Direct Latitude/Longitude Identification of Paper Maps Using a Camera Phone

Genta Suzuki[1] **Nobuyasu Yamaguchi**[1] **Mineki Takechi**[2]
[1]FUJITSU LABORATORIES LTD. [2]FUJITSU LIMITED
Kanagawa, Japan Tokyo, Japan
{suzuki.genta, nobuyasu, takechi.mineki}@jp.fujitsu.com

ABSTRACT
In this paper we propose a novel scalable technique to embed geographical coordinates into paper maps and directly identify them using a camera phone. Existing paper map recognition systems match a camera image to stored map data. This tends to result in failed matching where the number of stored maps is huge. With this problem in mind, we developed methods to embed latitude/longitude geographic coordinates into paper maps invisibly and recognize the coordinates directly without having to match them with stored data. Yellow shaded patterns are used for embedding because despite being hardly visible, they are still printable by many printers and capturable on camera. The implementation provides real-time recognition via print maps and commercial mobile phones. In addition, from a simulation incorporating over 18,000 map images featuring various cities, scales, and designs, we confirmed our recognition method can identify the correct location without false recognition.

Author Keywords
Paper maps, mobile phone, georeference, steganography, camera recognition, image processing.

ACM Classification Keywords
H5.2. Multimedia information systems: Artificial, augmented, and virtual realities, hypertext navigation and maps

INTRODUCTION
Mobile mapping applications interacting with Geographic Information Systems (GIS) are commonly used nowadays for navigation, searching and viewing environmental representations of locations such as Google Maps for Mobile. In these applications, users can view personalized and dynamic information in a georeferenced manner, whereby the information is overlaid on digital maps.

However, the readability and accessibility of digital maps on mobile displays are lower than paper maps because the displays are compact and low resolution [4]. To deal with this, mobile systems which interact with paper maps have been proposed [3, 4, 6, 7, 9]. These systems recognize the location information of paper maps via physical input devices.

When it comes to improving the systems and popularizing the stage for practical use, the recognition systems of paper maps must meet the following requirements:

– **Using a mobile phone and paper maps:** It is preferable to use mobile phones as mobile devices because they are in such widespread use nowadays. Additionally, it is desirable that paper maps for the systems not only include commercial maps on books, etc., but also user-printed maps.

– **Providing scalable recognition:** The recognition systems would operate for many maps in practical use. The scalability of recognition means the system can provide accurate and prompt identification of map locations, even if there are huge numbers of maps with different scales and designs.

The contribution of this paper is in providing a map recognition system which meets both requirements. We have developed a system which embeds latitude/longitude geographic coordinates into paper maps invisibly and a map recognition system which detects the coordinates by a camera phone. The data embedding method uses a yellow shading pattern which is hardly discernible to the human eye, although printable by most existing printers. Camera phones can specify the geographic coordinates directly by decoding the shading pattern. The system facilitates georeference via mobile interactive applications with paper maps, which inter-translates between screen coordinates and latitude/longitude geographic coordinates.

The rest of the paper is organized as follows. The next section discusses related work, while the third section shows the concepts and algorithms of our system, followed by the implementation of the system and sample application in section four. Based on the results and performances of the implementation, we discuss the advantages and disadvantage of the system in the fifth section. Finally, the last section provides conclusions and future works.

UbiComp'10, September 26–29, 2010, Copenhagen, Denmark.
ACM 978-1-60558-843-8/10/09.

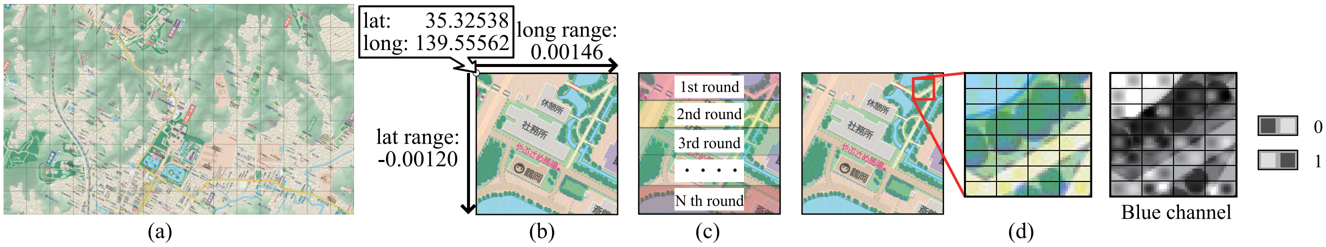

Figure 1. Latitude/longitude embedding method, (a) a map with latitudinal/longitudinal lines, (b) embedding data: latitude/longitude of top-left corner and ranges to other corners, (c) multiple embedding of data bit-strings, (d) embedding method using yellow shading.
Maps: Cartographic data Shobunsha Publications,inc.

RELATED WORK

There have been several approaches involving the recognition of paper maps using mobile devices. However, no systems currently meet both the aforementioned requirements.

The Magic Lens [7] tracks black dots composing grids and searches for the same image of the grids from a stored image patch. In [9], the map location is retrieved from a map database by analyzing the positional relationship of intersections. Although this research meets the hardware requirement, there is no mention of the scalability of recognition, hence it seems hard to guarantee the uniqueness of each pattern for all maps. Additionally, an increase in the stored data would reduce their recognition rate. With speed recognition in mind, if data are stored on a network server, it takes data transfer time.

Reilly et al. [6] use paper maps embedded with RFID (Radio Frequency Identification) tags and a mobile device with attached RFID reader to realize touch based interaction. Norrie et al. [3] adopt paper maps on which infrared-absorbing dots are printed invisibly and special digital pens incorporating a camera. The system retrieves the position of the map by pointing the pen. These works meet scalable recognition requirements because unique IDs can be assigned for each map. However, they use special reader device and special maps.

DIRECT LATITUDE/LONGITUDE IDENTIFICATION

Concept

The basic idea for the requirements involves embedding geographic coordinates invisibly into printed paper maps and recognizing them from the captured image of a camera phone. Our earlier work [2] and its commercial service called *FPcode* (Fine Picture code) [1] have successfully involved embedding 12 digit numerical codes into printed color images using yellow shading patterns. The method exploits the fact that while yellow shading is hardly recognized by the human eye, cameras remain sensitive to the same. Additionally, such shading can be printed by many types of existing color printers.

We aim to apply the method to paper maps and expand to embed latitude/longitude geographic coordinates, which are used in many GIS and GPS devices. This realizes direct latitude/longitude recognition without stored data and georeference of positions on the camera image. Applications can use this georeference to overlay contents such as text information, icons, and photos on the camera preview.

Embedding Latitude/Longitude into Paper Maps

The latitude/longitude embedding system for paper maps performs the following steps:

STEP 1) Dividing maps into grids

First, the given map image is divided into grids by black latitude and longitude lines (see Figure 1 (a)).

STEP 2) Generating data bit-strings

Next, data bit-strings, which include location information, are generated for each grid. Figure 1(b) shows 4 values of the location information: the latitude of the grid's top-left corner, the longitude of the same, the latitude range from top-left to bottom-left corner, and longitude range from the top-left corner to the top-right corner. These values are expressed as signed decimal degrees and to five decimal places (0.00001 is accurate to 1.11 meters at the equator) on a large scale map. After changing these values to binary-coded form, Bose-Chaudhuri-Hocquengham (BCH) error correction codes are encoded from the codes and added to the data bit-strings.

STEP 3) Embedding the data bit-strings into a map grid

The third step embeds the data bit-strings N times repeatedly into a blue channel of the map grid image. Figure 1(c) depicts the repetitive embedding. The blue channel complements that of yellow in the RGB color model. The system divides the grid into blocks and makes pairs of each two adjacent blocks. Subsequently, each data bit is assigned to each block pair. Finally, the gradation level of each block pair is sampled and changed to show the data bit, 0 (if left < right) or 1(if left > right), as shown in Figure 1(d). If the difference of the original gradation level exceeds a predefined threshold, the level will remain unchanged in order to retain image quality.

STEP 4) Printing map

Finally, a map image that is reconstructed by data embedded grids is printed using a color printer.

Recognizing Latitude/Longitude from Camera Images

The decoder works on camera phones and provides direct recognition of embedded latitude/longitude geographic coordinates from a camera preview image. The following show the decoding steps:

STEP 1) Detecting the map grid

In the first step, the decoder detects a grid from a camera image, while the Hough transform is used to detect black lines. If four lines are determined, the screen positions of the four vertices are calculated as grid corners.

STEP 2) Determining bit-strings by comparing blue levels

The second step acquires bit-strings from the grid. The decoder computes the screen positions of all embedded blocks from the grid corners and makes a list of blue levels for the same. Then, for each block pair, 0 (if left < right) or 1(if left > right) is specified by comparing the blue levels.

STEP 3) Determining data bit-strings by a majority

The decoder divides the bit-strings specified in the previous step into N parts and a majority vote of the same for each bit decides the data bit-strings in the third step.

STEP 4) Error detection and correction of the data bit-strings

This step attempts error-correction of the data bit-strings and finally acquires the embedded data. Errors in the data bit-strings are detected and corrected by decoding the BCH codes included in the data bit-strings. If the error correction succeeds, the original data, namely the latitudes and longitudes of the top-left corner, the latitude range from top-left to bottom-left corner, and the longitude range from the top-left corner to the top-right corner are decoded. From the results, the latitude/longitude geographic coordinates for all corners of the grid are decided. Where the error-correcting fails, the decoder reverts to the second step and tries to acquire data from another direction.

Georeference of Screen Positions

After decoding the embedded data, the decoder initializes API for georeference of screen positions from four pairs of the (x, y) and (latitude, longitude) of the grid corners. Homographic transformation is used for inter-translating between screen positions and latitudes/longitudes. In the translation, latitudes and longitudes are converted to screen positions with an accuracy of within 1 pixel. The scope of translation goes from screen positions to geographic coordinates, and the granularities are varied by the ranges of latitudes and longitudes embedded in the grid. The coordinates are calculated to be 1 digit smaller than the ranges. Therefore, if a large scale map is used, the latitude/longitude is determined to an accuracy of within 0.11 meter at the equator from the API.

IMPLEMENTATION

The Paper Map and Camera Phone

We develop the implementation using a Kamakura city map and the T004, which is a camera phone sold in Japan (see Figure 2(a)). The map is printed in A3 size using an EPSON PM-G4500 inkjet printer. The platform of the T004 is Qualcomm BREW [5] and its CPU is Snapdragon QSD8650 1GHz. The size of the camera preview image is VGA (480x640).

Application

We have implemented a latitude/longitude indicator and a contents browser to demonstrate our recognition system. Both applications overlay information on the real-time camera preview.

Latitude/longitude indicator for paper maps

The first application shows the latitude/longitude coordinates of a point on the screen. From every frame during the camera preview, the application fetches the latitude/longitude coordinates corresponding to the middle of the image by georeference. Figure 2(b) presents a snapshot of the application, with the green numbers denoting the latitude/longitude of a location marked by the green cross. In addition to displaying the coordinates, the application facilitates their transfer to other applications. When a user presses an “HP” button, the application is interrupted and an Internet browser shows the location on Google Maps.

Another useful aspect of this application is the ability to set a destination for navigation systems, such as in-car and personal navigation devices.

Contents browser for paper maps

The contents browser overlays photos and text information on areas focused on in the paper map. When the camera preview starts and the decoder recognizes the geographic coordinates of the map, the edge positions of the screen are translated into geographic coordinates. In the VGA size camera preview, (0,0), (480,0), (0,640), and (480,640) positions are translated. Subsequently, all data tagged in the geographic spatial are retrieved from GIS. Finally, the data are superimposed on the screen. Figure 2(c) shows the snapshot. Photos of Kamakura city pop up at the point where the photos are geotagged.

This application can be used to interact with GIS such as by searching shops and viewing sensor data. As for paper maps, it seems applicable for guide maps for tourist sites and amusement parks, exhibition floor plans, and so on.

DISCUSSION

Regarding the requirement for paper maps, we have evaluated that our embedding method can use a number of pairs of printings and papers, including commercial offset printing used for guidebooks and home inkjet printing. As for camera phones, we have confirmed that more than 300

(a) (b) (c)

Figure 2. Snapshots of the implementation, (a) the paper map and the mobile phone, (b) the latitude/longitude indicator shows the latitude/longitude coordinates (green number) of the center cross, (c) photos pop up over the camera image.

camera phones can decode FPcode that uses a uniform method to embed 12-digit numerical data. Based on the results, it is expected that a number of camera phones will be able to capture images, the image quality of which is sufficient to render them recognizable.

Towards the scalability requirement, our approach can distinguish maps where the drawing patterns are similar or the same but the locations differ (e.g. solid color grids of the sea). Concerning failed recognitions of embedded data, we have confirmed that there no false recognition occurred in the simulation. During the latter, attempts were made to decode 18,720 map grids from four directions (top, bottom, left and right). The breakdown of the grids, 104 grids (13x8) per map and a total of 180 maps featuring two cites (Tokyo and Hakone), 9 scale (from 1/500,000 to 1/1,000), and 10 design style are used.

The time spent on decoding is about 72ms in the implementation, while 58ms is required to detect the map grid, and 14ms for the rest. Although there is still room for improvement in the detection time, it seems faster than the time required to retrieve stored data from a network server because the decoder works on a standalone basis.

Our approach of direct latitude/longitude identification has the advantage that applications can operate with the same behavior for a number of maps, which show the same location in different designs by a different publisher. Additionally, the use of the reserved region, which is also included in the data bit-strings, makes the maps stand out. The reserved region is also used to represent the floor numbers of buildings.

One constraint of our system is the fact that pixel sizes of a target grid should retain a certain size in a captured image to ensure recognition succeeds. A minimum of around 200x200 pixels are needed to realize consistent performance. Since grids are about 3cm square, the maximum range of view is 7.2x9.6cm when the size of the camera image is VGA. This means our system is inadequate when attempting to photograph a whole map from separate points as with PhotoMap [8]. On the other hand, our system does lend itself to applications where the interaction model is “a magnifying glass”, namely where a camera phone is used for pointing on a paper map.

CONCLUSION AND FUTURE WORK

In this paper we present a mobile recognition system for paper maps using yellow latitude/longitude embedding patterns. The system provides direct, real-time, and scalable latitude/longitude identification of paper maps.

As future work, we intend to develop the system for other mobile platforms such as Android and iPhone OS. We also aim to adopt it for maps on public display, since they can show our embedding patterns.

REFERENCES

1. FPcode Service. http://jp.fujitsu.com/solutions/fpcode/
2. Moroo, J., and Noda, T. Data embedding method on printed materials. *Proc. STEG’04*, 59-69.
3. Norrie, M., and Signer, B. Overlaying paper maps with digital information services for tourists, *Proc. ENTER 2005*, 23-33.
4. Paelke, V., and Sester, M. Design exploration of augmented paper maps. *Proc. ISPRS Workshop Visualization and Exploration of Geospatial Data*, 2007.
5. Qualcomm BREW. http://brew.qualcomm.com/
6. Reilly, D., et al. Just point and click? Using handhelds to interact with paper maps. *Proc. MobileHCI 2005*, 239-242.
7. Rohs, M., et al. Towards real-time markerless tracking of magic lenses on paper maps. *Adjunct Proc. Pervasive 2007, Late Breaking Results*, 69-72.
8. Schöning, J., et al. PhotoMap: using spontaneously taken images of public maps for pedestrian navigation tasks on mobile devices. *Proc. MobileHCI 2009*, 1-10.
9. Uchiyama, H., and Saito, H. AR GIS on a physical map based on map image retrieval using LLAH tracking. *Proc. MVA 2009*, 382-385.