Analysis of Built-in Mobile Phone Sensors for Supporting Interactions with the Real World

Karin Leichtenstern, Alexander De Luca, Enrico Rukzio
Media Informatics Group, Ludwig-Maximilians-University Munich
{karin.leichtenstern, alexander.deluca, enrico.rukzio}@ifi.lmu.de

ABSTRACT
There is currently a lot of research going on in the field of mobile interaction with the real world. So far, the environment where the mobile phone is used is mainly perceived as an unpleasant and disturbing factor. Therefore it has rarely been used as a part of the interaction. But on the other hand there is a huge potential for new kinds of the interactions and novel services. Until now, mostly sophisticated and novel hardware has been used for the development of prototypes. In this paper we investigate which sensors are already built-in in modern mobile phones and analyze how they can be employed in real world interactions. Our focus is on investigating how mobile phone sensors can be accessed using the J2ME platform. We analyze the performance and quality of the recorded media data, and where it can be processed. Finally, we conclude with a discussion on what can already be accomplished with today's mobile phones and which new functions are potentially desired.

Keywords
Evaluation, Sensors, Mobile device, Mobile phone, J2ME, MIDP.

1. INTRODUCTION
It is commonly agreed that mobile devices such as mobile phones, smart phones or PDAs have become ubiquitous in our everyday lives. These devices are equipped with a high-resolution color display, they support different standards of wireless networking and they have reasonable processing power and working memory for a great variety of applications. So far the interaction with those devices occurred mostly between the user, his mobile device and the service he uses (e.g. making calls, writing SMS, browsing the web). The usage of the mobile phone takes place in the context of the real world surrounding the user and his devices. This creates challenges, but also offers potential for new services, as well as novel alternatives for interaction. We think that in future the surrounding physical world (people, places, and things [2]) will play an increasingly important role and the interaction takes place between the user, the mobile device, the service and the surrounding world. Recent developments in the field of mobile services have lead to the development of context-aware services that take for instance the location of the user, his preferences or the current time into account.

In addition to this, we see a lot of research projects which involve implicit and explicit mobile interaction with the real world. The presence of the user could be sensed for presenting personalized information on public display or the user could explicitly interact with artifacts in the real world. Most projects in this field were done with enhanced, big and powerful mobile devices such as a Pocket PC PDAs with additional sensors wired to them because the widespread consumer devices, particularly the mobile phones, did not provide enough resources and sensors for their prototypes. But what about the currently available consumer devices which are sold to and owned by the masses? Which sensors are integrated? What is the performance of these sensors? Which of them can be used for the interaction with the real world? How could one program such applications and services that take the existing sensors into account? What programming platform is supported best or by most mobile devices? Which prototypes can be built with these standard consumer devices? This is especially important to the phone manufacturers as no new devices are likely to be sold – the ones that are deployed are awaiting new applications to generate more revenue.

In the Section 2 we show existing projects that used mobile phone sensors. In Section 3 we present a general architecture which is used by most systems that support mobile interactions with the real world and which is also the basis for our tests. This is followed by an analysis how mobile phone sensors can be accessed with J2ME and what levels of quality and performance could currently be achieved. We focus in this paper on J2ME because this platform is supported by most mobile phones nowadays. Based on this, we conclude what is already possible and which requirements are still not fulfilled by the current generation of mobile phones.

2. RELATED WORK
The development and usage of the various sensors in mobile phones was in the focus of a lot of research projects recently. They gathered sensor data for the support of implicit or explicit interactions and for the development of context-aware services and applications [3, 4].

For instance Schmidt et al. [5] used light sensors, microphones, an accelerometer, a skin conductance sensor and a temperature sensor to predict the user's context. They combined the information they got from the different sensor to high-level context information such as "holing the phone in the hand" or "being in meeting". Hinckley et al. [3] used a proximity range sensor, a touch sensitive sensor and a tilt sensor to develop sensing techniques for mobile phones and combined them for recognizing, for instance, if the user picks up the mobile device.

Another application is sensing the surrounding world with the mobile phone. This can be done by sensors such as cameras, infrared sensors, barcodes- or RFID-readers or microphones. Kindberg et al. [2] developed in the context of the Cooltown project the possibility to discover the services that are related to objects in the physical world. They used infrared beacons which can be recognized by a corresponding sensor in the mobile phone. There exist a lot of other projects or products that allow the picking of physical hyperlinks such as Cybercode [6] (camera) from Sony, Airlic [7] (barcode reader) or Near Field Communication [8] (short-range wireless) from Philips.

Most of the previously mentioned work was done with mobile device that were augmented by new sensors, although there are products and implementations which use standard hardware. Rohs and Gfeller [9] used the camera of a standard mobile phone to interpret two dimensional visual codes that represent an ID. Some 505 series models from NTT DoCoMo have a QR Code reader which is based on the camera of these mobile phones.

It has been shown in a lot of prototypes that sensors in mobile phones can lead to new intuitive ways of interactions and to user friendly context aware service. For this, mostly non standard mobile device were used. Some projects already showed that sensors of standard mobile phones can be used for this.

3. GENERAL ARCHITECTURE

In this section we discuss based on the following figure the general relationship between the real world, the mobile phone and the services that are related to the real world objects.

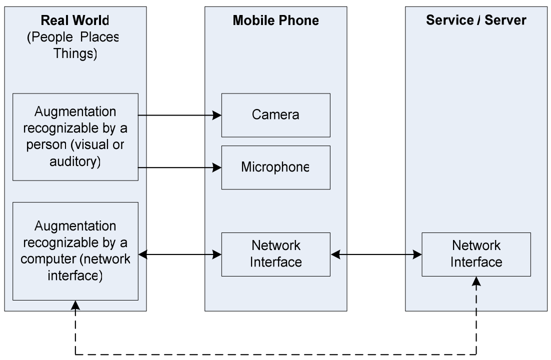

Figure 1. General architecture.

There is no such thing as a definite set of objects in the real world that can be used for mobile interactions. In general, it is possible that any object is augmented (e.g. by a marker or a RFID chip). We focus in this paper on augmented objects because this simplifies the interaction as things like recognizing the name of a building just based on its appearance is still very complicate for specialized hardware. On the one hand, the augmentation could be done in a way that is visible or audible by a human. On the other hand it can be done in a way that is only recognizable by a mobile device. This is a decision that has to be made depending on the actual requirements.

In general, a modern mobile phone has three different built-in sensors: a camera, a microphone and network interfaces (e.g. Bluetooth, GSM, UMTS). For interactions with the real world, one of these sensors must be used to establish a connection between the real world object and the mobile phone. Hereby the real world object could provide static (marker on an advertising poster) or dynamic (public display) information. Most often, interactions are related to a service that is provided by a server or to an application that is already installed on the mobile phone. In principle, it is possible, too, that the real world object augmented by a network interface can directly interact with the service provided by a server.

One possibility for interactions with real world objects is the usage of Bluetooth which is already feasible with the Bluetooth API JSR 82 [13], available on most mobile phones that support J2ME and have a Bluetooth interface. By this it is possible to build interactions with people that are in a defined vicinity to a real world object. New kinds of interactions could be established by exploiting this, but in this paper we focus on the usage of the camera and the microphone.

4. RECORDING MEDIA WITH MMAPI

The platform which is currently supported by most mobile device is the Java 2 Micro Edition (J2ME) [14] with 1.7 billion enabled devices [1]. The main reason is that J2ME is platform independent and nearly all operation systems on mobile devices such as Symbian, Palm OS, Windows Mobile and most mobile phone vendor specific operation systems support J2ME.

The J2ME Mobile Media API (MMAPI) [12] is the only option in J2ME / CLDC / MIDP to address the camera or the microphone of a mobile phone. J2ME is divided in different configurations, profiles and additional APIs. For mobile phones, the configuration ‘Connected Limited Device Configuration’ (CLDC) and the profile ‘Mobile Information Device Profile’ (MIDP) must be used. The basic functionality of the MMAPI is supporting, recording and playing of audio or video data.

In this test we analyzed the performance of the camera while taking pictures and of the microphone while recording audio information. We developed a corresponding J2ME application which can be downloaded at [11]. At the webpage you can also find more test results and samples of the data we got from the sensors. With the programs showed at [11], it is in a first step possible to query which media types in which encodings can be recorded by the mobile phone. Based on this information it is possible to adapt the other programs that test the camera as well as the microphone. These programs were used for the tests explained afterwards. In general, the results of the tests are displayed on the mobile phone screen and/or transmitted to a server for verifying the quality and memory size of the recorded data. We used the mobile phones Siemens S65 and Nokia 6600. We do not compare directly the test results of these two mobile phones because we want to present general results which are valid for more modern mobile phones.

This evaluation discusses the performance of the sensors in a general way, because different mobile phones support different formats and allow different parameter settings. We focus on the formats and parameter settings that are supported by the most devices.

4.1 Image

In this evaluation we concentrated on to test the JPEG format because most mobile phones support this format. Photos stored in the JPEG format are typically used for real life pictures. JPEG pictures offer both a relatively good quality and a moderate memory size. Other formats like BMP or GIF might be more suitable for analyzing the content of the picture but BMP compresses very badly which leads either to a memory consuming or a poor image quality. GIF or PNG should not be used for real live pictures.

4.1.1 Memory size of pictures

In the fist test we wanted to determine the memory size and the quality of the taken pictures on a mobile phone in the JPEG format. Particularly for applications in the field of computer vision, like marker detection, a good quality of the taken pictures is necessary. For the first test a J2ME program [11] was implemented to take pictures in different kinds of resolutions. Pictures were taken in resolutions of 80x60, 160x120, 200x150, 320x240, 640x480 and 1280x960 pixels.

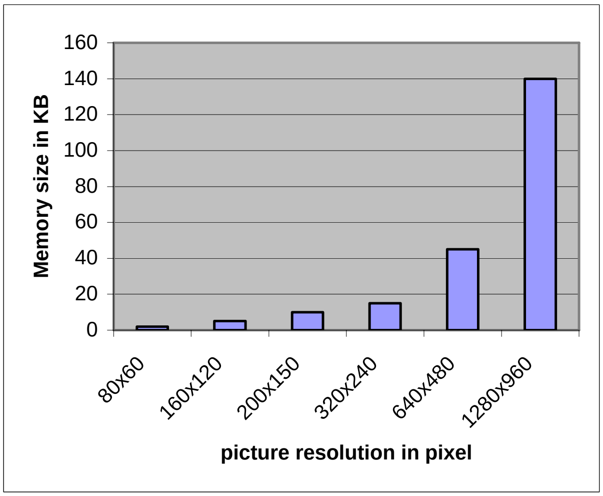

Figure 2. Memory size of different resolution for JPEG.

Figure 2 shows the relatively moderate memory size consumed by the taken pictures. In small resolutions up to 200x150 pixels the memory size did not exceed 10 Kilobytes. Even in the highest resolution of 1280 x 960 pixel the memory size reached only 140 Kilobytes. Beside the moderate memory size the pictures offer also a good quality. Regarding the memory size there was not noteworthy difference between the two phones beside the fact the maximal resolution of the Nokia 6600 is 640x480 pixels.

The quality is already in good region to use the picture for many kinds of detections. The moderate memory size allows to take and to store pictures on the memory card. Depending on resolution, a number of pictures can be stored. Currently J2ME applications for the interpretation of pictures on a mobile phone are relatively slow. Moreover, J2ME does not support some kinds of APIs like Java2D. Therefore an image interpretation, which needs the unsupported APIs, has to be done on a server using e.g. J2SE.

4.1.2 Speed performance test

In the second test we wanted to analyze the speed factor. We evaluated how many pictures could be taken in a given time. We decided to take photos in time from one up to ten seconds. In this way we wanted to verify if increasing time changes the number of taken pictures per second. Another variable is the resolution of the taken pictures. We took photos in resolutions 80x60, 160x120 and 200x150 pixel.

The following Figure 3 shows how many pictures could be taken per second in different resolutions. The results shows that as higher the resolution of the picture as decreasing the numbers of taken pictures per second. Moreover, the diagram shows the trend that in the small resolution of 80x60 pixel approximately 1.7 pictures per second could be taken. In the resolutions of 200x150 pixel the trend for the number of taken pictures per second was about 0.8. The overall interpretation of the results is worse. Despite the relatively small resolutions of the pictures the speed performance test shows bad results.

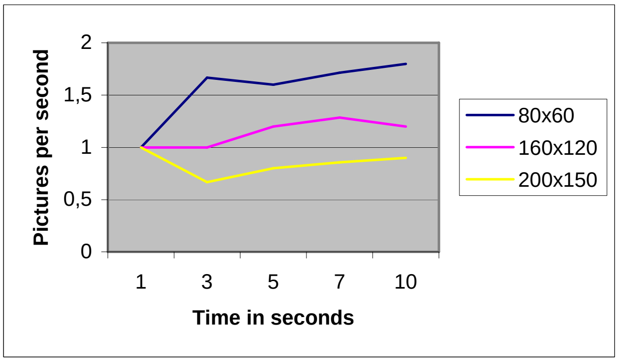

Figure 3. Pictures per second in different resolutions.

Real time applications such as movement detection directly on the mobile phone written in J2ME are currently not possible as not enough pictures may be taken in a given time.

4.2 Audio

In this evaluation we tested the quality of the microphone of the mobile phone when accessed by the MMAPI of J2ME. According to the MMAPI specification audio data in different quality levels from 8 bit / 8 KHz (quality per sample / sample rate) to 24 bit / 96 KHz could be recorded whereby the upper limit makes probably no sense when used with the current microphones because they are constructed for recoding speech. If choosing higher sample rates the file size increases a lot whereas the hearable quality increases only quite a bit.

For the test we used the encoding formats Pulse Code Modulation (PCM) and Adaptive Multi Rate (AMR). PCM is normally used in digital telephone systems. AMR is a lossy Audio data compression scheme optimized for speech coding. These encodings allow multiple applications. AMR for example could be used for voice recognition.

Figure 4 depicts the file size of recorded audio data for 5 seconds when using the encoding PCM and AMR for the bit rates 8 and 16 bit.

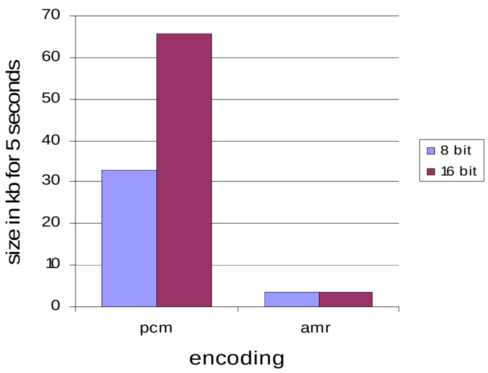

Figure 4. Memory size of different encodings.

The result of this test can be compared to the camera test. The file sizes are in a region that allows in general their processing by the mobile phone. Unfortunately, there is, too, no API available for the interpretation and modification of audio date which postpones this task probably to the server.

The file sizes for the encoding AMR show a typical effect that we recognized particularly during the evolution of the microphone. In principal there are a lot of possible parameters when recording audio data such as encoding, sample rate or bits per sample. Often there exist restrictions regarding these parameters that are not known by the programmer. In the Figure 4 for instance, the mobile phone did support only 8 bit per sample and not 16 bit per sample. This can be only recognized when looking on the file size or by analyzing the generated audio file afterwards.

5. CONCLUSION

In our analysis we found out that in most of the cases the quality of the sensor data is sufficiently good. With respect to the performance, the knowledge about the recorded quality, the processing of the data and the development of platform independent applications for interactions with the real world, there are still considerable problems.

The performance when accessing the camera is currently not sufficient for supporting gestures where about 10 to 15 frames per second are needed because we showed in our test that currently at maximum 2 pictures per second could be recorded.

Another problem is the lack of the knowledge on the quality of the recorded data. When building applications that should run on a lot of mobile devices it is very complicate to test every mobile phone before delivering the mobile phone application or to integrate a test routine that runs after the installation of the program on the mobile phone. It is currently already possible to ask the mobile phone for the supported audio or video encodings. But it would be a great advantage for this kind of programs to know which parameters (e.g. resolution, bit per sample, sample rate) are supported for a specific encoding on a mobile phone. Unless such simple or even more sophisticated methods regarding the quality of service are not present, it is very complicated to develop general programs that use sensor data and which run on several mobile phones.

The next step after recording of the media is the analysis of the audio or video data. Currently there are no APIs available for this task on the mobile phones because J2ME only includes the basic APIs that are needed for the most common applications. Hence the developer can integrate some classes for the interpretation of the sensor data in the program on the mobile device or transmit the data to the server and analyze there the sensor data. The first option has the advantage that no data must be transmitted. But for the processing of the media data on the mobile phone there is often not enough memory or processing power available. The advantage of the transmission of the media data to a server is that a server has a potentially huge set of memory and enough processing power. The disadvantages of this approach are the costs for data transmission and a delay which is not helpful for the most kind of interactions.

Our conclusion is that the development of prototypes for mobile interactions with the real world is already possible when using modern mobile devices whose capabilities are tested intensively before. But we have to wait a few years for matured implementations of the corresponding APIs on the mobile phones, as well as for more processing power and working memory on the mobile phones to support real world interactions.

6. ACKNOWLEDGMENT

The authors wish to express their gratitude to the other participants of the course "programming of applications for mobile devices", to Nokia and Siemens for lending us mobile phones and to Albrecht Schmidt for their kind support. This work was performed in the context of the framework of IST Project Simplicity funded by the EU [15] and the DFG funded research group "Embedded Interaction" [10].

7. REFERENCES

[1] Report on Heise.de from the 25/10/2004 – News page of a German IT publishing house, http://www.heise.de/newsticker/meldung/52477

[2] Kindberg, T.; Barton, J.; Morgan, J.; Becker, G.; Caswell, D.; Debaty, P.; Gopal, G.; Frid, M.; Krishnan, V.; Morris, H.; Schettino, J.; Serra, B.; Spasojevic, M. 2002. People, Places, Things: Web Presence for the Real World. In: *Mobile Networks* and Applications, 7(5):365–376, 2002.

[3] Hinckley, K.; Pierce, J.; Sinclair, M.; Horvitz, E. 2000. Sensing techniques for mobile interaction. In: *UIST '00: Proceedings of the 13th annual ACM symposium on User interface software and technology*, pp. 91—100, San Diego, California, United States, 2000.

[4] Gellersen, H.; Schmidt, A.; Beigl, M. 2002. Multi-sensor context-awareness in mobile devices and smart artifacts. In: *Journal Mobile Networks and Applications* 7, pp. 341–351, Kluwer Academic Publishersm, 2002.

[5] Schmidt, A.; Aidoo, K.; Takaluoma, A.; Tuomela, U.; Van Laerhoven, K.; Van de Velde, W. 1999. Advanced Interaction in Context. In: HUC '99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing, pp. 89-101, Karlsruhe, Germany, 1999.

[6] Rekimoto, J.; Ayatsuka, Y. 2000. CyberCode: Designing Augmented Reality Environments with Visual Tags. In: *Proceedings of DARE*, 2000.

[7] Airlic www.airclic.com

[8] Near Field Communication from Philips Semiconductors http://www.semiconductors.philips.com/markets/identification/products/nfc/

[9] Rohs, M.; Gfeller, B. 2004. Using Camera-Equipped Mobile Phones for Interacting with Real-World Objects. In: *Alois Ferscha, Horst Hoertner, Gabriele Kotsis (Eds.): Advances in Pervasive Computing, Austrian Computer Society (OCG),* ISBN 3-85403-176-9, pp. 265-271, Vienna, Austria, April 2004.

[10] Research Group "Embedded Interaction", http://www.hcilab.org/

[11] J2ME Application for testing the sensors http://www.hcilab.org/resources/tutorials.htm

[12] Mobile Media API, http://jcp.org/en/jsr/detail?id=135

[13] JSR 82, http://www.jcp.org/en/jsr/detail?id=82

[14] J2ME, http://java.sun.com/j2me/

[15] Simplicity Project, http://www.ist-simplicity.org