Visual Code Marker Detection

Tina T. Dong

I. Introduction

The distinctive patterns of visual code markers are made up of black and white squares. This property makes it easy to find the visual code markers by using morphological operations to locate regions of possible visual code markers. The selected regions are then oriented so that their lines are horizontal or vertical and then tightly bounded about the black corners so that the matrix of 11 by 11 elements can be determined. Once the elements are known, the fixed elements, such as the corner elements, guide bars and white space, can be used to determine the exact orientation of the matrix after which the bits of the visual code marker can be read.

II. Algorithm

A. Determining Possible Locations for Visual Code Markers

Visual code markers are distinctive patterns in that the entire region is made up of black or white squares. This property makes it easy to find the visual code markers by using morphological operations to locate regions of possible visual code markers. The dense area of black and white squares will have a high response to both image erosion and image dilation. First, the entire image is dilated with a 20 by 20 window and also eroded with a 20 by 20 window separately. If a particular region produces a response above a certain threshold, the region is very likely to contain a visual code marker.

B. Removing False Positives

Some other areas, such as text, may also produce a high response to both erosion and dilation. These regions are removed by shape and size. The visual code marker is a square, so the region should not be overly skinning or overly fat. Also, the region should not be too big or too small. Regions with fewer than 1500 pixels or over 9000 pixels are removed.

C. Orienting the Visual Code Marker

Since the visual code marker is made up of a bunch of squares, the visual code marker has a lot of straight edges. In fact, it should have lines of two slopes, the horizontal lines and the vertical lines. The Hough Transform is used to determine the orientation of the visual code marker by selecting the largest theta corresponding to the largest peak as the orientation angle. The region is then rotated by the orientation angle. Since the horizontal lines and vertical lines are perpendicular, the visual code marker is hopefully now a square. The visual code marker may be rotated by a multiple of 90 degrees, but the correct orientation will be determined after the bits are determined.

D. Fitting the Visual Code Marker

Once the region is rotated so that it is squarer, the region is fitted to surround the black corners. This fit is required for determining the bits. The center of the selected region is assumed to be in the visual code marker, and the bounds of the visual code marker are moved away from the initial center until no additional black pixels are added. (Black is defined to be below a certain threshold.) This algorithm assumes the edge of the visual code marker is white, and the borders will end in white edge region of the visual code marker. The algorithm is iterative and stops until increasing the bounds to the left, right, top and bottom no longer add any new black pixels.

E. Determining the Visual Code Marker Matrix

Once the tight fit has been found, the matrix can now be found. The selected square region is split up into an 11 by 11 grid, and the pixel value at the centroid of each cell is used to determine the blackness or whiteness of cell of the matrix. A threshold is applied to the matrix so that each element will be labeled as black or white.

F. Determining the Visual Code Marker Matrix

Once the tight fit has been found, the matrix can now be found. The selected square region is split up into an 11 by 11 grid, and the pixel value at the centroid of each cell is used to determine the blackness or whiteness of cell of the matrix. A threshold is applied to the matrix so that each element will be labeled as black or white.

G. Determining the Exact Orientation of the Visual Code Marker

The fixed corner elements and the fixed guide bars are used to determine the exact orientation of the matrix. The orientation with the largest number of matches with the fixed elements and white space is then the exact orientation of the matrix.

H. Recovering the Bits of the Visual Code Marker

Now that the orientation of the matrix is known, the bits can be read from the matrix.

III. Example

Figure 1: Input Image

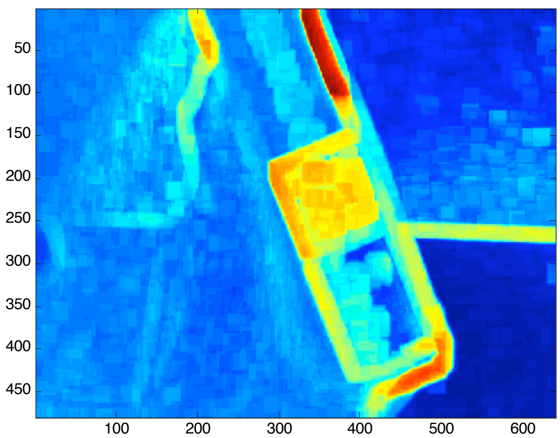

Figure 2: Response of Input Image to Morphological Processing

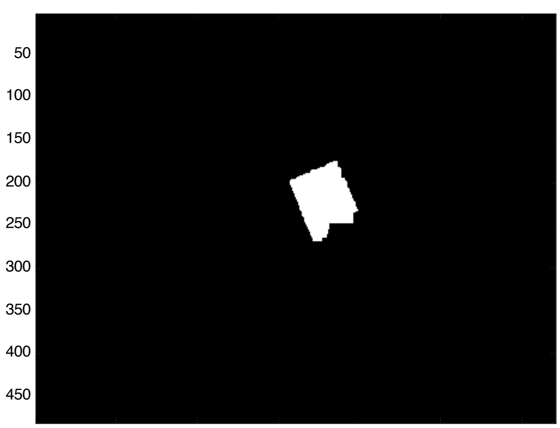

Figure 3: Selection of Regions of Interest

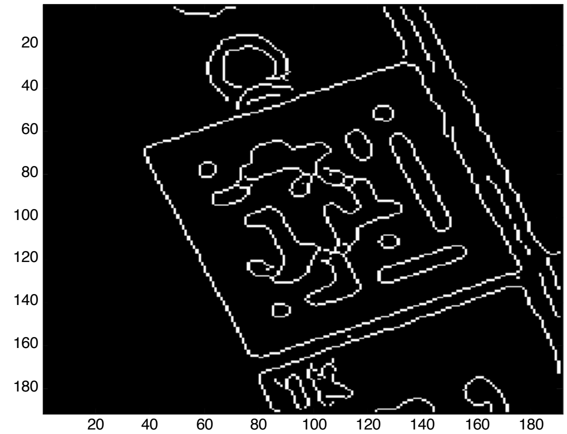

Figure 4: Edge Detection for Orientation Analysis

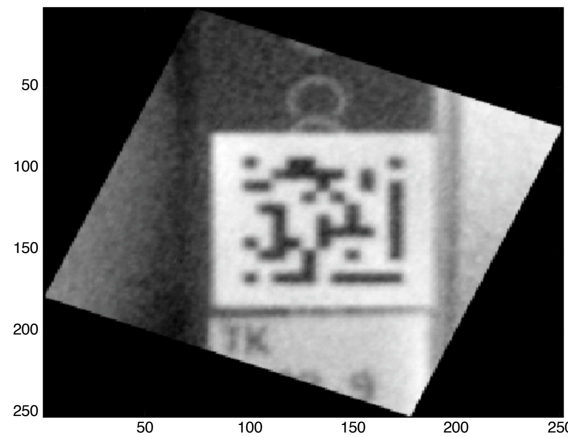

Figure 5: Rotated Selected Region

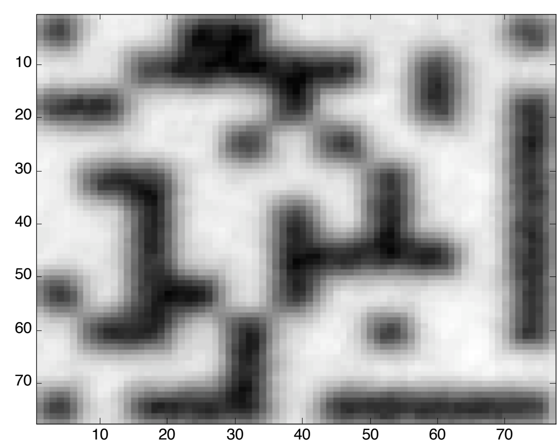

Figure 6: Cropped Visual Code Marker

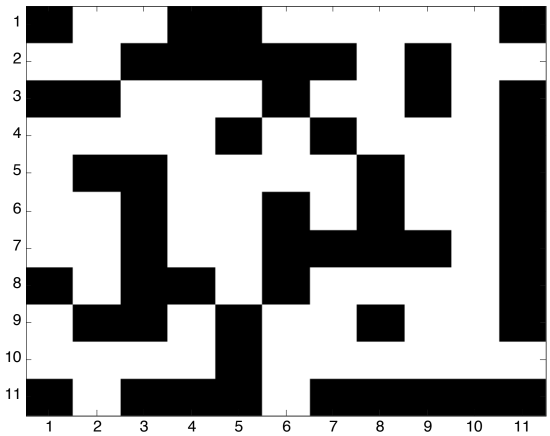

Figure 7: Visual Code Marker Estimate