Visual Code Marker Detection
EE368 - Course Project
Spring 2006

Sami Arayssi

Email: sarayssi@stanford.edu

Abstract— **This paper describes a proposed solution for the visual code marker detection problem which is one example of object detection applications in digital image processing. The proposed approach is based on performing first image segmentation using adaptive local thresholding. Then regions of potential corners are derived using the region counting algorithm and the spatial characteristics of a corner. Then the corners are grouped into triplets to form potential code markers using object based features. Finally, adaptive template matching is performed at these potential objects to detect the actual markers and then read the data.**

I. Introduction

One of the major applications of digital image processing is the detection of objects of interest in an image, often in difficult conditions. In fact, the image could be subject to rotation or tilting due to different view angles from the camera and other geometric parameters. Another challenge in object detection is the quality of the images which is often poor to start with and the type and direction of illumination. The irregularity of such photometric parameters is not only restrained to different images but are also existent within the same image. For example, the type and intensity of illumination varies and is affected by the reflectance properties of the various objects in an image.

Visual code marker detection is one example of the object detection problem in digital image processing. Some of the applications of the use of code markers is the storing of background information on the product it is found on as well as determining geometric position of the image[1] such that the 'reading' of an image is irrelevant to the projection of the scene on the image plane. This year's class project consists of detecting visual code markers in images captured by mobile phone cameras and reading their content. This problem is a perfect illustration of the object detection applications in digital image processing due to the poor quality of mobile phone cameras as well to the uncontrolled illumination pattern and capturing geometric characteristics.

The organization of this report is as follows: Section II presents two of the approaches for image segmentation that were considered at first. Section III on the other hand, describes the approach adopted in detecting the visual code markers. Furthermore, Section IV presents the results derived from applying the algorithm on the training set. Finally, some conclusions are drawn in Section V in addition to potential extensions of this approach.

[1] i.e. rotation, tilting and orientation.

II. Various Approaches for Region of Interest Detection

Regions of interest are parts of an images that potentially contain the visual code marker. Detecting such regions is a focal point in the success of the algorithm adopted to solve the problem. Several approaches have been tested for this part and some of them are briefly discussed next followed by the actual approach adopted in the proposed solution described in Section III. Two of the approaches[2] that were considered at first were color based segmentation and corner detection concentration.

A. Color based Segmentation

The images captured by the cell-phone camera are stored in the form of RGB images. The object markers present a color distribution pattern that is different than the other parts of the image in the case of "colored background" due to the fact that they are black and white objects. However, this approach is not very useful in the case of black and white background which result in great overlapping between the two regions.

The color based segmentation approach consists of bounding the color space region of the marker by a geometrical shape and then performs image thresholding with the pixels classified as foreground pixels if they fall within this region. The markers are extracted from the images and the chromaticity distribution is collected to bound the region of interest irrespective of the illumination characteristics of the image. The chromaticity distribution of the markers in the "Cb/Cr" plane collected fom the training images is presented in Figure 1.

The region can be clearly bounded by an ellipse with some points left unbounded. This at first is a tempting approach, but one should not forget that the background might also be black and white which makes this method loose its basic idea it was founded on.

B. Corner Detection Concentration

The markers are characterized by the presence of multiple small squares within a bigger square. This leads to the idea that visual code markers presents a relatively large concentration of corners which might be used to detect the regions of interest. So applying any corner detection algorithm[3] to the image after noise reduction and sharpening, we get a mask of corners which is then processed to find the smallest square that can

[2] These were not adopted in the final solution.

[3] In this case the Harris corner detection algorithm was used.

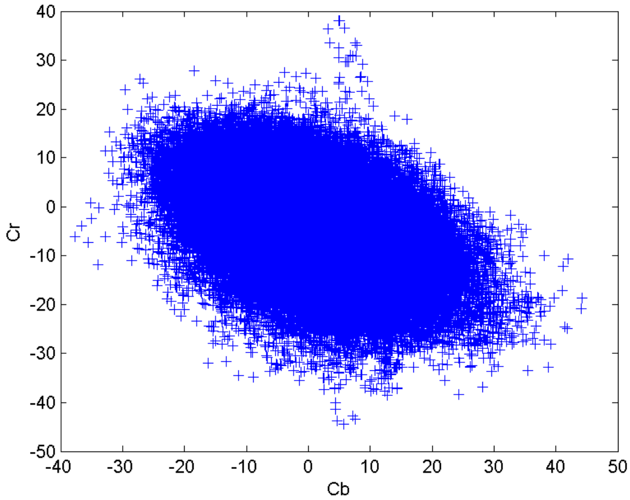

Fig. 1. Marker region in the Cb/Cr chromaticity space.

bound the largest concentration of corners and hence detect the regions of interest. An successful illustration of this approach is shown in Figure 2 and Figure 3.

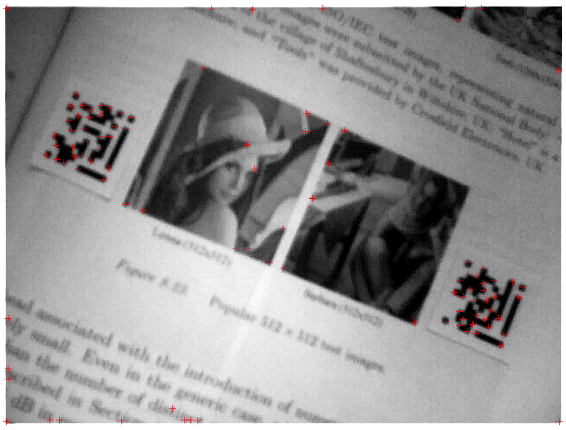

Fig. 2. Black and white image with corner detection applied after noise reduction and sharpening.

However, there is a defect in this approach when it comes to images that have objects in the background that also present a large concentration of corners and hence can be mistaken for markers. One can do some pre-processing before applying this method to get rid of these regions but the problem is that there is no common features for the background objects.

III. Adopted Approach for Visual Code Marker Detection

The idea in the adopted approach was to make it indifferent to the background color and directly related to specific geometrical features of a marker. The steps of the algorithm are presented next.

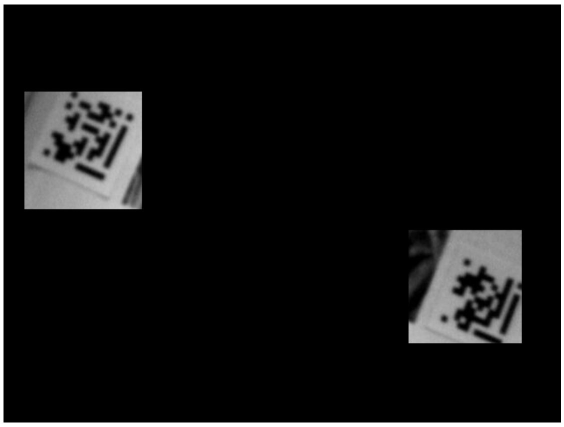

Fig. 3. Regions of interest detection by bounding the largest concentration of corners by the smallest square window.

A. Image Segmentation using Adaptive Local Thresholding

First, the illumination pattern of the image is regulated using the "scale-by-max" color balancing algorithm to make thresholding irrelevant to the difference in illumination between the various images. Adaptive local thresholding is performed on the luminosity component of image to separate the black regions of the markers that contain the required information used in further processing. In other words, thresholding is performed at each pixel with a different threshold value that is dependent on the grey-level distribution of the neighboring pixels. More precisely:

- A 40x40 window is drawn around pixel (x, y) and the mean M of the grey-level distribution of the surrounding pixels is found.
- Pixel (x, y) is set to belong to the darker region if

  $$f(x, y) < M - 10 \tag{1}$$

  The 10 values was chosen empirically.
- This is repeated for every pixel.

A result of this step is shown in Figure 4.

B. Mask Derivation

The processing of the resulting binary image starts by performing $open - close$ morphological operations to fill white and black gaps. Then two masks are derived by region filling using the region counting algorithm for two different thresholds:

1) A mask ($mask1$)[4] of the potential fixed corner elements of the marker by keeping the regions that are less than 180 pixels in pixel surface area.
2) A mask ($mask2$) including also the fixed guide bars to be used in upcoming processing stages. The threshold level in this case is a pixel surface area of 1500 pixels.

[4]The mask name used in the report is different than the variable names in the code.

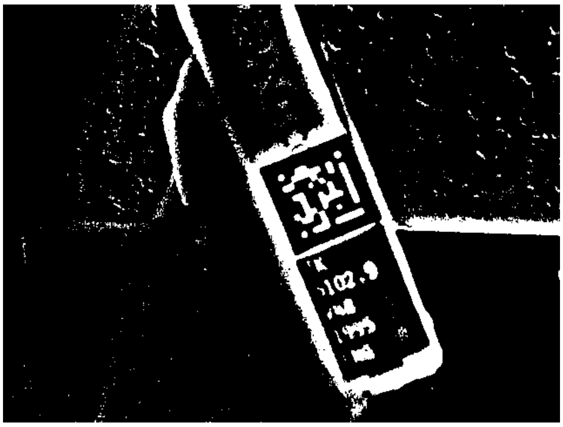

Fig. 4. Illumination regulation and adaptive local thresholding of $training\_1.jpg$.

Since $mask1$ is used to detect the candidate markers by holding the potential marker corners, any large concentration of white region is filtered out from $mask1$ by further processing since these are sure not to constitute a corner. The $mask1$ of the thresholded image of Figure 4 is shown in Figure 5.

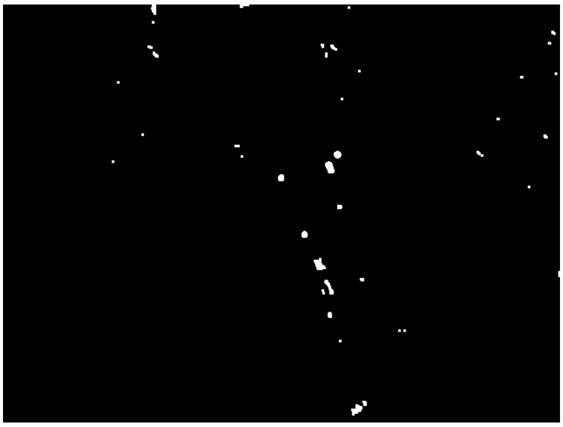

Fig. 5. The $mask1$ image of $training\_1.jpg$.

C. Detection of Potential Markers by Fixed Corner Grouping

The markers are detected by finding triplets of region centers in the $mask1$ image that are of the size a corner which satisfy the characteristics of the fixed corner elements of a marker as follows:

1) Filter out regions that do not fit in a square with a large concentration. These are sure not to be potential corners.
2) Find the centers of the resulting potential corners.
3) Group the resulting potential markers into triplets that represent the three corners of a potential marker. The criteria for triplet detection are:
   - Distances between the corners are > 40 and < 150. These refer to the limits of the dimension of the marker viewed from different distances from the camera.
   - They form two vectors that are perpendicular in the third corner. The orthogonality criterion is to have cos(angle) < 0.2.
   - The two orthogonal sides should have almost the same length. This is modelled by a ratio of sides > 0.75 or < 1.25.
   - The corner regions belonging to the same triplet have close pixel surface area. Consequently, the ration of the regions pairwise in the triplet is constrained between 0.5 and 2.

Figure 6 shows the regions that belong to the set of triplet corners.

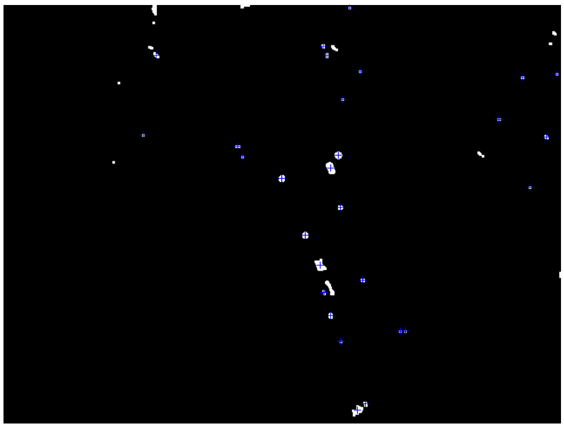

Fig. 6. The regions that belong to the set of corner triplets of $training\_1.jpg$. These are marked by a blue $+$.

D. Detection of Markers by Fixed Guide Bar Adaptive Template Matching

A marker is found among the potential triplets if the fixed guide bars are matched at the fourth corner based on the orientation of the potential marker and its dimensions. Each potential triplet (i.e. marker) is processed as followed:

1) Find fourth corner of the square formed by the triplet. Check if the fourth pixel is white (i.e. belongs to short bar). The triplet is filtered out if this is not satisfied because it is certain then that it does not belong to a marker.
2) The rotation angle of the triplet is found and the fixed guide bars template is rotated accordingly, shifted to the position of the fourth corner and matched after adapting/adjusting its size to the dimensions of the potential marker.
3) A threshold of 0.7 is used to detect the fixed bars and if the result of the template matching is close to that, the

template is moved a couple of pixels and then matching is done again to be sure it is a marker or to detect it if it was not found in the previous iteration.

E. Reading Data

Once the markers are found, the next task is to read the data. The challenge here is that the marker might be tilted and rotated. In other words, the marker plane is different than the image plane. As a result, a linear transformation is found between both planes given that we have the coordinate of the four corners in the image plane. In fact, let (x_i, y_i), $i = 1 \; to \; 4$, denote the coordinates of the corners[5] in the image plane. These are arranged in matrix B where

$$B = \begin{pmatrix} x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \end{pmatrix}$$

Let A be the matrix of the coordinate in the marker plane where

$$A = \begin{pmatrix} 0 & 10 & 10 & 0 \\ 0 & 0 & 10 & 10 \end{pmatrix}$$

The linear transformation matrix X between the two spaces is then found by solving the linear system

$$A = X * B \tag{2}$$

Finally, the data is read from the marker by mapping the points in the marker plane to the corresponding points pixels in the image plane. The result of this algorithm for the $training\_1.jpg$ image is shown in Figure 7.

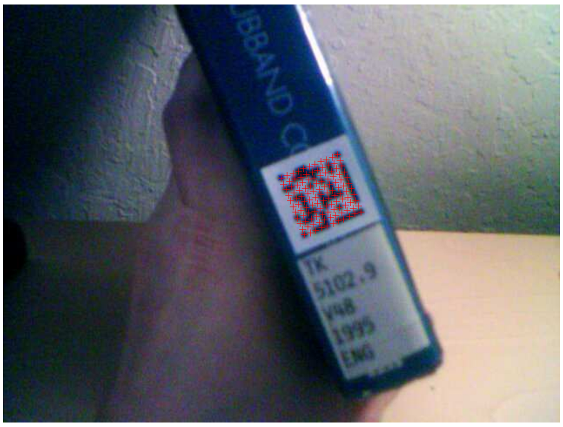

Fig. 7. Image showing the marker and the position of the read data $training\_1.jpg$. These are marked by a red $+$.

[5]The indexes refer to the corners in clockwise direction starting from the origin of the maker.

IV. Marker Recognition

V. Results

Table I presents a summary of the results when the algorithm is applied to the 12 images of the training set given by the course staff. The detection ration of the markers is 95.6% with no repeats or false positives. The average time of the program is 21.4 s.

TABLE I
SIMULATION PARAMETERS

Training	Markers	Detected	Repeats	False Positives	Time
1	1	1	0	0	8.26 s
2	2	2	0	0	16.18 s
3	3	3	0	0	21.31 s
4	1	1	0	0	31.17 s
5	3	2	0	0	29.63 s
6	1	1	0	0	23.38 s
7	2	2	0	0	18.33 s
8	1	1	0	0	9.65 s
9	3	3	0	0	21.75 s
10	3	3	0	0	12.06 s
11	1	1	0	0	41.28 s
12	2	2	0	0	23.71

VI. Conclusion

There are numerous algorithms that can be applied to solve visual code marker detection problem. The image segmentation as a first step is the basic step upon which the algorithm is founded. Consequently, a good job should be done at this step and in fact this was illustrated in the relatively good adaptive thresholding that was performed in this algorithm. I had several algorithms for this project which where abandoned in the search of a more robust and simpler algorithm. The image segmentation algorithm were presented in Section II. The reference used for this project were mainly [1], [2].

Finally, since I was not part of a group, time did not permit me to improve more the algorithm by making it faster and with a 100% success ratio or test other algorithms. Some of the things i would have improved are the way tilting of the images was addressed.

References

[1] B. Girod, “EE368: Digital Image Processing.” Lecture Notes, Spring 2006.

[2] D. Vernon, “Computer Vision,” in *6th European Conference on Computer Vision*, (Dublin, Ireland), July 2000.