USING CAMERA-EQUIPPED MOBILE PHONES FOR INTERACTING WITH REAL-WORLD OBJECTS

Michael Rohs,* Beat Gfeller

Abstract

The idea described in this paper is to use the built-in cameras of consumer mobile phones as sensors for 2-dimensional visual codes. Such codes can be attached to physical objects in order to retrieve object-related information and functionality. They are also suitable for display on electronic screens. The proposed visual code system allows the simultaneous detection of multiple codes, introduces a position-independent coordinate system, and provides the phone's orientation as a parameter. The ability to detect objects in the user's vicinity offers a natural way of interaction and strengthens the role of mobile phones in a large number of application scenarios. We describe the hardware requirements, the design of a suitable visual code, a lightweight recognition algorithm, and present some example applications.

1. Introduction

This paper describes a visual code system for camera-equipped mobile phones. A growing number of commercially available mobile phones come with integrated digital cameras to take snapshots and send them as MMS messages. Even though the image quality of these cameras is (still) comparatively poor, such devices can perform simple image processing tasks. The proposed system lets camera-equipped mobile phones act as sensors for 2-dimensional visual codes. The codes can be printed on paper documents, displayed on electronic screens, or attached to physical objects and act as a key to access object-related information and functionality.

Enhancing mobile phones in this way is promising, because they are in constant reach of their users and are thus available in many everyday situations. They provide continuous wireless connectivity, and models with integrated cameras are getting more and more popular. The ability to detect objects in the user's vicinity strengthens the role of mobile phones in m-commerce, education, and gaming scenarios. It offers a natural way of interaction and makes data entry more convenient. The mobile phone becomes a kind of "bridge" between entities in the real world and associated counterparts in the virtual world.

The visual code system presented in this paper is designed for the low image quality of mobile phone cameras. In contrast to other systems, the recognition algorithm simultaneously detects multiple codes, determines the orientation of the codes, and computes the coordinates of the target aimed at

*Institute for Pervasive Computing, Department of Computer Science, ETH Zurich, rohs@inf.ethz.ch

in the coordinate system induced by the code. The latter feature is relevant for applications involving item selection. The algorithm also allows the retrieval of unwarped rectangular areas of the camera image, which is useful for form input.

2. Related Work

Sony's *CyberCode* [4] is closely related to our approach. CyberCode is a visual tagging system based on 2-dimensional barcodes that can be recognized by CCD cameras. CyberCodes encode 24 bits of data, excluding error correction bits. In addition to the ID, the 3-D position of the tagged objects is determined. Proposed applications for CyberCodes are augmented reality systems, various direct manipulation techniques involving physical objects, and indoor guidance systems. In comparison to CyberCode, our code has an address space of 76 bits (83 bits without error detection) and a second guide bar (cf. Fig. 1), which allows the recognition of codes at a greater amount of tilting.

AirClic (www.airclic.com) provides tiny barcode readers that can be attached to mobile phones. The disadvantage of this approach is the necessity of an additional device, which increases the size of the mobile phone and consumes additional energy. Furthermore, barcode readers do not provide the orientation and selection features of our camera-based approach.

In Japan, a number of companies have announced mobile phones with the ability to read *QR Codes* [3]. Announced handsets are the 505 series of NTT DoCoMo, and the Kyocera A5502K. These devices implement the core functionality of decoding QR Codes. They do not, however, explore more complex interaction paradigms in the way our system does. An early application is a system for tracing information about farm products: Producers enter product information including product class, pictures, fertilizers used, and agrichemicals deployed into an online database. The system issues a 12-digit tracking code for each shipping lot, which is attached to the packaging as a QR Code.

Intelcom (www.intelcom.ru) has developed a software development kit for Nokia 7650/3650 for decoding *Data Matrix* [1] codes. An example application generates SMS messages from the phone number and text stored in the code.

3. Interaction Scenarios

New interaction possibilities open up when mobile phones are enhanced to act as sensors for real-world objects. The gap between the physical world surrounding the user and the virtual world is narrowed by offering a natural way of "picking up" data in everyday situations. Information becomes collocated with physical entities and is thus situated and grounded in the real-world context. Mobile phones are in an excellent position to contribute to this vision, because they are ubiquitously available devices, constantly within reach of the user. The short interaction time required for recognizing codes is crucial for usability in a mobile setting. It does not require much effort on behalf of the user and takes just a few seconds.

There are many application scenarios which could take advantage of using visual codes: In mail-order catalogs, items could be selected by scanning visual codes to conveniently place orders. In online banking applications, long reference codes or transaction numbers often need to be manually copied from printed forms. Visual codes and mobile phones could replace expensive dedicated devices which perform this task. In newspapers, visual codes could be used to store or rate newspaper articles, to get background information related to an advertisement, or for updates on information which quickly gets obsolete, like stock quotes and weather forecasts. Using the code coordinate system, it is possible to define printed image maps with predefined sensitive areas that are associated with online content. In this way, a single code can be associated with multiple items: Areas on a geographic map could be linked to weather forecasts corresponding to those regions and each row in a printed table containing stock quotes could be linked to a chart which is displayed on the phone.

In the emerging area of mobile marketing, visual codes could be used to simplify SMS campaigns and SMS lotteries. In a campaign, Coca Cola printed code numbers on the labels of 160 million soft-drink bottles. By sending a number via SMS, customers got logos, ring tones, and greeting cards, and could take part in a lottery. 5.9 million SMS messages were sent, which corresponds to a response ratio of 3.7%. Using visual codes would make the process more convenient for the user and might result in even higher attendance in such campaigns, which would be commercially very interesting.

The proposed 2-dimensional codes are also suitable for display on electronic screens, such as computer monitors, TV screens, public wall displays, and even on the tiny screens of handheld devices. Questionnaires, surveys, and polls on TV (like selecting the best candidate of a TV show) could be realized by overlaying visual codes over the screen image. This idea also has commercial implications. The music company Polydor used SMS as a mobile response channel in a TV spot advertising a CD. Viewers could order the CD via SMS prior to the official release date, which immediately increased sales figures. Again, using visual codes, the response ratio might be increased.

Camera-equipped mobile phones are not only capable of retrieving an object ID, but can also be used for more complex forms of interaction. Our code recognition system also provides the phone's orientation, amount of rotation, and coordinates of the image center.

4. Design of Visual Code and Recognition Algorithm

The mobile phone devices we consider here have severely limited computing resources and often lack a floating point unit. These characteristics make it essential that the code recognition algorithm is lightweight and minimizes the use of floating point operations in order to achieve reasonable recognition rates. The Nokia 7650 mobile phone is a device of this class. It runs the Symbian operating system and is programmable in Java and C++. Evaluation images showed a considerable amount of barrel distortion, low contrast, varying brightness across the image, and blurred edges. The relatively poor image quality determines the minimal size of code elements that can be recognized. We did not use color as a code feature, because of the large differences in color values, depending on varying lighting conditions. Moreover, color codes are harder to reproduce than simple black-and-white codes. These initial findings mark out the design space for the visual code and form the basis for the further discussion. For the envisaged applications, we identified the following technical requirements for the code and its recognition algorithm: recognition of multiple codes in a single snapshot; realization of a code coordinate system that is independent from the point of view; robustness with respect to rotation, tilting, blur, and pixel errors; reasonably short recognition time on the device; ability to unwarp parts of the image for form input (see Sect. 5.). The size of the code was decided to be between 1.5 and 2.5 cm in order to be recognized from a camera distance of between 10 and 50 cm.

The code layout is pictured in Fig. 1. It consists of a larger and a smaller guide bar for determining the location and orientation of the code, three cornerstones for detecting the distortion, and the data area with the actual code bits. On the right side, the coordinate system induced by the code is shown.

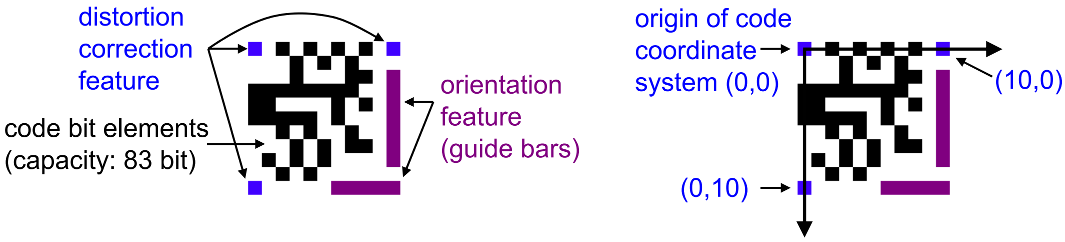

Figure 1. Components of the visual code (left) and code coordinate system (right).

The code can be reliably located even in a low quality image and when it is small and tilted. In the following, the main steps of the code recognition algorithm are outlined.

Correction of radial lens distortion. The image obtained from the camera shows a considerable amount of barrel distortion, which is corrected as described in [6].

Gray scaling and adaptive thresholding. An adaptive method [8] was taken to produce black-and-white images, because the printed code may be unevenly illuminated.

Region identification. This step consists of finding regions of neighboring black pixels, counting them, and assigning a number to each.

Region shapes and orientations. In order to identify candidates for orientation bars among the regions found, the notion of second-order moments is used [7]. From these moments, the major and minor axis of each region is determined. The ratio of the lengths of these axes is a good measure for the "eccentricity" of a region. Furthermore, the orientation vector of the major axis of each region is calculated from the moments.

Locating the codes. Locating codes in the image is done by looking for guide bar candidates and by finding corresponding cornerstones. For each of these candidates, the size and orientation of the region is used to estimate the expected positions of the second guide bar and the three cornerstones. It is then checked whether these components are actually present at the estimated positions.

Mapping between code and image coordinates. To read the data bits of a code, their positions in the image need to be determined. Since the code elements are coplanar, there exists a unique homography (projective transformation matrix) between the code plane and the image plane. The projective mapping [2] can be calculated once four corresponding points are known. This induces a code coordinate system with its origin at the upper left corner of the code and with a resolution of one code bit element (see Fig. 1). The inverse mapping is important for applications which select items in the image. Given a pixel coordinate, its corresponding code coordinate can thus be obtained.

Error detection. The code bits are protected by a (83,76,3) linear code with Hamming distance three, which encodes 76 bits of data to an 83-bit codeword.

The algorithm was implemented in C++ for Symbian OS (v6.1). Replacing floating point operations by shifted integer operations reduced the time consumption of the thresholding phase from 2 seconds to less than 0.5 seconds. The total execution time of the algorithm typically amounts to about 1.5 seconds if less than 5 codes are present in the image, and up to 2.5 seconds if 10 or more codes are present – which is rather uncommon in typical applications.

5. Applications

In this section, we show a few example applications that make use of the algorithm described above and outline some further application scenarios.

For testing the algorithm implementation we developed the *Visual Code Recognizer*, which is shown in the left half of Fig. 2. It is a basic demo application[1] that allows to take images, recognize them, and to send the IDs and image center coordinates to a back-end server. The leftmost screenshot in Fig. 2 shows the display of the phone immediately after an image has been taken. The crosshair in the middle facilitates the precise selection of elements in the picture. The second screenshot shows the display contents after the codes have been recognized. The number of codes detected and their respective IDs are displayed. The currently selected code is framed by a green quadrangle.

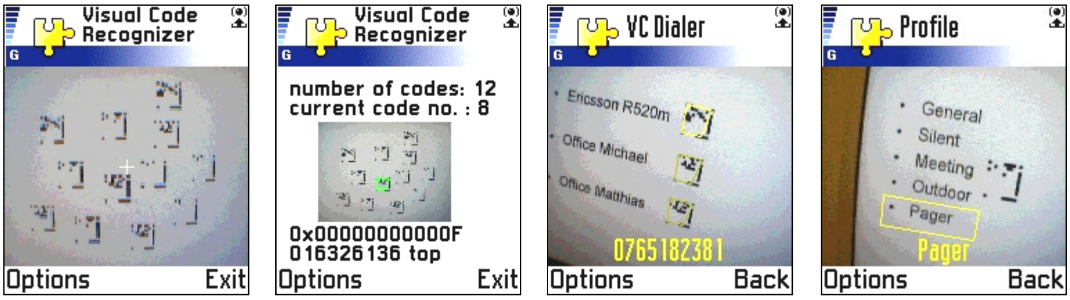

Figure 2. Application screenshots: *Visual Code Recognizer*, *Dialer*, and *Profile*.

An obvious application is to encode phone numbers as visual codes and print them onto business cards, into paper phone books, or onto printed matter. The *Visual Code Dialer* application, shown in the middle of Fig. 2, selects phone numbers by position. If it recognizes multiple phone number codes, the one closest to the image center is selected and displayed on the bottom of the screen.

To simplify interaction with the mobile phone itself, parts of its user interface can be externalized on paper or on a large screen by using visual codes. Deeply nested menu hierarchies – that are a consequence of the tiny display dimensions – are difficult to deal with. Such menu hierarchies could be unfolded, laid out on paper, and recognized by embedding associated visual codes. The *Visual Code Profile* (cf. rightmost screenshot in Fig. 2) application illustrates this scenario. It allows the selection of the current phone profile ("outdoor", "meeting", "pager", etc.) by aiming the crosshair at the desired item. Note that only a single code is necessary in this application. The selected item is identified by using the projective mapping described above. It maps the coordinates of the crosshair at the image center to the code coordinate system, which is independent of the camera orientation.

Since the codes can only encode a limited amount of information, they normally serve as a key that is resolved to the actual data of interest. The ETHOC system [5] realizes the required resolution mechanism. In addition, it generates and issues IDs and administers online resources and services related to physical objects. For use with mobile phones, ETHOC automatically generates WAP pages from the stored content. We extended ETHOC for use with visual codes. After selecting a code, the phone's WAP browser is triggered and the content displayed.

With visual codes, the idea of HTML image maps can easily be transferred to printed documents. Imagine an advertisement for cinema movies in a magazine, in which words and images are linked

[1]This and other applications as well as example codes are available at http://www.inf.ethz.ch/∼rohs/visualcodes.

to movie trailers, the local cinema's booking system, etc. Using the ability of the recognition algorithm to retrieve the targeted position independently from the camera orientation, it is possible to link arbitrarily shaped areas on the page to online content. We have developed an image map generator that allows to interactively define sensitive areas in an image by simply drawing their boundaries. Its output is an XML file that stores the coordinates of the areas relative to the codes contained in the image and the associated URL for each area. Another tool uses these image map files to perform the ID resolution task. It listens on a TCP socket for incoming resolution requests. A mobile phone that has scanned an image map, sends one of the recognized IDs together with the target coordinates. The image mapper identifies the area in which that position is contained and returns its URL.

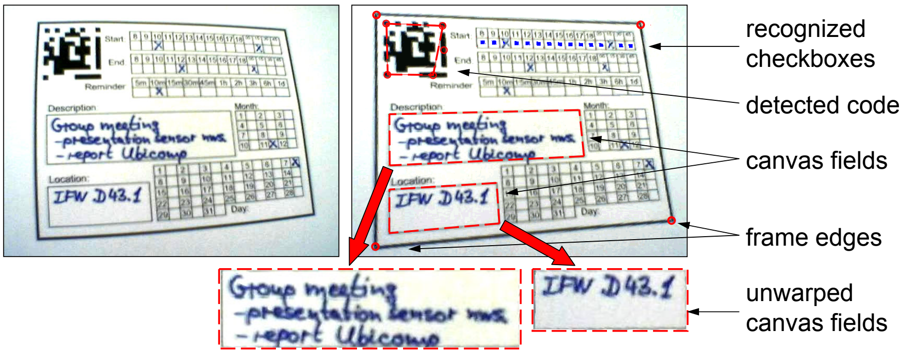

Figure 3. Calendar entry form: code, frame, checkboxes, and canvas fields recognized.

A final application example is the combination of visual codes with paper forms. The fields of such forms can be checkboxes, rulers, or canvases for free-hand input. Using a frame surrounding the form, individual items on the form can be precisely located. For this application, the radial lens distortion correction is essential. In addition, the projective mapping allows to redraw canvas fields without tilting. Fig. 3 shows a calendar entry form as an example. In the original image, the barrel distortion is clearly visible. On the right, the corrected image is shown. The recognition algorithm has detected the code as well as the edges of the frame. The top row of checkboxes depicts where the algorithm looks for checkbox marks. The bottom of Fig. 3 shows the unwarped canvas fields for entering the description and the location of the event. As future work, we envisage a description language for such forms: The code on the form would be taken as a key to obtain a form description, which would then be used to locate marked boxes and canvas fields.

6. Conclusion

In this paper we have demonstrated the feasibility of recognizing 2-dimensional visual codes with resource-constrained mobile phone devices, using low-quality images obtained from integrated CCD cameras. Based on the hardware features of a typical device of this class, we derived the requirements for the design of a suitable code and its recognition algorithm. We highlighted potential application areas and presented a number of prototypical applications.

Employing the integrated cameras of mobile phones as a sensor for real-world objects is a powerful way to enhance the usability and usefulness of mobile phones. The mobile phone offers a natural way of detecting objects in the user's immediate surroundings. It becomes an interaction device, which serves as a "bridge" between the physical world and the virtual world.

References

[1] International Organization for Standardization: *Information Technology – International Symbology Specification – Data Matrix*. ISO/IEC 16022, 2000.

[2] Paul S. Heckbert: *Fundamentals of Texture Mapping and Image Warping*. Master's Thesis, Dept. of Electrical Engineering and Computer Science, University of California, Berkeley, 1989.

[3] International Organization for Standardization: *Information Technology – Automatic Identification and Data Capture Techniques – Bar Code Symbology – QR Code*. ISO/IEC 18004, 2000.

[4] Jun Rekimoto and Yuji Ayatsuka: *CyberCode: Designing Augmented Reality Environments with Visual Tags*. Proceedings of DARE, Designing Augmented Reality Environments, 2000.

[5] Michael Rohs and Jürgen Bohn: *Entry Points into a Smart Campus Environment – Overview of the ETHOC System*. ICDCS 2003 Workshops, May 2003.

[6] Roger Y. Tsai: *A Versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses*. IEEE Journal of Robotics and Automation, Vol. RA-3, No. 4, pp. 323-344, August 1987.

[7] R. C. Veltkamp and M. Hagedoorn: *State of the Art in Shape Matching*. Principles of Visual Information Retrieval, M. S. Lew (Ed.), Advances in Pattern Recognition, Springer, 2001.

[8] Pierre D. Wellner: *Adaptive Thresholding for the DigitalDesk*. Technical Report EPC-93-110, Rank Xerox Research Centre, Cambridge, UK, 1993.