# Visual Code Marker Detection

**Kapil Rai**
Student, EE368,

Stanford University

## Abstract
*In this project report a scheme for Visual Code Marker detection in Cell phone Camera based images is outlined.*

## Keywords
Visual Code Marker, Eigen Image, Gaussian pyramid, Linear discriminant analysis

## INTRODUCTION
This report consists of five sections. Section II outlines the issues associated with the VCM detection and standard approaches. Section III discussed the theory of operation & describes the algorithmic steps in detail. Section IV discusses the implementation details. In Section VI results are presented with conclusions.

## Section II
Visual Code Markers have a distinctive pattern for classification. With the three corners and two guide bars, these markers are distinctively polarized and are suitable for detection. Since the VCM are always superimposed on another object (background), a classification method should carefully eliminate any collinear and repetitive patterns in the background. Often the background consists of textual and pictorial combination or simply a map with different illumination. Also, the relative size of the VCM (zoom), its orientation and the perspective, place considerable demand on the image pre-processing and registration methods that must precede the training or recognition steps.

## Section III : Theory of operation
The training images are in 480x640 (JPEG) format. The preprocessing steps are shown in the figure 1. Firstly, the gray scale image is generated. Since the VCM are distinctive gray level images, the color segmentation is not of direct value. However, any background information (noise, clutter) that has prominent color components can be minimized by eliminating the color information. The RGB images were converted to the HSV space and the saturation axis was used to identify the edges. On this axis, the VCM markers stand out. However, it must be clarified that all that is gotten from this step is only a measure towards the Region of Interest classification. Let the regions identified and the associated properties be called the feature set F1.

Histogram Equalization is performed on the gray image to restrict any excessive illumination and to balance out the luminance (or lack of it). This equalized gray image is now used for image pyramid creation.

This residual grayscale image is arranged in a Gaussian and Laplacian pyramid to facilitate multiresolution processing.

Repetitive pattern elimination.


The Recognition process entails two key steps – Another feature set is generated based on the edge detection method. The Laplacian of the Gaussian operator is used to detect the edges in the gray scale image. Regions are then connected using morphological operations. A square structural element was used for the close and open operations. Region labeling is then used to identify the connected regions. Based on the following properties, these are further classified –

1. Area : Regions exceeding a certain threshold are retained.
2. Perimeter : Regions exceeding a certain perimeter are kept
3. Perimeter Correlation : After matched filtering with the perimeter pattern, a correlation threshold is to be exceeded for a region to be classified.

4. Euler number : This is a measure of subobjects in the region and given that a large Euler number points to several small sized objects inside, it can be used along with other features to eliminate large areas.

Using the region properties, like the centroid and the bounded box around the centroid for every classified region of interest, firstly the relevant pixels in the original grayscale image are masked out. This results in a Feature set F2. A set union, F, of the two sets F1 and F2 is deduced for running the detection process.

Using the local elliptic fit on every region in the set F, the local orientation of the region with respect to the world axis is obtained. A method to adaptively compute the affine transformation was also attempted. However, this was abandoned in wake of time.

After correcting for the rotation, by the orientation of the region, a normalized correlation is performed with each image in the Eigen Image set as well as with Eigen NonImage set, at every eigen image resolution. Based on a detection threshold the image is classified as belonging to the Object set or Non-Object set.

The total number of objects identified with their coordinates and the Bounded Boxes are then passed to the next level in the pyramid. Similar search is conducted for the corresponding regions and maximum correlation computed with various resolution of the eigen images.

This procedure is repeated for all the levels in the pyramid. If at the end of this procedure, no regions passed the detection test, then it is concluded that there are no objects in the image. Likewise, concordant detections at every pyramid level are required for classification of the object.

**VCM Decoding :**

Following the detection step, firstly the right resolution (pyramid level) is determined. This allows for handling various zoom levels in the original image. The criterion for the correct resolution level is –

[I,J]= Arg(Max(Correlation(I,j))), where i: pyramid levels, j: eigen image resolution level

Using the combination of the I,J the guide bars are firstly localized using correlation on the boundary strip. Specifying some points on these bars, the coefficients for affine transformation are deduced. Then the corner points are localized.

Last task that remains is helped by reducing the size to the native 11x11 format and extracting out the 83 bits in the data fields of the VCM.

**Implementation details**

In this section the various steps used for implementing the above algorithm are described. Firstly the method used for creating the data base is outlined. This is done in two different ways –synthetic and based on the real data. The synthetic data base provides the initial training sequence and is eventually to be updated by real images and the discards, appropriately.

**Data Base Creation-I : Synthetic**

Given that a VCM is a 11x11 matrix with 83-element code, a large sample of 10000 such matrices was created using a random number generator and the VCM generation script ‘generate_code.m’. These binary images were used to compute the autocovariance matrix from which the Eigen images were computed. Since the code generated images are only MN=121 elements long thereby allowing for a straightforward eigen vector computation, and the principal objective of the data base is to detect the VCM accurately and not to classify this further, direct computations were used. The Sirkovich-Kirby method was used only to classify the objects into VCM or non-VCM categories.

The steps used were :

1. Generate 11x11 VCM images viz. V1, V2,.., VL where L>>MN. Let y be the MNxL matrix defined as –
2. $Y = [V_i - \mu]$, where $V_i$ is a MNx1 column vector representing one VCM, i=0,1,…,L-1
3. The autocovariance matrix for the ensemble of the training vectors is computed as $R = YY^H$
4. The eigen vectors are computed as in the following equation – $[\Gamma, D] = eig(R)$, where $\Gamma$ is a matrix comprising Eigen vectors corresponding to the λ, eigen values contained in the diagonal matrix D.
5. These eigen values are arranged in descending order and the corresponding eigen vectors are arranged in the same order.
6. Now any pattern from the set Vi is used and the Eigen images are generated as per the following pseudo-code -

   For i=1:I

   $Y_R(i) = \mu + a_i \Gamma^T$ where $\mu$ is the mean image and the weight matrix is computed as $a_i = (Y_i^T \ \Gamma_i)^T$

   Next i
7. These $Y_R(i)$ are the eigen images and along with the mean image serve the recognition process.
8. To facilitate detection using the multiresolution processing, these eigen images are also interpolated to other sizes. There are as many resized eigen images as the number of levels in the image pyramid. This is the final synthetic eigen image set.
9. Some repetitive patterns with standard shapes like rectangles, & text(repetitive rectangles at low resolution) were also used to create a separate database based on the above principle. This database is called Eigen NonImage set.

### Data Base Creation – II : From Real images

From the training images provided, the visual code markers can be extracted and a database created using similar operations as outlined in the above section on synthetic eigen images. Actually, these images are used to update the synthetic data base. Both the mean and the eigen images are recomputed when a new real image based VCM is introduced to the data base. This, heuristically, adds the real-world dimension to the otherwise purely synthetic data base, which theoretically is enough to identify VCM in a camera image.

The rejected images are also projected on the Eigen NonImage set and the data base is suitably updated. This alternative Eigen NonImage set makes the recognition process much more robust by removing these objects from the Region of Interest.

### Coordinate transformation - I

The other key operations are about converting from the local region coordinates to the global image coordinates at the lowest pyramid level. At higher levels, the result from the lower levels is to be used as a starting point.

This transformation was derived and is outlined below.

The correlation peak found as [I,J] in the above section, subtends an angle on the region boundary. This is given as $\theta = \tan^{-1}(J/I)$, $r = \|I, J\|$, the radius vector

$\alpha = \theta_1 - \theta$, where $\theta_1$ is the orientation of the region, determined by the axes of the fitted ellipse around the region.

(Xw,Yw) are the coordinates of the upper left corner of the bounded box, capping the region.

Then, the corrected coordinates of the upper left corner of the region are given as –

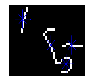
$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} X_w \cos(\alpha) \\ Y_w - \sin(\alpha) \end{bmatrix} \begin{bmatrix} I \\ J \end{bmatrix}$$

### Coordinate transformation - II

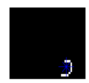
Once the lowest level generates the rotation corrected point, the large error due to the rotation deformity is reduced. Once the fine block search is concluded at a higher pyramid level, and the guide bars or the corner points are located then the affine (inverse) transformation is used to correct for scaling, shear and rotational deformities. Matlab function cp2tform is used with the corner points identified in the target and the Base image. This TFORM structure gives the inverse transformation. There is a bug in Matlab that was circumvented by explicitly creating another TFORM structure with the array describing the inverse transformation.

### Results

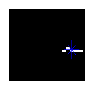
Firstly, the region classification method at the lowest pyramid level with the details outlined above worked very efficiently. Based on the tuning parameters, this process could be further automated. The region estimates for several images are shown below.



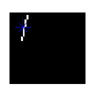
3.2212



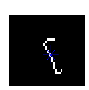
1.1915



2.0267



3.1646



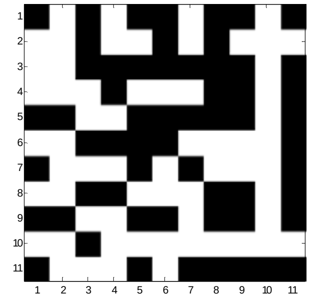

These regions were then treated as individual test images. At the lowest level again, the correlation metric used to identify the nearest neighbour search was always significantly higher for the areas of interest. This should help minimize the false alarm rate.

The figures below show the regions with their centroids and the correlation metrics. A balance can be obtained between the false alarm rate and the missed detection rate by opeing up the region properties mentioned in the section above.

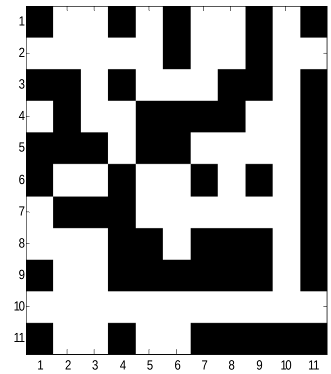
The work required to complete the detection and localization of the visual code markers could not be completed in time. Therefore, from the attached code, these functions in the code are missing.

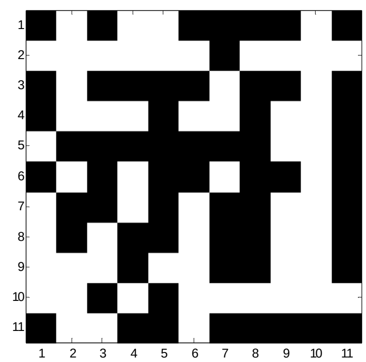
### References

1. Course Notes, EE368, Spring 2006, Prof. B. Girod



2.





3. EIGEN Images created from the synthetic data base.