

MOBILE AUGMENTED BARCODES: EXPERIENCES WITH A NOVEL MOBILE BARCODE SCANNER IN THE WILD

ABSTRACT

Mobile phones with built-in cameras are increasingly investigated mediators for interacting with real-world surroundings and fetching annotated digital information. In this paper, we introduce a novel mobile interaction approach exploiting common barcodes attached to consumer products such as books and CDs. The proposed mobile application follows a “Magic Lens” metaphor to directly overlay barcodes with requested digital information facilitating a quick comparison of products. We evaluated the prototype in a real-world setting and outline our experiences from these tests.

KEY WORDS

Mobile HCI, barcode scanner, augmented reality

1. Introduction

Nowadays, nearly all consumer products like books, CDs or DVDs are marked with barcodes to quickly identify them at the cash desk and retrieve data that is linked to the scanned barcode, e.g. purchase and sales prices or remaining stock level. In a similar vein, barcodes may also link to further information interesting for consumers, like book reviews or user ratings on appropriate Web sites. To harness this data a barcode reader and Internet access is needed. Modern mobile phones are capable of both, scanning barcodes with integrated cameras and downloading the desired content via the Internet.

The novel interaction approach described in this paper follows the “Magic Lens” metaphor [1] for augmenting a bar code with attached digital information. The basic idea is to visually overlay the barcode with whatever information the consumer is interested in to make gathering the information as easy as looking at the barcode “through” the mobile phone.

This paper describes a prototypical application for camera-equipped mobile phones that is able to decode ISBN (International Standard Book Number) barcodes using the live camera image and to overlay them with information provided by the Amazon E-Commerce Services. The primary use-case for the “Mobile Augmented Barcode Scanner” is to serve as tool for consumers to easily compare books by their average rating and price as listed in the US Amazon store. We evaluated this prototype in a field test in terms of overall application speed and responsiveness, barcode detection rate and energy consumption.

The remainder of this paper is structured as follows. In Section 2, we present some related work in the field of mobile interaction with the real world. Section 3 discusses our prototypical application called Mobile Augmented Barcode Scanner. In Section 4, we describe the prototype’s evaluation in a real-world setting. Finally, we draw concluding remarks in Section 5.

2. Related Work

Mobile phones with built-in cameras are increasingly investigated mediators for interacting with real-world surroundings and fetching annotated digital information.

In early work, Rohs [2] presented a way to use 2-dimensional visual codes a) to determine the code value and b) to calculate additional parameters, such as the distance to the mobile phone and its 3-D position. The processing of these visual codes is entirely performed on the phone itself. A wireless Internet connection is used to download online content that is linked to the sensed code. Additionally, actions on near devices, such as displays, can be triggered. These features enable the mobile phones to act as a bridge between real-world objects and their associated entities in the virtual world.

Rukzio et al. [3] also investigated marker-based interaction methods and compared them with several other mobile interaction techniques. Their prototypes are implemented with different technologies such as RFID tags, light sensors and visual codes. Depending on the underlying sensor technology different interaction approaches are used, e.g. pointing or touching objects or scanning of markers.

In contrast to such marker-based approaches, also natural image features can be used to identify photographed objects without the need of an explicit marker. Pielot et al. [4] presented an application to retrieve information and invoke services related to a photographed poster. The primary goal is to eliminate the need for user interaction through virtual or multi-tap keyboards, e.g. to operate online search engines to retrieve content-related information. The image analysis is performed on a central server using the Scale-invariant Feature Transform algorithm [5]. The content-related information is then transmitted back to the mobile phone.

Natural features can be exploited for real-time augmented reality applications even on mobile phones. E.g. *Studierstube Tracker* [6] is a computer vision library for

detection and pose estimation of various 2D fiducial markers in real-time. The framework was designed for high performance and low memory footprint. It is available for both PCs and mobile phones. Though Studierstube Tracker supports a variety of 2D markers (e.g. DataMatrix, Split and Grid Markers), it does not detect 1D markers .

In the meanwhile, mobile barcode scanners such as *CompareEverywhere*¹ are available in several stores for mobile applications. However, an evaluation of such an application in real-world settings has not been conducted so far.

3. Mobile Augmented Barcode Scanner

The proposed Mobile Augmented Barcode Scanner is designed as an easy-to-use tool for directly retrieving basic information about books within a store without having to explicitly search the Internet. The application aims at eliminating the need for user input in order to fetch relevant information like average user ratings or prices. As shown in the design concept in Figure 1 the live camera image is augmented with basic information. The barcode itself is just barely visible.

The primary use-case is not to get all information that is linked to a given ISBN, e.g. a summary, authors or awards, but let the user first select a desired information module for a quick product comparison. Currently, two modules are implemented: one provides user ratings, the other displays the price (see Figure 2).

3.1. Platform

Our Mobile Augmented Barcode Scanner prototype is implemented using Android 1.6 for the mobile phone “HTC Dream”. Android provides a comfortable application programming interface for accessing the built-in camera. The platform further offers an automated garbage collection and seamless remote debugging.



Figure 1. A first concept of how the ISBN barcode should be superimposed with the book’s average rating as listed in the US Amazon store.

¹ <http://cyrket.com/p/android/com.compareeverywhere/>



Figure 2. A screenshot of the prototype showing how the barcode is being superimposed by a price overlay.

3.2. Scanning

The decoding of barcodes within the live camera image is done by the open-source ZXing barcode image processing library² designed to detect various 1D and 2D barcodes. As for the implemented prototype only 1-dimensional ISBN barcodes need to be processed and to optimize the performance, we customized the library and removed the support for other barcodes. The main disadvantage of ZXing is that distorted or rotated barcodes cannot be detected properly.

3.3. Tracking

As the barcode should always be correctly superimposed with the corresponding overlay, it is necessary to track the movement of the barcode within the live camera image. The ZXing library provides such a rudimentary tracking functionality. Though, the tracking is not very accurate, e.g. when rotating the phone the barcode cannot be detected and hence not properly overlaid anymore. Furthermore, ZXing’s tracking functionality does not work at interactive frame-rates. If the mobile phone is moved to fast there is a noticeable lag until the overlay is drawn at the correct position.



Figure 3. Scenario 1: The shelves are illuminated by both natural and artificial light causing only minor shadows.

² <http://code.google.com/p/zxing/>

4. Evaluation

To evaluate our barcode scanner in a real-world setting, we ran a field test in the public library of the Vienna University of Economics and Business. Its main purpose was to evaluate the overall speed and responsiveness of the prototype's user interface and the barcode detection rate and speed. Furthermore, we assessed the application's energy consumption.

4.1. Methodology

The field test was run by one of the developers familiar with the prototype. On the one side we measured the time needed to decode a barcode in the live camera image. And on the other side we assessed the amount of time that was necessary to download the corresponding data from the Internet and to superimpose the barcode.

The decode speed measurement was manually started by pressing on of the mobile phone's buttons as soon as the barcode was visible on screen. The measurement was stopped immediately after the ZXing library returned a valid ISBN barcode. Thereby, the next measurement was automatically triggered beginning straight before the data download and stopping as soon as the overlay was displayed on screen.

Furthermore, we measured the mobile phone's energy consumption during the field test.

To acquire realistic results the presented application was tested in three different scenarios.

The first scenario as seen in Figure 3 was about testing the prototype application in bright surroundings with nearly no shadows or unequally illuminated spots. The bookshelves were illumination by both natural and artificial light.

The aim of the second scenario was to test the barcode detection rate in a poorly illuminated place like shown on Figure 4. The light was unequally distributed on the bookshelves. Furthermore the ceiling lights caused major shadows and reflections.



Figure 4. Scenario 2: The shelves are only illuminated by ceiling lights causing many shadows.



Figure 5. Scenario 3: Neither natural nor artificial light illuminates the bookshelves.

In third scenario we evaluated the performance in dark surroundings without any proper lightning (Figure 5).

4.2. Results

During the field test in total 123 ISBNs were scanned - 57 barcodes in the first scenario and 66 in the second one.

As the used mobile phone did not include a flashlight, it was impossible to properly detect a single barcode in the third scenario. Therefore the results that are discussed in the following only refer to the first two scenarios.

Table 1 shows the summarized results of the field test. The average amount of time needed to detect a barcode in the live camera image, download the necessary data from the Internet and overlay the barcode was about 3,66 seconds.

Scanned books:	123
Different ISBNs:	108
Average time needed to detect a barcode on the live camera image:	1,89 sec
Average time needed to download the data over UTMS and to display the overlay:	1,77 sec
Total:	3,66 sec

Table 1. Overview of the results of the field test.

As shown in Figure 6 the amount of time needed to detect a barcode was nearly equally distributed between both scenarios. Furthermore, in six cases the time needed to decode a barcode was greater than seven seconds. These detection anomalies were caused by bad lightning conditions (partial shadows, highlights).

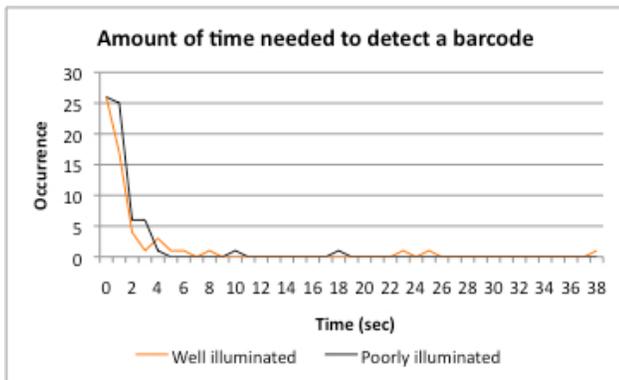


Figure 6. 90% of all barcodes were detected in less than 4 seconds.

Figure 7 shows that the data download, e.g. of ratings or prizes, and the process of overlaying the barcode in the live camera image took less than less than two seconds in 74% of all test cases. For the remaining 26% the data download took up to five seconds.

4.3. Discussion

To get best possible data regarding the energy consumption of the Mobile Augmented Barcode Scanner, the mobile phone's battery has been fully charged and turned on straight before the field test began. After 1h 4m of using the prototype application the first low battery warning (charge < 15%) was shown on screen. So approximately there were 11 minutes left before the phone would have shutdown itself.

This massive energy consumption was primarily caused by the display that was on all the time, though its brightness was turned to the lowest possible value. But also the high processor load of the barcode decoder and the constant network traffic drained a lot of energy.

The ZXing barcode library and the Mobile Augmented Barcode Scanner itself could be optimized to reduce workload and hence reduce battery usage. But as the application cannot be properly used without display and wireless Internet connection the overall energy savings potential is quite small.

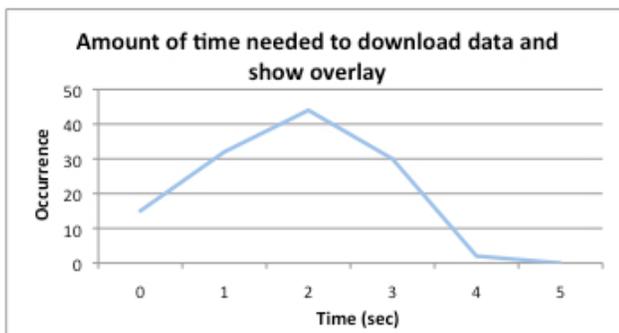


Figure 7. In 74% of all test cases it took less than two seconds to download the data and show the overlay on screen.



Figure 8. If the difference in brightness within the barcodes surpasses a certain value, e.g. caused by highlights (left image) or partial shadows (right image), it cannot be detected properly anymore.

During the field test furthermore we recognized some barcodes can only be detected by ZXing if certain requirements are met. In general, the barcode that is being scanned has to be equally illuminated. If there is a high difference in brightness within the barcode image as shown in Figure 8, ZXing cannot properly decode barcodes anymore. In such case, the detection can take up to 40 seconds. However, the time needed for the data download remains unaffected.



Figure 9. Barcodes of the size of a thumbnail or smaller cannot be detected correctly because the thin bars on the image are too blurry.

Furthermore, we found out that Zxing is unable to decode very small barcodes as shown on **Fehler! Verweisquelle konnte nicht gefunden werden..** The smallest movement of the mobile phone causes the bars on the image to be blurry and hence undecodable.

During the field test there were only 6 barcodes that could not be detected within less than 7 seconds due to very bad conditions such as describe above.

5. Conclusion

The overall speed and responsiveness of the Mobile Augmented Barcode Scanner prototype is adequate to guarantee a fluid user experience.

Currently the two remaining drawbacks are the high amount of energy the prototype application consumes and the lack of a full-fledged tracking functionality.

With the arrival of new display technologies like OLED the energy consumption might be significantly reduced. Additionally it might be necessary to automatically close the application in case the user does not use it for some time. At the same time it must be ensured, that the startup

of the application is as short as possible to guarantee the least possible distraction for the user.

Furthermore, it might be possible to reduce the energy consumption caused by the wireless Internet connection by caching the downloaded data. In case the user scans the same ISBN again within a certain amount of time, the data does not have to be re-downloaded but is directly retrieved from the cache.

References

- [1] E.A. Bier, M.C. Stone, K. Pier, W. Buxton, T.D. DeRose, Toolglass and magic lenses: the see-through interface, *Proc. 20th SIGGRAPH annual conference on Computer graphics and interactive techniques*, New York, NY, 1993, 73-80.
- [2] M. Rohs, Real-world interaction with camera-phones, *Proc. 2nd International Symposium on Ubiquitous Computing Systems (UCS 2004)*, Tokyo, Japan, 2004, 74–89.
- [3] E. Rukzio, G. Broll, K. Leichtenstern, A. Schmidt, Mobile Interaction with the Real World: An Evaluation and Comparison of Physical Mobile Interaction Techniques, *Proc. European Conference on Ambient Intelligence (AmI-07)*, Darmstadt, Germany. 2008, 7-10.
- [4] M. Pielot, N. Henze, C. Nickel, C. Menke, S. Samadi, S. Boll, Evaluation of Camera Phone Based Interaction to Access Information Related to Posters, *Proc. 3rd International Workshop on Mobile Interaction with the Real World (MIRW)*, Amsterdam, Netherlands, 2008.
- [5] D. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, *International Journal of Computer Vision*, 60(2), 2004, 91–110.
- [6] D. Schmalstieg, D. Wagner, Mobile Phones as a Platform for Augmented Reality, *Proc. IEEE VR 2008 Workshop on Software Engineering and Architectures for Realtime Interactive Systems*, Reno, NV, 2008, 43-44.