed for reliable registration of the annotated objects.

Data transmission requirements in mobile AR LBSN services, especially due to the real-time nature of the dynamic information, are significantly different from those in such mobile AR applications that show only fixed items such as information services and personal navigation. In these applications the situation is quite similar to web browsing, in which the user just wants to have fast response times for convenience, but the LBSN use becomes impossible if the information is delayed. As the dynamic objects move, they are not in the same position any more, when the information of their position is received through the system, thus leading to incorrect registration of the annotation. For example, in the use case of two friends looking for each other in a crowd, say outdoor concert or around a city, we need to have fast enough end-to-end transfer of position information in order for these friends to find each other. Due to the non-deterministic nature of movement, using path detection algorithms and giving advance in the registration may lead to incorrect placement of the annotation. In the case of objects moving with walking speed, the delay typically does not prevent friend finding operation in the optimal operating conditions. However, specific focus has to be put into data transfer as well as consequences of frequent transfer such as power consumption and the amount of data traffic. In the case of faster moving objects, or task that requires more precision, the performance can be improved via peer-to-peer communication instead of one user updating the position to server and the other reading it. This reduces the end-to-end transmission time to less than half of what can be achieved in server based setup. Downside of peer-to-peer communication is increased system complexity in a social network use due to the need to share information with multiple friends within the community. Thus, it is preferable to use a server based approach, as long as it can be made to comply with the required performance.

In order to determine how much registration error a user will tolerate in a typical LBSN mobile AR use case of finding a friend, we conducted a test with 30 persons. In this test each person was shown two alternative views through our mobile AR application and asked how much the maximum error they tolerate is in this situation. Each person placed the annotation of a friend into the place in the screen that was the maximum error they can accept. The test views were created so that the situation A was a view from an open area and situation B was a view within the city streets. In both of these situations the friend was located 100 meters away from the user. Field of view was 51°, which is the same amount we have used in our experimental system, described later in this article. We showed the test users where the friend is, marked it with an arrow, and measured the distance of the annotation to the location of the friend. The size of the friend from distance of 100 meters was less than 2 mm in the screen.

The users generally demanded more accurate registration in situation B (street). Only 23% of the test users wanted better accuracy in situation A than in situation B, and out of these 10% were within one meter to the value they wanted in situation A, so it can be stated that 87% of the users wanted same or higher accuracy in the street (situation B) than in an open area (situation A). We can determine that it is natural for a user to desire that the annotation is close enough for him to find the right path to the friend and close enough not to mistake the friend annotation to some other item in the view. Table 1 presents the findings in the test.

TABLE 1: RESULTS OF THE TEST FOR MAXIMUM REGISTRATION ERROR ACCEPTABLE BY A USER.

	Situation A (open area)	Situation B (street)
Acceptable error (average)	6,4°	5,2°
Standard deviation	3,8°	3,4°
Median acceptable error	5,1°	3,7°
Min acceptable error	1,7°	1,1°
Max acceptable error	15,6°	14,6°

Based on these tests we have the result that, on average, the users can accept error of 6,4° in an open area and 5,2° in the street for friend finding use case. There were some users who would accept higher amount of error: 13% said that for them it is enough that the system shows the right direction. But the majority wanted to have accuracy close to the average (if these 13% are not counted, the remaining user base results into standard deviation of 2°). As the situation B (street) requires more accuracy, we shall use it to define the requirement for registration accuracy of mobile AR based LBSN service friend finding use case to be 5°. The other main use cases of mobile AR based LBSN services require similar or less accuracy. Therefore we define the system level requirement for registration accuracy based on the friend finding use case.

Based on these requirements we can calculate, how the inaccuracy accumulates relatively fast causing significant registration problems, and thus determine the requirements for position and orientation accuracy as well as end-to-end data transmission. The total registration error is a factor of four components: 1. positioning accuracy of the user, 2. positioning accuracy of the friend, 3. orientation accuracy (of the user) and 4. transmission delay. Commonly achieved maximum accuracy of 5 m for AGPS positioning and 1.5° for electronic compass orientation already results to 12,6 meters worst case inaccuracy for a distance of 100 meters, i.e. 7,2° maximum error when both users are standing still. As it is a rare case that both users are maximally opposite to each other for the spatial accuracy, we use average error value as the position errors of the users. In the case of 5 m error for positioning of both the user and the friend, and 1.5° error for orientation, this results in combined error of 4,4°. Because users are typically mobile the resulted cumulative error is higher. For a similar setup with the friend moving with walking speed of 5 km/h parallel to the user results in 5,1° average error with end-to-end transfer delay of 1 second. In the otherwise similar scenario, but with the friend moving 40 km/h, combined with end-to-end transmission delay of 1 s, the resulting worst case error already becomes 10,6°. Table 2 presents the errors in typical usage scenarios.

TABLE 2: REGISTRATION ERROR ACCUMULATES BASED ON POSITION AND ORIENTATION ACCURACY AND TRANSMISSION DELAY. DISTANCE TO ANNOTATED OBJECT IS 100 M. OWN SPEED IS 0 KM/H.

Position Accuracy	Orientation Accuracy	Transfer Delay	Friend Speed	Average Error
3 m	1°	1 s	0	2,7°
3 m	1°	1 s	5 km/h	3,5°
3 m	1,5°	1 s	0	3,2°
3 m	1,5°	1 s	5 km/h	4,0°
3 m	1,5°	2 s	5 km/h	4,7°
5 m	1°	1 s	0	3,9°
5 m	1,5°	1 s	0	4,4°
5 m	1,5°	1 s	5 km/h	5,2°
5 m	2°	1 s	5 km/h	5,6°
5 m	1.5°	1 s	40 km/h	10,6°
5 m	1,5°	2 s	5 km/h	5,9°
10 m	1,5°	1	5 km/h	8,0°
10 m	2°	1	5 km/h	8,5°

If the end-to-end transmission delay grows to 2 seconds in the scenario where the friend is moving 5 km/h, orientation accuracy is 1,5° and positioning accuracy 5 meters, the resulting error is 5,9°. In the case of slightly non-optimal accuracy of 10 meters in positioning and 2° for orientation with the friend moving walking speed of 5 km/h parallel to the user, we can calculate the resulting error to be 9,0°. Figure 1 presents an example how the registration error accumulates as a factor of the errors in Position (P) and Orientation (O) accuracy as well as delay in end-to-end Transfer of information (T).

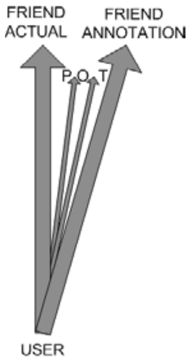


Fig. 1. Annotation of moving objects can not be accurately placed due to errors in Position (P) and Orientation (O) accuracy as well as delay in end-to-end Transfer of information (T).

As an example of how the error accumulates as a sum of errors in Position (P), Orientation (O) and delay caused by Transfer of information (T), we look into the case where positioning accuracy P = 5 m, orientation accuracy O = 1,5°, delay in transfer of friend’s position T = 1 s and friend is moving with speed of 5 km/h parallel to the user with a distance of 100 meters. Position accuracy P = 5 m corresponds to 2,9° error in the angle. Error caused by transmission delay is the distance travelled by the friend in the 1 s it takes to transfer the information about the position to the user, which in this case is 1,4 m corresponding 0,8° error. Thus we gain the total error of 5,2° as sum of P, O and T.

Due to the nature of typical LBSN mobile AR use cases the requirement for registration accuracy for the annotated objects is not strict and minor registration errors can be tolerated. Based on our studies the user is able to tolerate an error of 5° in the friend finding use case LBSN service. The friend finding use case utilizes mobile AR view to provide a path towards the right direction, and the user naturally corrects the direction along the way. It is sufficient that the user approaches the friend towards the approximately right direction and the absolute error gets smaller when the distance is reduced by the friends approaching one another. User experience is hindered if the annotated objects unnecessarily move in the AR view, which shall be avoided to the extent possible. Based on the combined error in registration, the required positioning accuracy needs to be better than 5 meters and orientation accuracy better than 1,5° for mobile AR LBSN applications. We now define the key requirements for mobile AR LBSN solutions as depicted in Table 3.

TABLE 3: REQUIREMENTS FOR MOBILE AR LBSN SERVICES

Operating Environment	Global, Primarily outdoors
Disturbance Sensitivity	Medium
Registration Accuracy	< 5°
Spatial Accuracy	< 5 m
Orientation Accuracy	< 1,5°
End to end data transmission time	< 1 s
Device Size	Pocketable
Device Features	AGPS, Digital compass, Gyroscope (preferably), 3G cellular network connection, Orientation sensor, Display (preferably wide screen), Camera for video see through AR, Sufficient processing power
Power Management	No significant degradation of battery life

Compared to registration related requirements for certain other mobile AR use cases, such as personal navigation and information systems [7, 12, 13], the requirements for mobile AR based LBSN are more demanding. This is due to additional sources of dynamic error based on the end-to-end transfer of information and spatial accuracy of the friend. For the mobile AR based information services, such as display of annotation of information related to a building, there are two sources of dynamic error:

1. spatial accuracy of the user, and
2. orientation accuracy of the user,

Respectively, in the mobile AR based LBSN services there are four sources of dynamic error:

1. spatial accuracy of the user,
2. orientation accuracy of the user,
3. spatial accuracy of the friend, and
4. end-to-end transfer delay.

These, combined with the typical size of the object of interest, cause the registration related requirements for mobile AR based LBSN services to be more demanding. The size of the object of interest is relevant for the user’s perception of the accuracy. As shown, the user desires the annotation to be placed on top or near the object. This causes LBSN to be more demanding, as the small size of a human compared to the size of a building adds the challenge of placing the annotation accurately to correct place in the screen.

Looking at the capabilities of modern cellular phones we can see that some models are already close to the basic AR requirements for LBSN services. For example, Apple iPhone 3GS [38], HTC Hero [39] and Nokia N8 [40] all provide AGPS, digital compass, 3G network connection, orientation sensor, display and camera for video see through AR, as well as sufficient processing power. Optionally to the 3G cellular connection the services may also use Wireless Local Area Network (WLAN) for data transfer. While not available everywhere, there are, however, many city areas where WLAN can be used. The requirement for end-to-end transmissions with WLAN is the same as with 3G, as their characteristics from mobile AR LBSN viewpoint are similar.

Utilization of mobile phone as the device to access mobile AR services brings the benefit of user feeling better immersion due to the real world always being perceived as it is, and not generated virtually [14]. An additional benefit is the possibility to decide, based on the need, how transparent the annotations are. With optical see through it is not possible to make non-transparent annotations, but the video see through approach allows any amount of transparency. Critical mobile phone requirements for mobile AR are accuracy of orientation and position sensors, both of which can reach the mobile AR requirements in optimal conditions. The electronic compass of iPhone, for instance, has nominal accuracy of 1,4° on all three planes [41]. Similarly, AGPS can currently reach positioning accuracy of a few meters in good conditions. The electronic compass alone, however, may not be sufficient to provide reliable orientation information. Typically, the electronic compass provides changing orientation values even when the orientation is not changing due to disturbances in the magnetic fields. This results in difficulty for knowing, which of the values are correct and which are an error caused by some disturbance. The effect of erroneous values can be reduced via effective filtering, but a very extensive filter leads to reduced capability to react to real changes in orientation. A gyroscope can be used to provide more reliable orientation information, but such is not currently available in most mobile phones. For mobile AR based LBSN services it is recommended to equip the mobile phone with gyroscope. In non-optimal conditions also the other sensors are not accurate enough and a disturbance, for example, in positioning accuracy can cause mobile AR services not to provide the expected end user experience.

In addition, if the service is needed also indoors, positioning technologies beyond Assisted GPS are mandatory. Advances in WLAN based positioning systems are likely to make global indoor positioning feasible in the future. For system requirements our basic assumption is that for a friend and object finding use case the requirements for indoor use are similar to the currently studied outdoor scenario. WLAN based positioning technologies with learning capability are seen as the most potential path towards this in the future [42]. With WLAN based indoor positioning fingerprinting approach already provides typically sufficient accuracy of a few meters. For a globally available system the challenge with WLAN based positioning is availability and coverage of fingerprint database.

Other important aspects of implementing mobile AR services with commercially available mobile phones are power consumption and data transmission. When looking into a typical LBSN mobile AR use case of finding a friend we can estimate the power consumption of the device based on the electronic components that need to be active. This service actually needs almost all of the most power consuming components active: display and backlight, camera and camera processor, AGPS, compass, data transfer via 3G modem and significant load to the main processor. The result of this unusually high load is greatly increased power consumption and thus significantly reduced battery life. The transfer rate of 3G cellular network is generally sufficient for transmission of data needed in mobile AR LBSN services, but transfer of real-time data is a challenge. The delays caused by network congestion, protocol overheads, service architecture and implementation need to be managed in order to meet the required end-to-end transmission time. In order to have seamless end user experience and mobility, the service architecture needs to be optimized for mobile AR [10]. To avoid unnecessarily stressing the system, frequent transmission of position information, as well as keeping the positioning sub system of the mobile phone active, shall be conducted only when needed. The system activates these for the user and the friend when the friend finding use case is active, thus allowing the system to save power when less frequent updates of position information are needed.

From the user interface viewpoint we see that mobile AR is a great added value for LBSN services, but not the only needed interface. Due to limitations of mobile AR input methods with the currently available mobile phones, the common browser based approach is much more practical for many typical social network functionalities than using the AR interface. VR and AR techniques provide greatest added value for increasing the feeling of immersion, assistance in navigation and the use of context. For other common functionalities, such as reading messages, updating one's status and settings, the traditional web based interface is well suited. With the increase of mobile phone capabilities it is likely that these supporting functions can also benefit from AR user interface, but the dual approach with mobile web browser and AR view is currently the best solution. With this the user can, for example, get an indication of a friend being near on a traditional UI and use AR to see where the closest friends and services are located. Such dual approach has already been utilized in our experimental mobile AR LBSN system.

IV. EXPERIMENTAL SYSTEM

In order to study the actual use cases of mobile AR user interface for LBSN, we have created an experimental mobile LBSN system that supports AR view for a friend and service finding use cases. This system, called LocTrac, has been built using common web technologies that allow utilization of readily available building blocks and standard interfaces for data transfer. Furthermore, we have prioritized use of open-source components in all parts and built both the server side system and the client on top of commonly known and widely adopted open-source building blocks. We have used HTTP protocol as transport and common web technologies for the main system UI that is accessible with both desktop and mobile browsers. For the AR view we have created a custom client that receives JSON [43] encoded coordinate data over HTTP protocol for frequent updates of the location of annotated items as well as on the need based fetch of graphical items such as annotation icons. Rationale for using JSON array has been to achieve better performance via reduced amount of data to be transmitted compared to XML, while still maintaining the benefits of a standard choice to technology.

Common web technologies and HTTP transport have been used also as the bearer for supporting data, such as information about the services, as well as traditional map navigation functionality. The basic service setup has been based on Apache HTTP server [44] and Glassfish application server [45], both of which are commonly used and provide reliable operating environment for the experimental system setup. System functionality has been implemented with Java Enterprise Edition techniques on the server side. We have utilized MySQL relational database for data storage [46]. For mapping technology we have used Google maps, and the nearby service search has been implemented based on Google’s service point repository [47]. The user created content as well as the position of the friends is provided by the actual users in our experimental system. Figure 2 presents the high-level logical architecture of our experimental system.

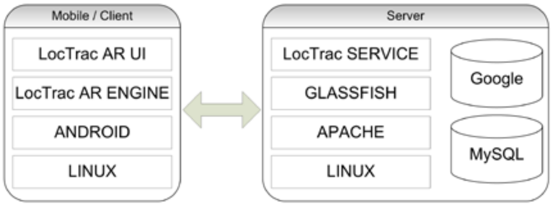


Fig. 2. High-level logical architecture of the experimental system.

We have implemented experimental mobile AR application for the Android mobile platform, but characteristics of the most modern mobile devices would allow similar level of functionality with, for example, Symbian, MeeGo and iPhone OS based devices. The client application has a view controller engine that handles use of the device sensors as well as data transfer with our experimental LBSN server. The application UI is created by view part of the experimental client, which uses display and camera to render the AR view for the user. Figure 3 presents our experimental mobile AR LBSN running on a commercially available HTC Hero mobile phone [39].

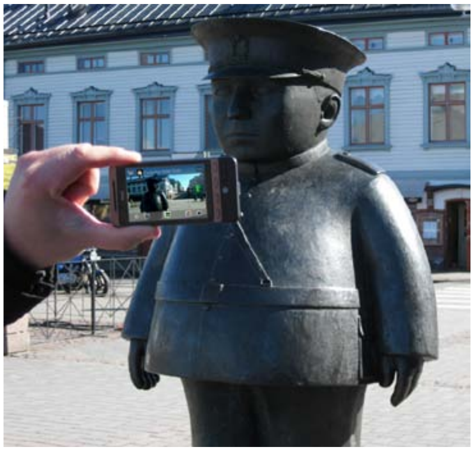


Fig. 3. Experimental mobile AR LBSN service in use.

With the LocTrac service the user can share his location using a mobile client, and see where his friends are by using the web based interface with either a desktop or a mobile browser, or via the mobile AR UI. LocTrac functionality includes all typical LBSN features, such as inviting friends to use LocTrac and sharing own location with friends when desired. Additionally functionality of LocTrac includes seeing the trails of friends, interacting with messaging, boundaries and landmarks, setting boundaries to receive alerts when crossed by friends, searching for nearby services, seeing the friends and points of interest located on the map or AR view, and creating landmarks for the user and friends to see.

For the server side setup and system architecture the web technology based approach gives a user the benefit of being able to access system functionality both with normal web browser as well as with our custom mobile AR application. This dual approach enables the user to utilize basic system tasks with non-AR view that is more user friendly for tasks that require user input, such as composition of messages or creation of landmarks, due to lack of necessary AR input mechanisms in current mobile phones.

When needed, the user can activate the AR based friend or service finding use case by simply lifting the device into horizontal orientation. Then the camera is activated and video-see-through AR view is generated. In the AR view the objects are rendered into the screen on top of the captured video of the direction the user points the device. Information of the location of the augmented objects is received from the LocTrac server in WGS84 format [48]. Based on this information, as well as position and orientation of the user, the system calculates the correct position of the objects in the video see through AR view on the display of the mobile phone. The objects are placed into the center of the view when the device is fully vertical. When the user moves around with the device, the different annotations become visible, thus enabling the user to have a 360° view of the surrounding world with a 51° field of view. We do not use the Z coordinate in our current implementation, but for the future it is beneficial to see also the altitude of the annotated items. Currently we adjust the vertical position of the annotations based on the data received from the device orientation sensor. With this approach we can simulate the missing Z coordinate in most of the typical usage scenarios.

We have created three operating modes for the AR client: by holding the device level with the ground in the hand the user sees a normal map with own location and friends annotated in similar fashion as with the desktop web browser, by lifting the device to about 90° angle in horizontal orientation the AR view is activated and the user sees the objects annotated into video-see-through AR view of the real world, and finally by holding the device vertically the user sees the same data in a traditional list view with a possibility to launch web browser for additional functionality. Example mobile AR views captured from the device screen during operation of our experimental system are presented in Figure 4.

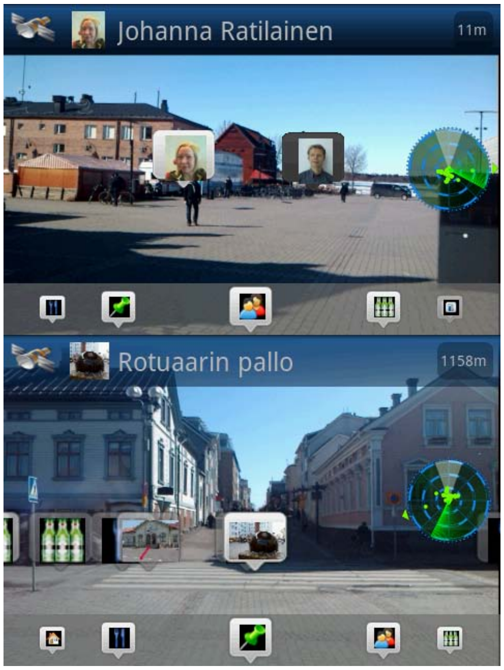


Fig. 4. Two example Mobile AR views with the LocTrac experimental system

In the mobile AR view we have created three main UI components on a full screen video-see-through view of the world. On the top of the screen there is an area with details of the currently focused augmented item. In the middle of the screen the user sees display of the real world annotated with friends, landmarks or services. Distance to the annotated item is shown in meters in the top of the screen alongside with the name of the item. On the bottom of the screen is a carousel type of a control, with which the user can select what category of items are annotated. In the experimental system we have offered three types of categories: Friends, Landmarks and Services. The Landmarks are static items created by the user and his friends. The Friends category is naturally the friends of the user, i.e. moving annotated items. The Services category in the example view includes a search for the nearby hotels, restaurants and bars. In each of these categories the system shows items within the user defined range.

Landmarks and services are non-dynamic by nature. While it is, of course, possible for some new services or landmarks to appear during the use of the system, they are handled as static and non-moving objects. When the user selects a category the items based on this category are displayed with the annotation icons coming from the system based on the images defined by the user and his friends inside the LocTrac service.

Friends are dynamic by nature and annotating them reliably is a key item for the success of any true mobile AR LBSN service. In our experimental system we have taken the approach to keep their data constantly updated during the time the user has the Friends category active in the AR view. Constant updates of the position of a nearby friend can be received with maximum frequency of 10 updates per second in optimal conditions, i.e. it takes less than 100 ms to transfer the data from the server with the experimental system client using 3G cellular network connection when the session is active. For each update the experimental mobile AR client application also sends its current position to the server thus allowing friends to see the position of the user. Naturally the constant updates consume the system resources heavily, but this is the best way to reliably transfer the updates of the position of the annotated objects in our current experimental system. With the maximum frequency of 10 updates per second it takes on average 250 ms to get the information of friend’s location transferred. The additional delay caused by the server is only a few milliseconds and it always sends the most recent data it has of the location of a friend. In our experimental system pilot we have used update frequency of 400 ms as the default value as it results into average end-to-end transmission delay of 1 second. This frequency does not use the maximum resources of the client and provides tolerance to minor network performance problems.

Based on the received information on the position of the annotated objects, as well as on the data received from the AGPS, compass and orientation sensor of the device, the client application calculates the position into which the annotation is placed in the AR view. Typically both the position sensor (AGPS) and electronic compass provide slightly varying values even when the device is not moving. In order to reduce the unnecessary movement of annotations, based on the variation of position and orientation data, our experimental system uses filtering and averaging algorithms. For the position data we use the average of three latest position values from the AGPS when calculating the correct place of the annotations. For reduction of normal variation in orientation sensor data we use recursive noise filtering. In order to discard occasionally appearing sudden large deviations of the electronic compass sensor data the client calculates average of latest six samples and compares the new value to this average. If there is more than 10° difference the value is discarded as the user is not able to make such sudden movements. Even with this filtering we sometimes experience the annotations moving unnecessarily around the screen due to disturbances in magnetic fields causing the electronic compass to provide erroneous orientation values. Having a gyroscope would solve this problem, but such was not available in the HTC Hero device we used in the experimental system. One possibility to improve filtering could be to use the orientation sensor for measuring whether or not the device is moving when the orientation value changes. It is not as good as gyroscope, but could assist in reducing the unnecessary movements of the annotations.

We have conducted a comparative test in order to study how our experimental system fills the user expectations, and to see how the additional error affects the friend finding use case. In this test two peer groups, five persons in each group, performed the same friend finding task. Test area was a small public park with dimensions of 100 by 100 meters. The park contained several pathways in many directions all around the park, as well as a lot of vegetation that prevented direct view to the friend until within a few meters away. In the beginning of each test the friend was located diagonally across the park 140 meters away from the user. The user was instructed to walk towards the friend visible through the AR view of our experimental system. The friend was walking alongside the edge of the park, thus requiring the user to change the direction in order to find the friend. As the users were not able to see the friend during the test, the only indication of the way to walk was from the annotation in the AR view of the park. For each test the setup was similar, but for group B the registration error was made higher by increasing end-to-end transfer delay.

During the test we had position accuracy of average 9 meters provided by the AGPS, and orientation accuracy of 1,5° provided by the electronic compass. Device used in the test was HTC Hero mobile phone connected to a 3,5G cellular network. In order to reduce the possible uncontrolled delay due to network congestion, we used end-to-end transfer time of 2 seconds for group A. This was compensated by reducing the friend's walking speed to 3 km/h, which resulted in 2 minutes walking duration along the edge of the park. The friend was instructed to stop at the edge of the park. None of the users reached the friend before he stopped, but many were very close. The absolute error caused by orientation sensor changed based on the distance, and was on average 1,8 m in the test. The average total registration error for group A was 10,1° as a result of 9 m positioning error, 1,8 m orientation error, 2 s end-to-end transfer time and 3 km/h walking speed. For group B the total average registration error was doubled by increasing the end-to-end transfer time to 20 s, thus resulting to 21,4°. The results of the test are illustrated in Table 4.

TABLE 4. RESULTS OF FRIEND FINDING TEST WITH THE EXPERIMENTAL SYSTEM.

	Group A	Group B
Registration error	10,1°	21,4°
Average time	02:46,2	04:13,4
Standard deviation	00:21,1	00:34,3
Median time	03:15,0	05:07,0
Min time	02:20,0	03:38,0
Max time	02:42,0	04:07,0

The test results very clearly show how the increased registration error affects the friend finding use case. On average it took 52% more time for group B to complete the task. Also it should be noted that none of the five test person in group B were faster than any of the five test persons in group A. The test setup was successful and all ten test persons conducted the test as expected. The experimental system worked well during the tests – there were no problems caused by the experimental system. We only had to restart the devices once between the groups to perform the change of end-to-end transfer time. Due to AGPS position error of 9 m during the experiment, we were not able to reach the registration accuracy of 5°. With the accuracy of 5° we would expect the test to be conducted with even less time than for group A.

The users were asked for feedback after the test and, in general, they liked the system. We did not tell the users that the other group had intentionally higher registration error. Main problem reported by all users was unnecessarily moving annotations caused by changes in the orientation provided by the electronic compass. As discussed earlier the electronic compass is very sensitive to all kinds of errors in magnetic fields, thus providing constantly changing values even when the orientation is not changing. Result of this is unnecessary movement of annotations even after the filtering our experimental system performs. In addition to LBSN service, some test users indicated that they would like to use the application for a hide and seek game.

V. CONCLUTIONS

We have used commercially available high end mobile phone devices as the mobile AR terminal in our experimental system. As we have predicted earlier these devices form a naturally available and familiar way for users to access globally available mobile AR services [6], [7]. Our experimental system proves that it is already feasible to utilize these devices to create LBSN services with mobile AR UI. There exists, however, some limitations that make it difficult to yet achieve wide consumer adoption. The main challenge is related to the current devices only meeting the defined mobile AR requirements in optimal case. If there are buildings hindering the AGPS accuracy, metal structures to disturb the electronic compass, congested network conditions to cause transmission delay, or other processes using system resources, the mobile AR performance degrades rapidly. The inaccuracy caused by these disturbances can be really significant – especially the electronic compass may occasionally point to a completely wrong direction, and typically provides slightly changing orientation values even if the user is not moving. These disturbances can be reduced via filtering and the accuracy can be a bit improved via averaging the values gained from the sensor as explained in this article. Without this filtering the annotated objects can not be accurately placed, as the raw values gained from the orientation sensor are not stabile. For LBSN services the filter for orientation can not be made too strict as it would cause delay in interpreting the real orientation changes. Thus it is recommended that the mobile phones are equipped with a gyroscope to provide better quality of orientation information.

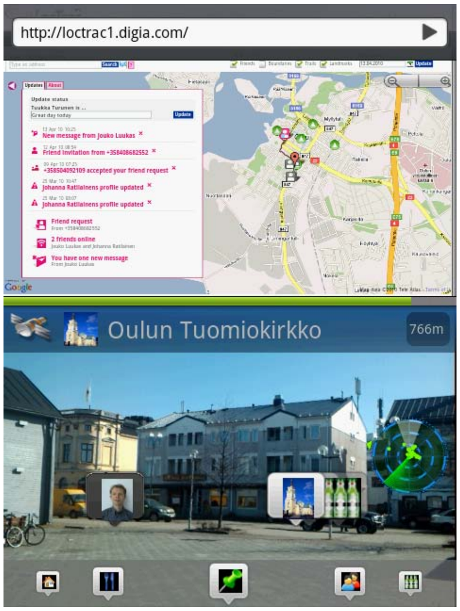
Beneficial to the creation of these services is also the natural ability of a human to correct the error of the annotated object placement during the task of navigating towards it. Even if the object is significantly incorrectly placed in the mobile AR view initially, the user will naturally navigate towards it as long as the systems keeps indicating the location and reduces the amount of absolute error as the user gets closer to the object. Our results show that registration error of 5° can be tolerated by the users of mobile AR LBSN services in the most common friend finding use case. We have presented the requirements for mobile AR LBSN services and that these can typically be achieved in optimal conditions by the commercially available mobile phones. The performance of the sensors and the combined error of position, orientation and transmission results to the accepted registration error. With efficient averaging, and statistically removing erroneous values, the quality of sensor data for both position and orientation can be improved to some extent, thus helping accurate placement of annotated objects and reducing the effect of annotations unnecessarily moving in the AR view, as we have done in our experimental system. To further improve the accuracy and experience we see it as one possible approach to use machine vision to determine which position an annotation belongs to. By averaging the position for a while it is possible to map the annotation based on items from the video stream, thus keeping it in a constant position on the user's mobile AR view. While this approach can assist in keeping the annotations of fixed points of interest stabile, it is however more difficult to apply to LBSN service friend finding use case. Being able to identify a specific human (i.e. the friend) from a long distance is demanding for the camera resolution compared to identification of certain shapes in a building, for example.

Another significant problem with mobile AR LBSN services is the power consumption of operation. Regular use of the mobile AR view quickly drains the battery leading to drastically reduced operating time. Mobile phones achieve long battery life through efficient power saving when the user is not using the device or some part of the device. During the mobile AR use case all of the device's most power consuming components are in essence active and, due to high real-time requirement of the information, it is not easy to save power. Additionally, the friends need to have certain parts of their devices active in order to provide close to real-time information of their location. And in the case of a user navigating towards a friend, the frequency of position updates by the friend's device also needs to be increased, thus resulting in increased power consumption. In order to result to acceptable battery life, the system needs to activate and deactivate the frequent position updates based on need. When a friend finding use case is active, the mobile device of the friend is required to send position updates according to the requirements presented in this article. For times when no one is requesting the frequent position updates, the mobile device can send only periodic updates indicating roughly where the user is.

The user is accustomed with certain applications such as games, video and navigation to consume power, but the power consumption of mobile AR LBSN services is much higher than any of the common high power consumption use cases, so care should be taken in order to create the services in such way that all unnecessary power consumption is avoided. Before reaching wide commercial acceptance, the mobile AR LBSN services must be created in such way that they do not significantly reduce the battery life of the device. We see that the problem with battery life is solved by three main developments. Firstly, the mobile AR LBSN services can be made to consume less power than our experimental system, which is also an area we consider looking more in the future. Secondly, the HW components of mobile devices are continuously evolving to consume less power to provide similar performance. And thirdly, the battery technology is advancing thus being able to pack more energy into the same size and weight.

To achieve seamless end user experience, specific focus has to be put into data transfer due to the real-time requirements of position data. In a mobile AR LBSN system not all data has to be transmitted real-time. There are some items that just need to be efficiently transmitted in order to avoid the user needing to wait for the AR application to initialize, and other items that require close to real-time transmission for the service to be usable. For all non-dynamic data, such as location of fixed items, and item data such as descriptions and images, as well as supporting data of the system, a fast one-time transmission is required. The user accepts similar delays as there are for web pages to load, and goal should be to minimize the waiting time. Better performance is needed to handle the dynamic data related to moving objects such as friends, which needs to be transferred with 400 ms frequency to meet the requirement of one second end-to-end transmission delay when actively used. It is possible to reduce the error caused by the data transmission by estimating the position of the friend at the time the user receives the position information. As the path of the friend is non-deterministic, it is not a good solution to rely too much on an estimated advance, but rather to keep the transmission frequency high enough to provide acceptable end-to-end transmission time for the position information. If the end-to-end transmission time grows above the one second requirement, the cumulative error caused by it and the sensor inaccuracy causes the registration of annotations to contain too much error for the service to be usable. By improving position and orientation accuracy it is possible to utilize less frequent transmission and to achieve the same combined accuracy, or to improve the registration accuracy by keeping the one second end-to-end transfer requirement. Having less frequent position updates directly benefits the energy consumption, and requires less processing power for the application. When the user is not actively using the mobile AR view, or when the friends are far away, there is no need for as frequent transfer – just periodic updates allowing fast start of the application when the user begins to use it. Consequently, the mobile AR LBSN service must understand the situation to provide excellent performance when needed, and to save power and data transfer the remaining time.

Based on our research and the experimental system we have created, we can determine that utilizing mobile AR as UI for LBSN services is possible already to some extent with the commercially available high end mobile phones, thus supporting the studies predicting growth for mobile AR based applications [14]. The initial results from test use of the system indicate that there is significant added value especially for the friend or service finding use case. It is much more natural for a user to look at the world through the mobile AR view in order to find the right path towards an object, than to determine the path from a map. The augmented object does not necessarily need to be an icon annotated on top of a video see-through AR view – it may be an arrow showing the right path, an avatar representing the friend, or a widget displaying additional information. Using mobile AR enables the users to navigate towards a friend or a service in natural way, whereas using a map based UI requires the person to be familiar with the concept of a map, as well as concentration on the task. Figure 5 shows a comparison of mobile web and mobile AR view of our experimental LocTrac service.


http://loctrac1.digia.com/

Update status
Tuukka Turunen is ...
Great day today

13 Apr 10 10:25
New message from Jouko Luukas

12 Apr 10 08:54
Friend invitation from +358408682552

09 Apr 10 07:25
+358504092109 accepted your friend request

25 Mar 10 10:47
Johanna Ratilainens profile updated

25 Mar 10 08:07
Johanna Ratilainens profile updated

Friend request
From +358408682552

2 friends online
Jouko Luukas and Johanna Ratilainen

You have one new message
From Jouko Luukas

Oulun Tuomiokirkko 766m

Fig. 5. LocTrac experimental system allows access to the same service with both mobile web and mobile AR user interface.

Mobile AR view allows users to easily see where the friends or points of interest are located, and thus to determine which way to go. For the other parts of the UI, such as managing the social circles, sharing experiences, reading what friends are doing and similar, the added value of mobile AR is not so significant, particularly with the limitations of currently available input methods of the mobile phones. Certain aspects, especially being able to see context specific user created content in relation to the real world, are clearly benefits of a mobile AR view. When the input methods of mobile phones improve to better accommodate AR, the user can take advantage of the complete system efficiently via an AR interface. Before this it is better to use the system with a dual UI approach such as in our experimental system. Those tasks that are best done with mobile AR, are conducted via the AR interface and others, mainly supporting tasks, are conducted with a traditional UI rendered e.g. as mobile web UI.

ACKNOWLEDGEMENTS

T. Turunen and T. Pyssysalo thank the development team for support, and Digia Plc for permission to write this article.

REFERENCES

[1] D. Wagner, T. Pintaric, F. Ledermann, D. Schmalstieg, "Towards Massively Multi-User Augmented Reality on Handheld Devices", in Proceedings of *the Third International Conference on Pervasive Computing*, 2005.

[2] A. Henrysson, and M. Ollila, "UMAR - Ubiquitous Mobile Augmented Reality", in Proceedings of *the Third International Conference on Mobile and Ubiquitous Multimedia*, pp. 41-45, 2004.

[3] A. Henrysson, M. Billinghurst, and M. Ollila, "Face to Face Collaborative AR on Mobile Phones", in Proceedings of *International Symposium on Augmented and Mixed Reality*, pp. 80-89, 2005.

[4] G. Takacs, V. Chandrasekhar, N. Gelfand, Y. Xiong, W.-C. Chen, T. Bismpigiannis, R. Grzeszczuk, K. Pulli, and B. Girod, "Outdoors augmented reality on mobile phone using loxel-based visual feature organization", in Proceedings of *ACM* international conference on Multimedia Information Retrieval, pp. 427–434, 2008.

[5] P. Antoniac, P. Pulli, T. Kuroda, D. Bendas, S. Hickey, and H. Sasaki, "HandSmart Mediaphone, Advanced Interface for Mobile Services", *World Multi Conference on Systematics, Cybernetics and Informatics, Orlando, Florida, USA*, 2001.

[6] T. Turunen, T. Pyssysalo, T. Lankila, and J. Röning, "Realization of Mobile Augmented Reality Based Services in Cellular Systems", in Proceedings of the *International Symposium of Advanced Distributed Systems*, 7 p, 2000.

[7] T. Turunen, T. Pyssysalo, T. Lankila, and J. Röning, "Realization of Mobile Augmented Reality Based Personal Navigation Services in 3rd Generation Cellular Networks", in Proceedings of the *EuroComm2000 conference*, 7 p, 2000.

[8] M. Anastassova, "User-Centered Design and Evaluation of Augmented Reality Systems for Industrial Applications: Some deadlocks and breakthroughs", *VRIC'07, 9th International Conference on Virtual Reality, Lava, France*, pp. 215-224, 2007.

[9] R.T. Azuma, "*A Survey of Augmented Reality*", Presence, 6(4), pp. 355-385, 1997.

[10] T. Turunen, T. Pyssysalo, and J. Röning, "Performance of a Service Connectivity Architecture for Mobile AR", in Proceedings of *The International Symposium on Information Systems and Engineering, Las Vegas, USA*, pp. 363 – 369, 2001.

[11] S. Henderson, and S. Feiner, "Evaluating the Benefits of Augmented Reality for Task Localization in Maintenance of an Armored Personnel Carrier Turret", in Proceedings of *IEEE* International Symposium on Mixed and Augmented Reality, pp. 135-144, October 2009.

[12] Layar augmented reality browser, Available from: http://layar.com, [Accessed March 15, 2010].

[13] Google Goggles – Use pictures to search the web, Available from http://www.google.com/mobile/goggles, [Accessed June 4, 2010].

[14] ABI research anticipated dramatic growth for mobile augmented reality, Available at: http://www.abiresearch.com/press/1516-ABI+Research+Anticipates+%E2%80%9CDramatic+Growth%E2%80%9D+for+Augmented+Reality+via+Smartphones, [Accessed March 15, 2010].

[15] R.T. Azuma, "Tracking Requirements for Augmented Reality", *Communications*, pp. 50-51, 1993.

[16] J. Rekimoto, "NaviCam: A Magnifying Glass Approach to Augmented Reality", *Presence*, 6(4), 1997.

[17] P. Milgram, D. Drascic, J. Grodski, A. Restogi, S. Zhai, and C. Zhou, "Merging Real and Virtual Worlds", in Proceedings of *IMAGINA '95, Monte Carlo*, pp. 218-230, 1995.

[18] Augmented Reality base technologies for applications in industrial service environments, Available from: http://www.artesas.de, [Accessed March 15, 2010].

[19] W. Friedrich, “ARVIKA-Augmented Reality for Development, Production and Service,” in *International Symposium on Mixed and Augmented Reality*, pp. 3–4, 2002.

[20] J. Platonov, H. Heibel, P. Meier, and B. Grollmann, “A Mobile Markerless AR System for Maintenance and Repair,” in *International Symposium on Mixed and Augmented Reality*, pp. 105-108, 2006.

[21] R.T. Azuma, “The Challenge of Making Augmented Reality Work Outdoors”, Proceedings of *the First International Workshop on Mixed Reality, Tokyo, Japan*, pp. 379-390, 1999.

[22] B. Thomas, V. Demczuk, W. Piekarski, D. Hepworth, and B. Gunther, “A Wearable Computer System with Augmented Reality to Support Terrestial Navigation”, in Proceedings of *the 2nd International Symposium on Wearable Computers*, 1998.

[23] Q. Wang, J. Mooser, U. Neumann, and S. You, “Augmented exhibitions using natural features”, *International Journal of Virtual Reality*, 2008.

[24] A. Damala, P. Houlier, and I. Marchal, “Crafting the Mobile Augmented Reality Museum Guide”. In: 9th International Conference on Virtual Reality. *IEEE, Laval, France,* pp. 303-306, 2007.

[25] D. Schmalstieg, and D.Wagner, “A handheld augmented reality museum guide”, in *IADIS Mobile Learning*, 2005.

[26] S. Feiner, B. MacIntyre, T. Höllerer, and A. Webster, “A Touring Machine: Prototyping 3D Mobile Augmented Reality Systems for Exploring the Urban Environment”, in Proceedings of *the First International Symposium on Wearable Computers*, pp. 74-81, 1997.

[27] T. Starner, B. Leibe, B. Singletary, and J. Pair, “MIND-WARPING: Towards Creating a Compelling Collaborative Augmented Reality Game”, in Proceedings of *Intelligent User Interfaces*, pp. 256-259, 2000.

[28] M. Flintham, R. Anastasi, S. Benford, T. Hemmings, A. Crabtree, C. Greenalgh, T. Rodden, N. Tandavanitj, M. Adams, and J. Row-Farr, “Where online meets on-the-streets: Experiences with mobile mixed reality games”, in Proceedings of *the 2003 CHI Conference On Human Factors in Computing Systems, ACM Press, New York*, 569-576, April 2003.

[29] Definition of Social Network, Wikipedia.org, Available from: http://en.wikipedia.org/wiki/Social_network [Accessed March 15, 2010]

[30] Definition of Social Network Service, Wikipedia.org, Available from: http://en.wikipedia.org/wiki/Social_network_service [Accessed March 15, 2010]

[31] D. Boyd, N. Ellison, "Social Network Sites: Definition, History, and Scholarship", *Journal of Computer-Mediated Communication*, vol 13, issue 1, 2007.

[32] A. K. Dey, “Understanding and Using Context”, *Personal and Ubiquitous Computing*, pp. 4-7, 2001.

[33] B. Schilit, N. Adams, and R. Want, "Context-aware computing applications", *IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, US*, pp. 89–101, 1994.

[34] A. Joly, P. Maret, and J. Daigemont, “Context-awareness, the missing block of social networking”, *International Journal of Computer Science and Applications*, Technomathematics Research Foundation,6(2), pp 50-65, 2009.

[35] J. Rekimoto, and Y. Ayatsuka, “CyberCode: designing augmented reality environments with visual tags”, in *Designing Augmented Reality Environments*, pp. 1–10, ACM Press, 2000.

[36] F. Ababsa, and M. Mallem, “Robust camera pose estimation using 2d fiducials tracking for real-time augmented reality systems”, in *ACM SIGGRAPH VRCAI*, pp. 431–435, 2004.

[37] J. Mooser, W. Lu, S. You, and U. Neumann, “An Augmented Reality Interface for Mobile Information Retrieval”, *ICME 2007* pages 2226-2229, 2007

[38] Apple iPhone 3GS, Available from: http://www.apple.com/iphone/, [Accessed March 15, 2010].

[39] HTC Hero, Available from: http://www.htc.com/www/product/hero/overview.html, [Accessed March 15, 2010].

[40] Nokia N8, Available from: http://europe.nokia.com/find-products/devices/nokia-n8 [Accessed November 15, 2010].

[41] Apple iPhone magnetometer accuracy, Available from: http://www.appleinsider.com/articles/09/05/20/japans_asahi_kasei_to_supply_magnetometer_for_next_gen_iphone.html, [Accessed March 15, 2010].

[42] B. Li, J. Salter, A. Dempster and C. Rims, "Indoor Positioning Techniques Based on Wireless LAN" in *AusWireless '06*, Sydney, March 2006.

[43] JSON Array, Available from: http://www.json.org/fatfree.html, [Accessed March 15, 2010].

[44] Apache HTTP server, Available from: http://www.apache.org/, [Accessed March 15, 2010].

[45] Glassfish application server, Available from: https://glassfish.dev.java.net/, [Accessed March 15, 2010].

[46] MySQL, The World’s most popular open source database, Available from: http://www.mysql.com [Accessed March 15, 2010]

[47] Google AJAX search API, Available from: http://code.google.com/intl/fi/apis/ajaxsearch/, [Accessed March 15, 2010]

[48] World Geodetic System, Available from: http://en.wikipedia.org/wiki/World Geodetic System, [Accessed July 4, 2010].

Tuukka Turunen (1974) obtained the degree of Licentiate in Technology in 2001, and Diploma in Engineering (MSEE) in 1999, both at the University of Oulu in Finland. Between 1996 and 2000 he conducted research on the future of mobile phones, including mobile augmented reality capabilities, at the University of Oulu. During this time he was also member of the faculty of the Department of Electrical and Information Engineering at the University of Oulu. During 1997 and 1998 he worked at Nokia Mobile Phones responsible of certain core cellular system development tasks. From 1999 he has held various research and development, and management positions at Digia Plc. Currently he is Director, Special Projects at Digia Plc, Mobile Solutions business with main responsibility areas of offering development, strategic assignments, and trying out new things. He has written 30 scientific and technical publications including a book, conference papers and articles.

Tino Pyssysalo (1970) obtained the degree of Licentiate in Technology in 1996 and Diploma in Engineering (MSEE) degree in 1994, both at the Helsinki University of Technology in Finland. During 1994 and 1998 he conducted research on telecommunication protocol verification and mobile software. In 1998, he started researching mobile augmented reality from the efficient transport protocol point of view at the University of Oulu, where he held a position of an Associate Professor. During that time, he also became a great fan of Symbian OS and in 2001 he joined Digia Plc, where he has worked as a Senior Software Specialist since then. He is the main author of one programming book and he has published some 20 scientific papers.

Juha Röning (1957) obtained the degree of Doctor of Technology in 1992, Licentiate in Technology with honors in 1985, and Diploma in Engineering (MSEE) with honors in 1983, all at the University of Oulu in Finland. From 1983 he has been a member of faculty of the University of Oulu, where he is currently Professor of Embedded System and head of the Department of Electrical and Information Engineering. He is principal investigator of the Intelligent Systems Group (ISG). In 1985 he received Asla/Fullbright scholarship. From 1985 to 1986 he was a visiting research scientist in the Center for Robotic Research at the University of Cincinnati. From 1986 to 1989 he held a Young Researcher Position in the Finnish Academy. In 2000 he was nominated as Fellow of SPIE. Professor Röning has two patents and has published more than 250 papers in the areas of computer vision, robotics, intelligent signal analysis, and software security. His main research interests are in intelligent systems, especially mobile robots, machine vision, and software security. He is a member of SPIE, IEEE, International Society for Computers and Their Applications (ISCA), Sigma Xi, Finnish Pattern Recognition Society, and Finnish Artificial Intelligence Society (FAIS).