

VISUAL CODE MARKER DETECTION

Nandhini Nandiwada Santhanam, Alex Giladi

EE368, Stanford University, Spring 2006

I. INTRODUCTION

THIS paper proposes a visual code marker detection algorithm. Such systems can be used for parsing the marker codes from images taken by a low-quality cellular phone camera.

II. ALGORITHM

A. Assumptions

We assume that each image contains at least a single marker, and no image contains more than three. We also assume that the size of the marker is limited, and it is about 15%-20% of the image width, and it is non-sparse.

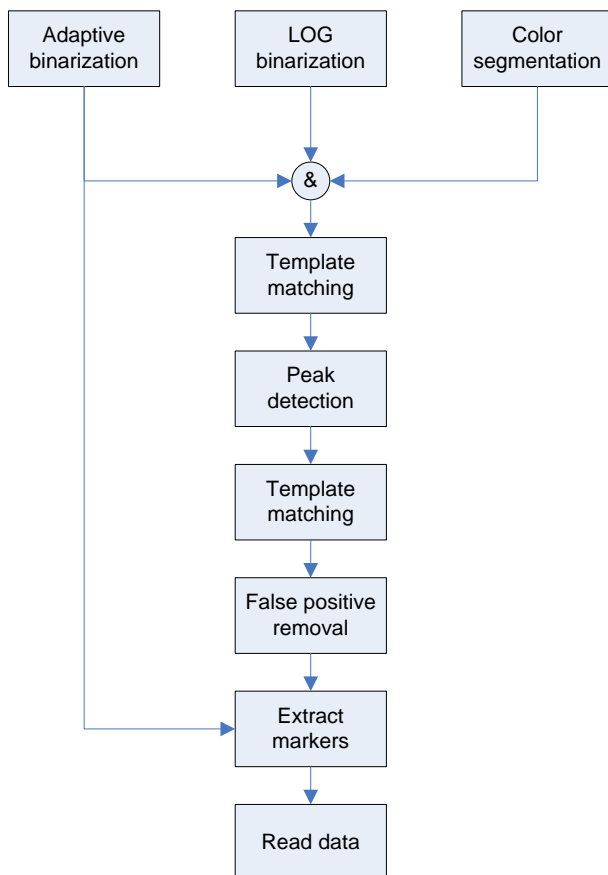


Figure 1: Algorithm stages

B. Algorithm overview

The proposed algorithm has three stages: binarization, marker detection and marker parsing.

An RGB input image is transformed by binarization into a two-level image containing the markers. This stage reduces the amount of unnecessary detail to make it possible for the next stage to detect the marker.

Next stage uses template matching between the binarized image and a binary template to detect the most probable marker location(s).

The approximate orientation of the marker is found, and a convex hull is created about that.

The last stage of the algorithm considers each of these convex hulls, identifies bits representing the marker, and detects its origin.

C. Binarization

Binarization is the most critical part of this algorithm, since it is at this stage that possible marker regions are detected. Clutter removal heavily depends on this stage as well.

Three different binarization techniques are used, and their outputs are AND-ed together.

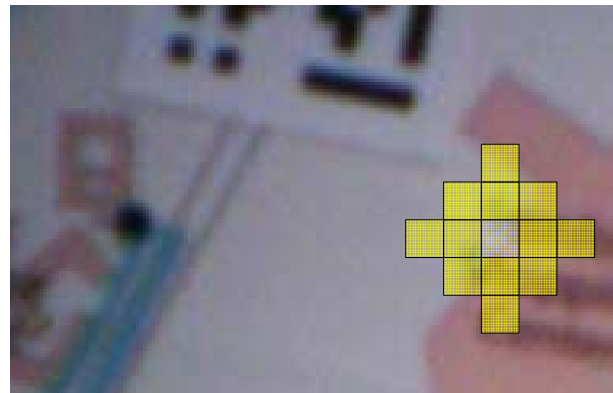


Figure 2: Adaptive binarization

1) Adaptive binarization

This technique further develops the idea stated in [2]. There an image was subdivided into blocks, and the per-block black-or-white decision was relative to the previous n block means. This technique is quite efficient in binarizing nonevenly lit sources.

However, this algorithm does not utilize all spatial correlation, since it only looks backwards in x direction. Still another problem is initializing the means at the image edge, since we do not have the relevant history.

We are not bound by any causality constraints, therefore we propose using a diamond-shaped history structure, and computing the block means in a separate pass. This solution

overcomes the aforementioned problems of [2] and produces seemingly better results.

2) LOG-based binarization

A simple edge detection technique using the $\log(\text{Laplacian of Gaussian})$ filtering followed by median filtering in order to obtain a smoother image is used.

The reason for using such technique lies in the fact that markers are basically black and white patterns, therefore strong edge components associated with them. Edge detection uses this bias to segment the image. The edge detected, and filtered image is then thresholded (a 75% threshold was experimentally found to perform best).

3) Color segmentation

Using grayscale-based methods alone does not utilize the color information – we know that the markers contain only black and white. The color segmentation module performs a pixel-wise operation on every color component. The R, G and B values of a given pixel are considered; and a pixel is assumed to be of a color if the value of the color component is greater than the (scaled) gray level value.

A pixel is considered colored if the grayscale value Y of the pixel is in the range of 30-180, and at least one of the conditions below holds:

$$R > 1.05Y$$

$$G > 1.25Y$$

$$B > 1.45Y$$

4) Large area removal

The resulting black-and-white image is labeled, and only objects within areas in a given range are left in the image, all other objects are removed. Still another model is considering the $D = \sqrt{(\Delta X)^2 + (\Delta Y)^2}$ as the longest linear dimension of an object. Any object with D exceeding a given threshold will be eliminated.

D. Detecting approximate marker location

1) Template matching

A template matching approach is used (in the spatial domain) to identify a marker. A circular template of radius 9.5 is matched with the final binarized image from the previous stage image. After this, the cross-correlation pseudo-image is thresholded to a binary image by considering gray scale values above 15% of the maximum value obtained by cross-correlation. This ensures that the entire marker region is highlighted, and also that the marker does not merge with surrounding regions.

The technique used at this stage does not consider noise, since we don't have a reliable statistical model to detect it. Therefore it is imperative to eliminate noise and irrelevant detail (false positives) from the image before this stage.

2) Removing false positives

Regions with a skewed aspect ratio (beyond 1.4 or 1/1.4) are considered false positives, and so are regions which have an area greater than 10K pixels. It was experimentally determined that these two constraints were most efficient in eliminating false positives.

E. Parsing

1) Selecting region of interest.

The image is now processed on a per-marker basis. The approximate rotation of the marker is determined by taking a bounding box about the marker and calculating its area. This area is calculated for every rotation (the angle of rotation is incremented by 10° in the range of 0° - 90°).

The minimum area corresponds to the approximate angle of rotation. The result of this step is creation of a bounding box corresponding to the actual size of the marker. It can be further rotated by the approximate rotation angle so that only the marker elements are detected for reading the data, and in identifying the origin of the marker.

Bounding boxes are used as masks for extracting the relevant marker areas from the adaptively binarized image (see C.1 above).

2) Reading data bits

The penultimate step of the algorithm, this stage has to take into account overall rotation of the image, as well as any inclusion of perspective.

First the angle of rotation of the marker is determined using the Radon transform (only angles between 0° - 45° and 135° - 180° are considered for obtaining the origin co-ordinates). The image is then rotated by the same angle to correct this.

The shear correction approach used can be described as follows - First obtain the four 'extrema' co-ordinates of the marker, and use a method akin to bilinear interpolation to resample to a 121×121 grid

Data can be now read by considering each 11×11 block as representation of a single bit. The entire block is summed; the data is considered a 0 if the normalized sum is below a threshold (50%) and a 1 otherwise.

The guide bars are now located, and the 11×11 matrix is rotated yet again so that the guide bars are along the bottom right in the marker array.

Additional rotation required to align the guide bars together determine the origin co-ordinates.

Data is read on a bit-by-bit basis once the 11×11 aligned grid is obtained.

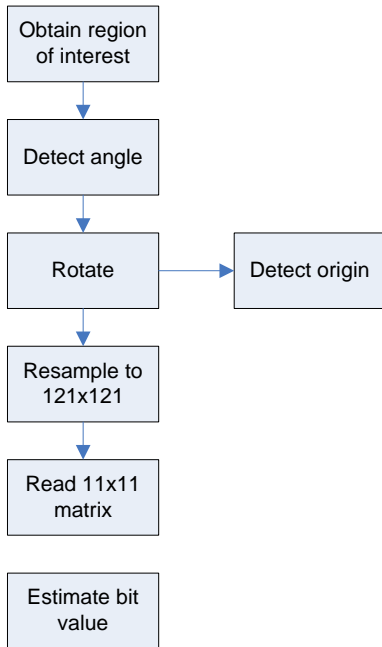


Figure 3: Parsing stages

III. RESULTS

The current score obtained on the standard training running the evaluate function set is 1456.

The table below specifies the detailed scores for the set.

Image	Number of markers	Score	Number of false positives
1	1	83	0
2	2	164	0
3	3	249	0
4	1	83	0
5	3	76	1
6	1	80	0
7	2	166	0
8	1	-41	1
9	3	249	0
10	1	83	0
11	1	78	0
12	2	145	0

In order to illustrate the importance of correct binarization, we ran the algorithm with a simple thresholding at 50% intensity. The score obtained that way was negative.

Below are the results for the training image #9, as an illustration of the way the algorithm works.



Figure 4: Original image

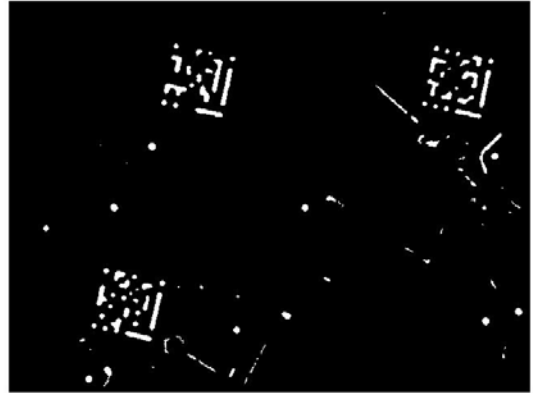


Figure 5: Binarized and cleaned image, output of (C)

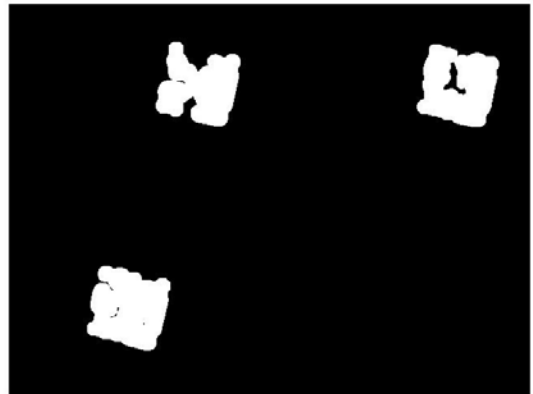


Figure 6: Cross-correlation peaks

IV. FUTURE WORK

Currently, our algorithm only works with limited size non-sparse markers, which are about 15%-20% of the image width.

The algorithm can be changed and extended to support arbitrarily-sized markers, as well as the sparse ones.

V. WORK DIVISION

Nandhini focused on marker detection and parsing (including shear correction); Alex focused on binarization and clutter removal. Everything else was equally divided.

VI. REFERENCES

- [1] M. Rohs, B. Gfeller, *Using Camera-Equipped Mobile Phones for Interacting With Real-World Objects*, In: A Ferscha, H. Hoertner, G. Kotsis (Eds.): *Advances in Pervasive Computing*, Austrian Computer Society (OCG), ISBN 3-85403-176-9, pp. 265-271, Vienna, Austria, April 2004
- [2] P. D. Wellner: *Adaptive Thresholding for the DigitalDesk*. Technical Report EPC-93-110, Rank Xerox Research Centre, Cambridge, UK, 1993.