Detecting and Reading Visual Code Markers in Mobile Phone Camera Images

Christine Paulson

***Abstract*—This paper presents an algorithm for detecting and reading visual code markers in images captured by mobile phone cameras. Localization relies on identifying regions whose geometrical relationships match key features common to all code markers. Code markers may appear at varying sizes, rotations, and shears. A projective transform is used to reconstruct the code maker, such that the positions of data bits can be easily predicted. Reading of individual data bits is performed using thresholding. The performance of the algorithm is evaluated on how well visual code markers are detected, and if data bits are read correctly.**

I. Introduction

VISUAL code markers are two-dimensional bar codes, which can be used in conjunction with mobile phone cameras. Since mobile phones are ubiquitous, and many have cameras and wireless networking capabilities, users could benefit from the use of visual code markers to aid in accessing information. For example, a user who captures an image of a consumer product label with an embedded visual code marker could access relevant product information with a simple click of a button.

The use of visual code markers by mobile phone cameras is discussed by Micheal Rohs and Beat Gfeller in [1] and [2]. In this study, I use the visual code marker format described by Rohs and Gfeller. Others engaging in similar work include Sony and Intelcom [1].

The goal of this effort is to both identify the location of code markers in the image, and to read the data bits encoded in the marker. The method to detect and read visual code markers consists of four stages:

1. Preprocessing and Segmentation
2. Region Analysis
3. Registration
4. Thresholding

II. Algorithm Procedure

A. Preprocessing and Segmentation

Mobile phone cameras produce noisy and low resolution images. The goal of the preprocessing stage is to transform a noisy 640 x 480 RGB image, as shown in Fig. 1, into a 640 x 480 binary image that highlights the guide bars and fixed corner elements common to each code marker.

Fig. 1. 640 x 480 RGB image captured by a mobile phone camera. The image contains three visual code markers. Every visual code marker has common features –two guide bars and three fixed corner elements to aid in detection. Images may be noisy and have poor contrast. Code markers may be rotated and distorted.

Although the code markers are ideally black and white, images captured with a low quality camera and under different illumination are observed to have a large variation in color. Therefore, segmentation based on color is unreliable. Instead, to decouple the color and gray-scale information, the RGB image is converted to a YIQ model. A gray-scale image is formed by extracting only the luminance component.

To correct for low contrast problems associated with poor illumination, a histogram equalization method is applied to the gray-scale image. A cumulative distribution function of the gray-scale pixel intensities is used to identify the intensity range between one percent and ninety-nine percent probabilities. This range is then linearly scaled from zero to one. This histogram equalization method improves contrast in low-light images, but does not apply to local regions with poor contrast.

The next task is to enhance features of the image similar to the guide bars and fixed corner elements of the visual code marker. This is accomplished by applying a 9x9 Laplacian filter. A morphological open-close filtering sequence is applied to smooth the image and reduce noise. The resulting gray-scale image is shown in Fig. 2.

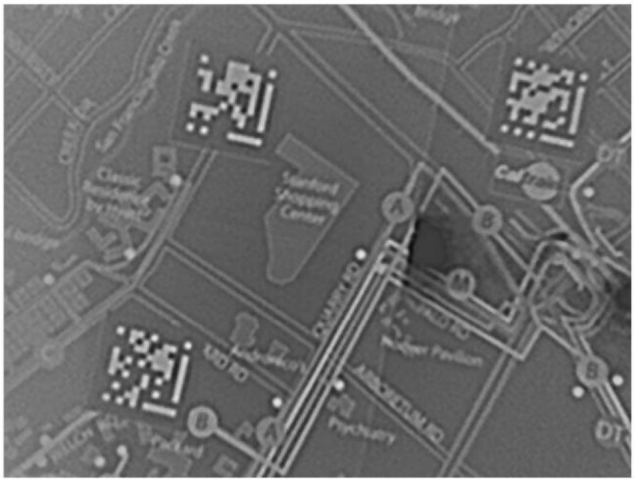

Fig. 2. Histogram equalization, a Laplacian filter and morphological operations are applied to the gray-scale luminescence component of the image. The goal of these operations is to enhance the features of the visual code marker so that they can be extracted from image clutter.

A threshold is applied to convert the gray-scale, filtered image to a binary image. By empirical observation of a series of filtered training images, the threshold was chosen to minimize the presence of unwanted clutter regions, while ensuring that all guide bar and fixed corner features were clearly visible. Morphological operations are performed to remove isolated foreground pixels.

The result of the preprocessing and segmentation stage is a binary image, shown in Fig. 3, whose regions include key identifying features of the visual code markers. The exclusion of remaining clutter regions is performed in the next stage, region analysis. The gray-scale luminescence image is also used during the registration and thresholding stages for reading data bit values.

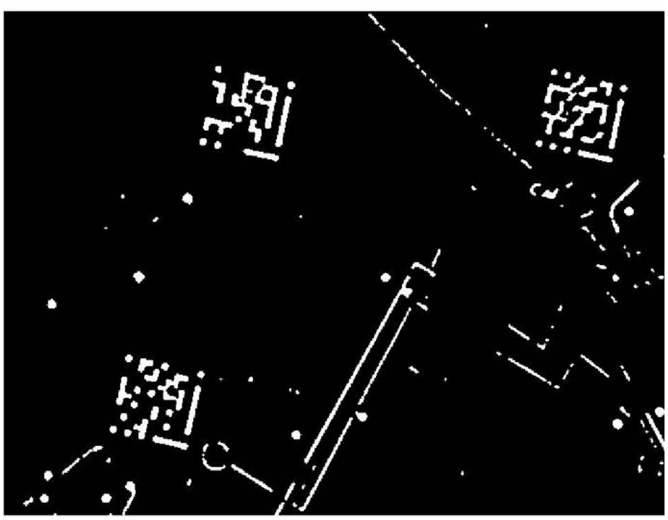

Fig. 3. The binary image result of the preprocessing and segmentation stage shows highlighted features of visual code markers and image clutter. The task of the region analysis stage is to label each isolated region and determine which are the guide bars and fixed corner elements belonging to visual code markers.

B. Region Analysis

In the region analysis stage, a binary image containing both clutter regions and visual code marker regions is analyzed using region labeling and region properties. Regions are identified by locating connected components and assigning a label number to each distinct region. The goal is to identify and localize regions associated with key features of visual code markers. First the locations of short and long guide bars are identified. Then, using properties of the guide bar regions, the fixed corner regions are found.

In general, two types of region analyses are performed. One type evaluates candidate regions based on their intrinsic properties. A second method examines the geometric relationships between two or more regions. Through several rounds of elimination, all clutter objects are removed and code marker features identified.

1) Locating Guide Bars

To improve run time in later more computationally intensive relational analyses, regions whose shape and size differ greatly from a typical guide bar are eliminated immediately from consideration. The minimum required region area is twenty pixels. The eccentricity is a property describing how much a shape deviates from being circular. An eccentricity of zero is a perfect circle and of one is a perfect line. The eccentricity of the rectangular guide bar regions is typically within the range 0.979 and 0.998. Any regions with eccentricity outside of this range are eliminated. The remaining regions are shown in Fig. 4.

Fig. 4. When locating regions corresponding to guide bars, many regions are eliminated because they are too small, or do not meet the typical eccentricity range of a guide bar. After this elimination, the relational geometric properties of two regions are evaluated to identify which pairs represent code marker guide bars.

Next, remaining regions are compared to each other to identify candidate region pairs that may represent a pair of guide bars. For two regions to qualify as a pair, they must be in close proximity, and have a relative angle between 45 and 135 degrees. Ideally, the guide bars will have a relative angle of 90 degrees; however angular distortions introduced by a perspective viewpoint are common.

Guide bars are identified from candidate region pairs if 1) their lengths exhibit the anticipated short-bar to long-bar length ratio of 5/7, 2) a line formed by the major axis of the long bar intersects the short bar (shown in Fig. 5), and 3) a line drawn through the centroid of each region, relative to a line formed by the major axis of the short bar is approximately 69 degrees. An estimate of the observed shearing distortions is made by comparing the relative orientations of the two regions. The length ratio and angular relationship requirements are modified according to the anticipated image skew. All region pairs that satisfy these requirements are considered to be guide bars.

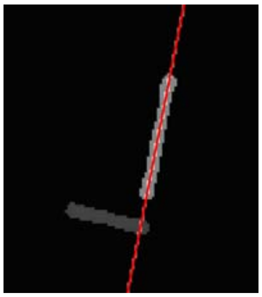

Fig. 5. To qualify as a guide bar pair, the line formed by the major axis of the long bar must intersect the short bar. In addition, the two regions must be approximately normal to each other, and possess the appropriate length ratio.

The location of the lower right (LR) corner of the visual code marker is identified by the intersection of lines formed by the major axes of the two bars. This point, along with the guide bar centroids, is used to estimate the positions of fixed corner elements.

2) Locating Fixed Corner Elements

Identification of guide bars alone is not sufficient to identify the presence of a visual code marker. Occasionally, clutter artifacts are indistinguishable from genuine guide bars. The identification of three fixed corner elements is required to confirm the presence of a visual code marker. In addition, the corner elements play a vital role in characterizing image distortion.

As in the identification of guide bars, regions with the wrong size and shape are removed from consideration. After filtering, the corner elements appear small and round. Regions with area smaller than five pixels and larger than 135 pixels are excluded. Furthermore, corner element regions are required to have an eccentricity less than 0.8.

The upper right (UR) corner element is determined by the region 1) that lies within ten pixels of the line formed by the major axis of the long bar, 2) whose distance from centroid to the LR corner of the marker is approximately twice the distance from the long bar centroid to the LR corner. The lower left (LL) corner is identified in a similar way, with respect to the short bar. The estimated distance of the LL corner to the LR corner is five times the distance from the short bar centroid to the LR corner.

The position of the remaining upper left (UL) corner is estimated using the geometry of a parallelogram, with three corners known. The region whose centroid is closest to the estimated UL corner position is selected.

At this point, a pair of guide bars associated with three fixed corner features is identified as a visual code marker. An illustration of the candidate corner regions, the identified guide bars, and the final corner selections is shown in Fig. 6. The coordinate positions of the four corners of the marker are used in the registration stage to characterize local scaling and distortion.

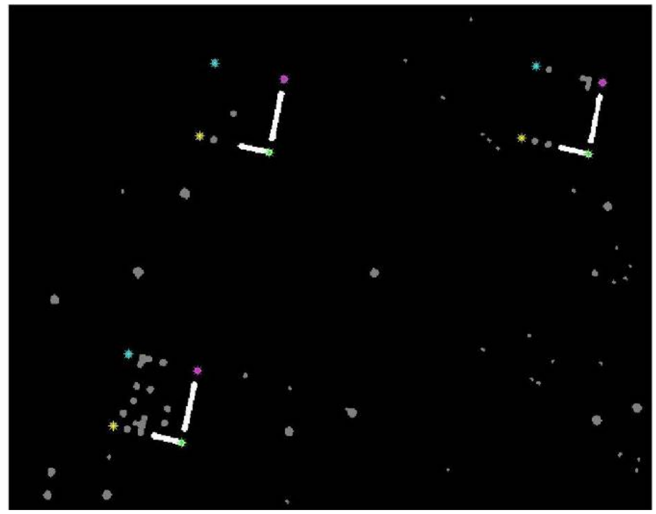

Fig. 6. Shown in gray are the candidate corner regions after large and eccentric regions have been excluded. Guide bars as previously identified are shown in white. A colored star indicates which regions have been selected as fixed corner elements.

C. Registration

Since the image can be taken at a variety of distances and angles, a correction for scaling and distortion must be performed. Since the features of the code marker are coplanar, the observed scaling and shear distortion can be modeled using a projective transform. Straight lines in the code marker are preserved in the distorted image, but their 90 degree angular relationships are altered. With the positions of the four corners of the visual code marker known in the distorted image, the projective transform is applied to reconstruct the square shape of the code marker. This image registration method is applied to both the gray-scale luminescence image (shown in Fig. 7) and the binary image obtained in the pre-processing and segmentation stage.

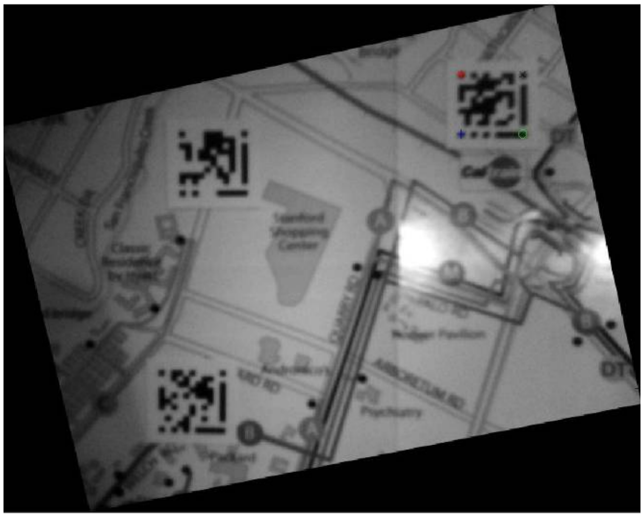

Fig. 7. Image distortion is corrected using a projective transformation. This transformation is applied separately for each visual code marker. Square sub-images selected from the registered image contain a code marker with ideal proportions.

If multiple code markers are present in a single image, the registration method is applied independently for each marker. A square sub-image is obtained for each code marker by isolating the relevant region of the registered image. Some error is introduced into the locations of the four corners of the code marker during the application of the projective transform. To refine the corner position estimates, the region analysis stage is repeated on each sub-image. Finally, the individual data bit locations are estimated by fitting a grid within the four code marker corners.

D. *Thresholding*

Once bit locations are obtained, data bits are read from each gray-scale sub-image. Histogram equalization is applied to the sub-image obtain maximum contrast between 1- and 0-valued bits. Bit values are determined by applying a threshold at the estimated bit centers. Fig. 8 shows a visual code marker sub-image and a reconstructed code marker obtained by thresholding and sampling at estimated bit centers.

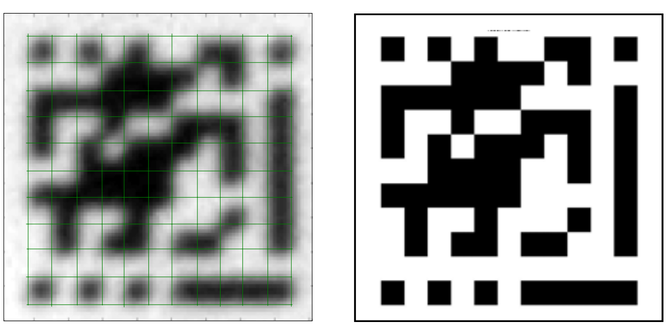

Fig. 8. A registered gray-scale sub-image is shown (left) with a uniform grid superimposed. The reconstructed visual code marker (right) is obtained by sampling the sub-image at estimated bit centers, and applying a binary threshold operation.

III. Results

Twelve training images are used in the design of the algorithm and tuning threshold values. Performance of the algorithm on these images is perfect, as shown in Table 1.

Image	Markers Present	Markers Found	Bit Errors
1	1	1	0
2	2	2	0
3	3	3	0
4	1	1	0
5	3	3	0
6	1	1	0
7	2	2	0
8	1	1	0
9	3	3	0
10	3	3	0
11	1	1	0
12	2	2	0

Table. 1. Algorithm performance on training images is perfect.

Twelve additional images of varying difficulty were used to test the limitations of the algorithm. These results are presented in Table 2. The algorithm performed well in cases where the visual code marker was nearly the same size as those in the training set, and where the shearing distortion encountered by the perspective angle of the camera was low. In more extreme cases, where severe perspective angles were encountered, the algorithm failed to locate some markers.

Image	Markers Present	Markers Found	Bit Errors	Difficulty	Comments on Code Markers in Image
1	2	1	83	medium	1 large and 1 small marker
2	3	1	166	hard	1 normal, 1 curved, 1 small
3	3	1	166	hard	1 normal, 1 curved, 1 small large perspective angle
4	3	3	0	easy	3 normal
5	1	1	0	easy	1 normal
6	2	2	0	medium	1 normal, 1 curved
7	2	2	0	easy	2 normal
8	1	1	0	easy	1 normal
9	2	2	0	easy	2 normal
10	2	2	0	medium	2 normal in very close proximity
11	2	0	166	hard	2 large large perspective angle
12	3	3	0	hard	3 normal + partial markers irregular illumination

Table. 2. Algorithm performance on unseen images varied. Performance was good on images similar to the training set. Performance was poor on some images that presented code markers with significant variance from training images in size and shearing distortion.

IV. Conclusion

In summary, the algorithm developed to identify visual code markers relies on heavily on region labeling and analysis techniques. Generally, the reading of bits is 100% reliable if the feature detection and image registration stages perform well. These tasks are non-trivial when shearing distortion is introduced in the images by the perspective of the camera. In cases where this distortion is not severe, the algorithm performs very well.

References

[1] M. Rohs and B. Gfeller, “Using Camera-Equipped Mobile Phones for Interacting with Real-World Objects,” in *Advances in Pervasive*

Computing, A. Ferscha, H. Hoertner, G. Kotsis, Ed. Vienna, Austria, 2004, pp. 265-271

[2] M. Rohs, "Real-World Interaction with Camera-Phones," *2nd International Symposium on Ubiquitous Computing Systems*, Tokyo, Japan, Nov 2004

[3] R. Gonzalez, R. Woods, and S. Eddins, "Digital Image Processing using MATLAB, " Prentice Hall, Upper Saddle River, New Jersey, 2004

[4] R. Gonzalez and R. Woods, "Digital Image Processing, " 2nd ed., Prentice Hall, Upper Saddle River, New Jersey, 2002