IMAGE BASED BARCODE READER

THURAYA S. FAWZI

SEPTEMBER 2012

Title of the Thesis : **IMAGE BASED BARCODE READER**
Submitted by: **Thuraya S. FAWZI**

Approval of the Graduate School of Natural and Applied Sciences, Çankaya University

Prof. Dr. Taner ALTUNOK
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Sciences.

Assist. Prof. Dr. Murat SARAN
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assist. Prof. Dr. Reza HASSANPOUR
Supervisor

Examination Date: October 01, 2012
Examining Committee Members

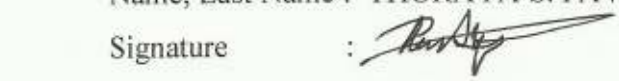
Assist. Prof. Dr. Reza HASSANPOUR Çankaya Univ.

Assist. Prof. Dr. Kasım ÖZTOPRAK Karatay Univ.

Assist. Prof. Dr.Yuriy ALYEKSYEYENKOV Çankaya Univ.

STATEMENT OF NON- PLAGIARISM

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last Name : THURAYA S. FAWZI
Signature : 
Date : October 01, 2012

ABSTRACT

The purpose of this study is to generate a system capable of barcode recognition from digital images and decoding the information stored on them. This project is only capable of decoding code 39 barcode type and deals with images with high resolution. There are three main steps in this study: preprocessing the image, detecting the barcode and decode the information stored on them. Preprocessing include several filtering steps to generate a smoother version of the image. Hough transform is mainly used in locating the barcode depending on theta and rho values and after finding the right threshold value the barcode will be detected. Decoding step is implemented by separating bars from spaces and converting each pixel's binary value into it's corresponding ASCII code. Then the information stored on barcode will be printed.

Keywords: Barcode, Hough transform, Threshold, Rho, Theta

ÖZET

Bu çalışmanın amacı, dijital görüntülerde bulunan barkodu tanıma ve içinde depolanan bilgileri çözme yeteneğine sahip bir sistem oluşturmaktır. Bu proje yalnızca kod 39 barkod tipi çözme yeteneğine sahip ve yüksek çözünürlüklü görüntülerle ilgilidir. Ön işleme görüntü daha yumuşak bir sürümünü oluşturmak için çeşitli filtreleme adımları içerir. Hough transformasyon esas olarak teta ve rho değerlere bağlı olarak barkod'u tanımlamakta kullanılır sonar doğru eşik değerini bulduktan sonra barkod'un yeri tespit edilecektir. Kod çözme adımı, barları boşluklardan ayırmak ve her pikselin ikili değerine karşılık gelen ASCII koduna dönüştürmek ile uygulanır. Sonra barkod üzerinde saklanan bilgiler yazdırılacaktır.

Anahtar Kelimeler : Barkod, Hough Transformasyon, Eşik, Rho, Teta

ACKNOWLEDGMENTS

I would like to express my sincere gratitude to my advisor Assist. Prof. Dr. Reza HASSANPOUR for his guidance, suggestions, encouragement, understanding and supports throughout this research.

Also I would like to express thankfulness to my parents, my husband and my two lovely children for their support and patience through my study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF GRAPHS

LIST OF FIGURES

INTRODUCTION

Barcode could defined as a set of black lines and white spaces printed on labels with different thickness representing a set of numbers. Later they developed into geometric shapes including rectangles, hexagons, dots and other geometrical patterns forming 2 dimensional barcodes. It is a very popular way to uniquely identify objects. Barcodes are used to quickly and efficiently look up an item in a database .

They are useful in identifying an item or a product with a series of lines, spacing, numbers, images, or data arranged in a way that can be easily read by a mobile computer or barcode scanner that has been specially designed to read different types of barcodes. The barcode can only work within a systems environment. The components required in a barcode system include: a barcode label, a scanner, a decoder, a computer with a database and a printer. A scanner reads a label using a given symbology, a decoder then converts this signal into digital form so that a computer can perform its functions.

While barcode technology appears on practically every item purchased or cataloged on a store's shelf, the data that they individually contain varies. Different types of codes can be specially created or tailored for specific industries or businesses. Anyone interested in buying one for business use is advised to study what their current industry standard is and become educated on the different types of barcodes that exist, as well as how much data they can contain before deciding upon which style to use.

CHAPTER I

TYPES OF BARCODE

Barcodes generally have three types: 1- dimensional barcodes, 2-dimensional barcodes and 3-dimensional barcodes.

1.1 1-DIMENSIONAL BARCODES

There are over 300 different types of 1 dimensional barcodes and the most popular types are :

A. CODE 39 which is an alphanumeric barcode that can be used to encode 26 uppercase letters, 10 digits and 7 special characters and can be as long as necessary to store the encoded data. The barcode symbol includes a quiet zone, start character'*', encoded data character,'*'stop character, and a trailing quiet zone. Each data character is made up of 5 bars and 4 spaces for a total of 9 elements. This symbology is widely used in non-retail applications.

Figure 1. Code 39 Barcode Sample

B. INTERLEAVED 2 OF 5 BARCODE is a higher density numeric symbology based upon the Standard 2 of 5 symbology. The symbol can be as long as necessary to store encoded data and can encode data in both bars and spaces while Standard 2 of 5 only encodes information in the bars. The structure of this barcode is made up of a start quiet zone, start pattern, data, stop pattern and trail quiet zone. Its name" interleaved " came from the fact that a digit is encoded in the bars and the next digit is encoded in the spaces. Which makes the digits interleaved together. It is mainly used in the distribution and warehouse industry.

Figure 2. Interleaved 2 Of 5 Barcode Sample

C. CODE 128 BARCODE is a very high density alphanumeric barcode. It is designed to decode all 128 ASCII characters. Each barcode consists of a quiet zone, start character, data character, check sum, stop character and quiet zone. There are three subtypes of code 128 (A, B,C) to represent all 128 ASCII values. Which can be mixed within a single barcode.

Code 128A represents ASCII characters 00 to 99 (0-9, A-Z and control codes), special characters and FNC1-4. Code 128B represents ASCII characters 32 to 127 (0-9, A-Z, a-z), special characters and FNC1-4. Code 128C represents 00-99 (encodes each two digits with one code) and FNC1.

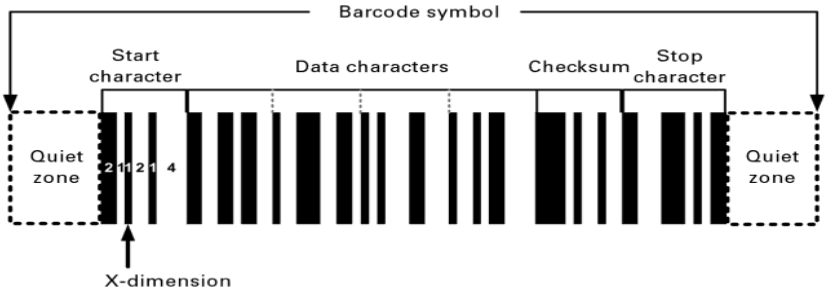


Figure 3. Code 128 Barcode Structure

D. UPC CODE (UNIVERSAL PRODUCT CODE) Its main two types are UPC-A and UPC-E. The UPC-A barcode is 12 digits long, including its checksum. UPC-E barcodes are special shortened versions of UPC-A barcodes and are 6 digits long. Both UPC-A and UPC-E bar codes may have optional 2- or 5-digit supplemental codes appended to them.

Figure 4. UPC Code Barcode Sample

E. EAN (EUREPEAN ARTICLE NUMBERING) has two different versions EAN-13 and EAN-8. EAN symbols are fixed in length and encodes only numbers. Scanners equipped to read EAN symbols can read UPC symbols too, but UPC scanners will not necessarily read EAN symbols.

A special EAN-13 barcode with a 5-digit supplemental code is used on books to encode the International Standard Book Number (ISBN) and the price. EAN-8 is a shortened version of EAN-13 code which is introduced to barcode some small products which are not big enough to carry the full EAN-13 barcode.

Figure 5. EAN Barcode Sample

F. CODABAR barcode type was produced to use it in labeling retail price systems. Then it's use has been expanded including non-retail stores applications such as medical industry, libraries, and shipping. It's character set contains start and stop characters which should be one of the four alphabetic characters (a, b ,c and d). Each character is represented by seven elements including four bars and three spaces but it does not have a fixed number of wide spaces per character, nor does it have a set pattern for the start and stop characters. These properties make the codabar barcode weak and less secure than other barcode symbologies.

Figure 6. Codabar Code Barcode Sample

1.2 2-DIMENSIONAL BARCODES

It is a graphical image that stores information of any product in two ways, horizontally and vertically. It is similar to 1 dimensional barcodes but can represent more data per unit area. Two-dimensional barcodes were improved for applications where only a small amount of space was available for an automatic ID symbol. The first application for such symbols was unit-dose packages in the healthcare industry. These packages were small and had little room to place a barcode. The electronics industry also showed an early interest in very high density barcodes, and two-dimensional barcode since free space on electronics assemblies was infrequent.

Tow dimensional barcodes can be seen on websites, product packages, street signs, magazines and so many uses. Most mobile phones with cameras are built with barcode readers so that can provide services in selling digital tickets, coupons and airline boarding. There are a variety of 2D barcodes with distinctive shapes. Some of them especially those are used in mobile devices are shaped as square arrays. The most popular types are:

A. AZTEC BARCODE that **i**s made of squares growing from the center around a center mark. The size of the symbol is characterized by the number of "layers" outside the centre mark, and this can range from 1 – 32.

Figure 7. Aztec Barcode Sample

B. SMALL AZTEC CODE is another version of Aztec Code used for encoding messages that include maximum 95 characters. It considered as a space savor by removing one set of rings from the finder pattern, eliminating the reference grid, and using a shorter mode message which limits the symbols to four data layers. It's encoding rules are generally the same as for standard Aztec Code. Both Small Aztec symbols and standard Aztec code decoder are compatible with each other so that the two types can be intermixed in applications.

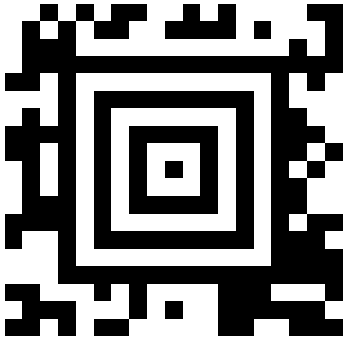


Figure 8. Small Aztec Code Sample

C. DATA MATRIX BARCODE is a two dimensional matrix symbology which is made up of square modules arranged within a finder pattern. DataMatrix symbols may be square or rectangular.

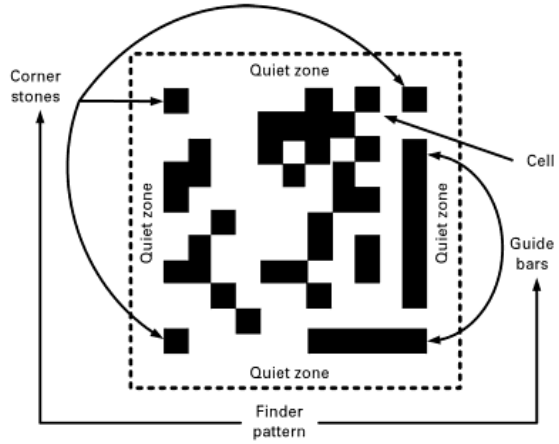


Figure 9. Data Matrix Barcode Structure

D. CODABLOCK F is a stacked barcode symbology based on Code 128. It can encode the full ASCII character set in a symbol which consists of multiple rows of Code 128

type symbols, using a common “Start A” start character and a common “Stop” stop character. Apart from the start and stop character the other characters in adjacent rows have a horizontal line between them.

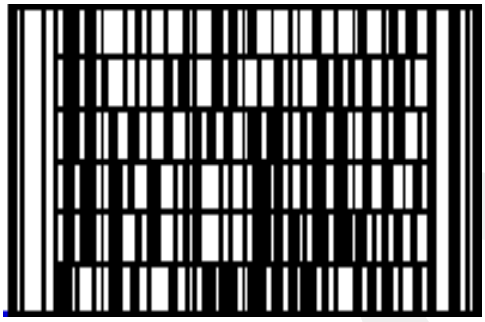


Figure 10. Codablock F Barcode Sample

E. QR CODE BARCODES is a matrix code also called quick response code with square shaped symbols. This code can easily identified by their finder pattern of nested dark and light squares at three corners of the symbol.

Figure 11. QR Code Barcode Sample

1.3 3-DIMENSIONAL BARCODES

3 dimensional barcodes like in 2 dimensional, codes can contain different types of information like pricing, height, weight and other product information. The need to this system began to avoid the problems occurred by high temperature, chemicals and solvents that would destroy any barcode as in linear or 2 dimensional barcodes.

This code is embossed on a surface and can be read by using differences in height of each line rather than contrast to distinguish between bars and spaces. The 3 dimensional barcodes make it almost unacceptable to change or prohibit the barcode's information

and results in fewer stocking faults and in turn minimizes operating costs of a manufacturing process.

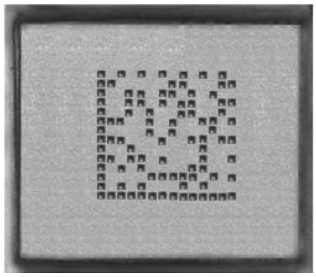


Figure 12. 3 Dimensional Barcode

1.4 BARCODE READERS

There are many types of barcode readers which decodes and analyzes the barcode's image data supplied by it's sensor and sends the barcode's content to the reader's output port.

1.4.1 Pen Wands

It is the simplest barcode reader that is known for it's durability and low cost. It consists of a pen or wand including a light source and photodiode that are placed next to each other. To read a barcode it has to remain in direct contact with it therefore it should be held at a certain angle and moved over the barcode at a certain speed.

1.4.2 Laser Scanners

Laser scanners use a system of mirrors and lenses to allow the scanner to read the barcode regardless of orientation. They work the same way as pen wands but they use a laser beam as the light source.

They can be either stable or hand held and does not have to be close to the barcode to read it's information. Specialized long- range laser scanners are capable of reading a barcode up to 30 feet away.

1.4.3 Slot Scanners

Are usually used to scan barcodes on identification cards and membership passes. They can read barcode's using a visible red light and invisible infrared. To read a barcode on an identification card it should be printed around half an inch in from the edge of the

card , while the scanner remains stable the card with the barcode on it should be pulled by hand through the slot.

1.4.4 CCD Scanners (Charge- Couple Device)CCD scanners are similar to camera scanners in work. The scanner work by loading an array of light sensors to the head of these scanners which then analysis the intensity of light from the barcode labels.

CCD scanner has better read-range than the pen wand and due to it's low cost and speed it is assumed suitable to be used in retail sales and usually used to handle credit cards, gift cards in major applications by reading the magnetic strips located at the back of these cards.

A disadvantage of a CCD scanner is that it can not read a barcode that is wider than it's input face decreasing the possibility of using this type of scanners in warehouse or industrial applications.

1.4.5 Image Scanners

An image scanner uses a small video camera to capture an image of the barcode and then uses digital image processing techniques to decode the barcode. There are different techniques used to read barcodes in image scanners that will be discussed in detail in the following chapters. It can read a barcode from about 3 to 9 inches away and generally costs less than laser readers. Depending on whether the image is clean or not and the distance between the camera and the image containing the barcode.

CHAPTER II

RELATED WORK

Many image processing techniques and algorithms have been developed to read barcodes from images. Some of these techniques will be discussed in this chapter briefly. To read a barcode image we will face tow main problems. The first one is locating the barcode in the image so that it could be read in spite of the blurriness and noise and the second problem requires decoding the barcode after being detected.

There are different methods and techniques used in image barcode reading, some of them are:

1. Discreet Cosine Transform (DCT) Domain
2. Hough Transform Method
3. Difference of Gradients Technique
4. Bottom Hat Filter Method
5. Neural Networks Algorithm
6. Mathematical Morphology Operations
7. EM Algorithm
8. Fourier Transform

Some of these methods were used alone for the detection process while other researchers and companies combined two or more of these methods to perform detection. Next the use of every method or technique from whom will be denoted with a brief explanation of some of the methods.

a. Alexander Tropf and Douglas Chai represented Discreet Cosine Transform (DCT) Domain (1) as explained in [1].
b. Ruwan Janapriya, Lasantha Kularatne, Kosala Pannipitiya, Anuruddha Gamakumara, Chathura de Silva and Nalin Wickramarachchi used a combination of Hough Transform Method (2), Difference of Gradients Technique (3) , and Mathematical Morphology Operations (6) in their study as explained in [2].
c. A study presented by Tewson Seeoun used Hough Transform Method (2).
d. Shu-Jen Liu, Liang-Hua Chen ,Hsiao-Rong Tyan , Jun-Wei Hsieh had used Neural Networks Algorithm (5) in barcode detection in their research as in [2].
e. James Juett and Dr. Xiaojun Qi presented a study and a project using Bottom Hat Filter Method (4) and Mathematical Morphology Operations (6).
f. A paper named "Barcode Recovery Via The EM Algorithm" by Turin and Boie, R.A. uses EM Algorithm (7) in barcode detection.
g. In our study and project we have used Hough transform Method (2) and a series of new filtering approaches for detecting barcodes.

2.1 BARCODE DETECTION METHODS

2.1.1 Discreet Cosine Transform (DCT) Domain

The basic concept of this method is that computing the DCT of each 8 x 8 block of image's pixels. This is done by encoding the image's frequency information and then averaging all of the DCT blocks so that this single block will be a delegate of the entire image. This averaged block contains all the predominant frequencies and their corresponding directions.

After computing the directions of these predominant frequencies they are usually assumed to represent the barcode. The barcode now may be detected by finding the suitable directionality of an extended region.

The main problem of this method is that when the image contains more than one object or the barcode occupies just a little part of it, Then the information reflected from the average DCT block will not represent only the barcode we need to detect. This fault causes using the DCT domain method insufficient and faulty.

2.1.2 Hough Transform Method

As in other approaches the problem of damaged barcode images, reflections and blurriness exits. So the Hough Transform method was used to reduce these problems by reading barcode images faster and decoding them.

The Hough Transform method usually used to detect lines, ellipses, and straight patterns using segmentation. It is used in barcode detection after applying edge detection methods. Due to studying only prescription images, Muniz L. and A. Otero [3] assumed that all barcodes lie on ninety degrees and that using line equation all non zero edge pixels are mapped to Hough plane.

After examining the Hough plane and capturing ninety degree lines, the region of vertical lines is detected as barcode area. Real time applications requires many multiplications, and because of the slow of this method it is not enough for these applications therefore it is considered as a case study.

Later Wavelet transform based barcode localization technique was applied by obtaining horizontal, vertical and diagonal features. The barcode area is detected by applying binary erosion, dilation and thresholding [4]. Then Wavelet transform barcode localization technique was improved by combining it's features with the logical OR operation to procure a binary image. Finally, barcode area orientation will be find by applying free angle thresholding in a binary image[5].

2.1.3 Difference Of Gradients Technique

This technique was developed at UC Santa Cruz in the computer lab to locate barcodes in images. The main concept of this technique is that locating a barcode by searching for regions with high horizontal gradients and low vertical gradients, depending on the assumption that any barcode consists of vertical bars. Then this mathematical operation will be applied on the image with barcode followed by applying low pass filter to the result to reduce isolated edges and noise, for an image I(x, y) :

$$\left|\frac{\partial}{\partial x} I(x,y)\right| - \left|\frac{\partial}{\partial y} I(x,y)\right| \qquad (1)$$

When the result is examined the barcode will be located at the point where this operation is maximized and the left and right bounds of the entire barcode will be demonstrated. The advantage of this technique is it's high accuracy, efficiency and swiftness. Using

this equation will subtract out edges in the incorrect orientations reducing a great part of edges that do not include barcode.

A disadvantage of this technique is it's assumption of barcodes being horizontal, this assumption sometimes will cause a little lost in a vertically aligned barcode lines.

2.1.4 Bottom Hat Filter Method

This method locates the barcode in two stages. The first stage in this method is to detect anything that could institute a barcode and posteriorly abrogate non barcode regions by eliminating the noise from background. The second stage is concerned with using the information obtained from the first stage to obtain the exact location of the barcode. These stages begins with converting the original colored image into grayscale image then, applying a simple contrast stretching to highlight differences between dark and light areas. This highlighting is done by applying a close morphology operation to the image then subtracting the original image. The highlighted areas which are produced using bottom hat filtering truly pursue the shape of the bars in a barcode and produces a filtered image which can be reused for each orientation tested.

The next step is converting the grayscale image into a binary image. This is done depending on Otsu's method [6] and using graythresh function in MATLAB. Noise reduction must be performed to eliminate any object that have been highlighted into a black and white pattern but they do not have a barcode's characteristic. This step prevents noise from interfering with the suitable localization of barcodes.

Barcodes have a unique characteristic which is strong directional continuity at one orientation while all others have very low continuity. This algorithm has benefit from this characteristic because noise does not have this continuity and will be reduced. These orientations are obtained by applying image opening algorithm.

Objects with the most dense in the produced directional opening images will be assumed as a barcode. Then a graythresh function will be reused to convert back these density images into binary images. Regions in this binary density may represent a barcode. After identifying a definite barcode region, it's exact height and orientation may be calculated by locating one of it's bars.

The endpoints of a barcode are calculated by checking each bar in the image starting at the centroid of density region and translating back into regular image coordinates. The exact angle and height of a barcode can be unequivocally calculated after finding two endpoints of it, and at this point the localization stage will be performed.

2.2 BARCODE DECODING

When locating a barcode is accomplished applying one of the barcode detection methods, it should be decoded to extract all product's information. The main factor in barcode decoding is the resolution of barcodes which are assumed to have high quality.

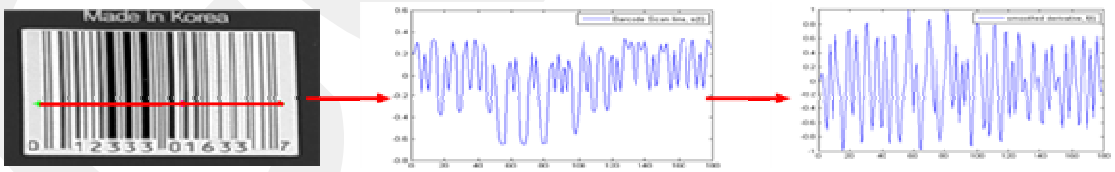
There are many decoding methods that can be used to decode barcodes after being located.

2.2.1 Barcode Decoding Using Hidden Markov Models Applications

This method suggests to apply Hidden Markov Model's (HMM) statistical machine learning framework to deal with the problem of barcode decoding. To begin decoding a barcode, the left and right boundaries obtained by locating the barcode must be used to obtain a scan-line of the barcode. Then, computing the strain points of the scan line. These strain points either matches the endpoints of each bar or matches noise only. In order to distinguish between which strain points represent noise and which represent endpoints of barcodes, the HMM has been used. [7]

Each state in the HMM is a pair of strain points. One can make several observations for each pair of points. So, the derivative of the obtained scanline at each point must be mesuared beside the distance between adjacent points.

The characteristic of using the HMM method is that determinig which sequence of states best explain the observations [8], and that allowing to differentiate between noise and actual bar endpoints as shown below:



(a) (b) (c)

Figure 13. (a) The barcode (b) A scan line from the barcode, s(t)
(c) The derivative of s(t), called f(t)

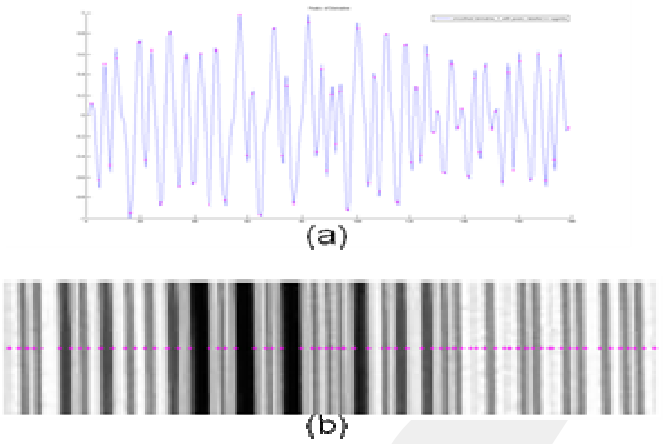


Figure 14. (a) The derivative of the scan line f(t), with peaks identified. (b) The edges of the bars, which corresponds to the peaks of (a)

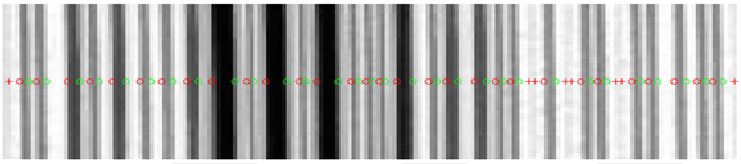


Figure 15. Final result. The left side of each bar is marked with a red circle, the right side with a green circle, and noises with red crosses.

2.2.2 A Minimum Path Approach To Barcode Decoding

This approach is a unique barcode decoding approach that is based on Dijkstra's algorithm for the shortest path. It uses the same techniques as described in barcode decoding using Hidden Markov Model's application, to find potential bar edges which also called nodes.

Each possible transition between nodes and each node is assigned by a cost. There are different statistics on the pixels between the edges that the cost may depend on them, for example the slope at the edges, the single bar's length, and the average intensity between the two edges. Noisy bars which refer to a false bar will receive a high cost but the bars which are located as real bars will receive a low cost.

When a cost is assigned to nodes based on the slope at edges, very clean and very well defined bars are represented by a large slope, also the sharpness degree of the transition from black to white or vise versa is indicated by a slope.

To accomplish best interpretation, assigning a cost to each node and transition must be observed carefully [9]. Assigning cost will be implemented by applying Dijkstra's algorithm which finds the best path through the bars by minimizing the sum of the costs of all the nodes and transitions in the path. Finally, only the path which leads to true bars will be identefied and all other noise will be skipped.

Both the length and intensity of each bar is considered in assigning a cost of transation. To reduce the problem of assigning cost to bars with incorrect length, A higher cost is given gray colored bars rather than dark black or pure white bars.

The Minimum Path Approach for barcode decoding is incapable of decoding extremely blurry barcodes and needs more work to achieve perfect results. Comparing this approach with the Hidden Markov Model, it is more capable of getting better results. It is easer and more conjectural.

CHAPTER III
MAJOR ELEMENTS IN THE PROPOSED IMAGE BASED BARCODE READER

3.1 BARCODE IMAGE

Barcode images are digital images containing barcodes in different types and resolutions. They can be colored format RGB (Red Green Blue), a bitmap image (BMP), GIF (Graphics Integration Formats), PNG (Portable Network Graphics) and TIFF (Tagged Image File Format). In order to read a barcode from image, that image first must preprocessed by applying several MATLAB operations.

3.1.1 Grayscale Image

Grayscale images are represented as a data matrix. The values of these matrices perform consistency within some range. Each one image pixel is denoted by one element of the matrix. Grayscale images differs from binary images in that they do not have only two colors (black and white) but they contain shades of gray varying from black at the weakest intensity to white at the strongest.

The data matrix of grayscale images can be of class unit 8, class unit 16, integer 16, single, or double.

3.1.2 Binary Image

A binary image can be defined as a digital image with typically two colors, black and white. It is also called bi-level or two level. Each pixel is stored as a single bit. 0 is assigned to background color which is black and 1 is assigned for the foreground color of an image which is white. It is mainly used in digital image processing as masks for morphological operations. It also can be used in segmentation, thresholding, and dithering as a result of these operations. This format also used in fax machines and document management solutions due to small size of image files. Binary images sometimes can be inverted so that 0 values are displayed as white and 1 values as black. This inverting operation can be done using the NOT operator in MATLAB.

3.2 SEGMENTATION

A digital image can be divided into several sections by applying segmentation methods. It's main purpose is to locate objects and boundaries in digital images. This can be done by labeling each pixel in the image so that each labeled pixel will share the same characteristics such as intensity, color, and texture. Segmentation can be considered a useful method in many applications. It is used in medical imaging for locating tumors, diagnosis, computer guide surgery. Also it is used in face , iris and fingerprint detecting. Segmentation has many methods and different techniques. Some of the methods are thresholding, histogram methods, edge detection, and clustering.

3.2.1 Thresholding

It is one of the segmentation's simplest techniques and usually considered the first step of it. Thresholding is done by assigning a threshold value which can be chosen either manually or automatically using automatic threshold algorithms. Each individual pixel of an image is labeled depending on the threshold value. If the threshold value is less than the marked image pixels then it is considered an object pixel with a value of '1' , otherwise it is a background pixel with a value of '0'.

3.2.2 Histogram

Image segmentation using histogram based method is considered more effective than other methods. This method works by scanning all of the pixels of an image and then using the captured picks and valleys in the histogram to locate image's clusters. Sometimes, it could be so hard to identify peaks and valleys in the image which will lead to an obstacle on histogram based method. This obstacle can be removed and solved by applying the histogram on multiple frames and merging the results to distinguish the peaks and valleys more efficiently.

3.3 FOURIER TRANSFORM

Fourier Transform is a representation of an image in the frequency domain which disintegrates the image into it's real and imaginary components. The number of frequencies in the frequency domain is equal to the number of pixels in the image or locative domain.

3.3.1 Fast Fourier Transform

The Fast Fourier Transform (FFT) is used in digital image processing as an efficient implementation of Discrete Fourier Transform(DFT). It's main use in digital image processing is to convert an image into frequency domain from it's image or spatial domain. The purpose of using (FFT) is to increase the efficiency and speed of applying filters to images because they are applied more faster in the frequency domain than to do in the image domain. Applying filters to images in frequency domain is done by convolutions. FFT turns the complicated convolution operations into simple multiplications. After applying filters by the FFT, an inverse Fourier Transform is applied then to obtain the result of the convolution.

3.3.2 High Pass Filter

Digital image processing applies high pass filters to digital images in order to implement image enhancements, modifications and noise reductions. The high pass filter is done in either spatial domain or the frequency domain.The implementation of high pass filter is done by duplicating the layer of the image, then putting a gaussian blur, inverting it and finally blending with the original layer using an opacity with the original layer. A generalization of high pass filter is called high boost filter which is done by applying the unsharp masking or sharpening operations to an image in image editing software.

CHAPTER IV
BARCODE RECOGNITION AND DECODING

4.1 PREPROCESSING THE IMAGE

The main concept of this project is to capture digital images, control whether they contain barcodes or not and then retrieve the barcode's information into readable numbers and characters. In order to read images containing barcodes several factors will affect barcode recognition operation. Some of these factors are blurriness and noise because blurry images will make barcodes not clear and the bars and spaces of the barcode could overlap to each other which will result to wrong readings of the barcode. Therefore, in this project we will consider only images without blurry to locate barcodes from them. To locate a barcode in a digital image first a number of operations must be applied to the image to make locating the barcode possible. These operations are called preprocessing the image.

4.1.1 Converting To Grayscale Image

During this project we had used the MATLAB software which is considered a technical computing language. MATLAB software has many ready to use functions under the image processing tools, using these functions facilitated our programming of the project. The first step in preparing the image to be read is converting a colored image into a grayscale image which is represented by an m x n matrix of integers. This operation is done by using the MATLAB function (rgb2gray).

4.1.2 Thresholding During this step we attempted to adjust our experimental image by applying thresholding to our grayscale image. To do this we have selected a sample row of the barcode in the image and plotted it's signal as shown in figures 16 and 17:

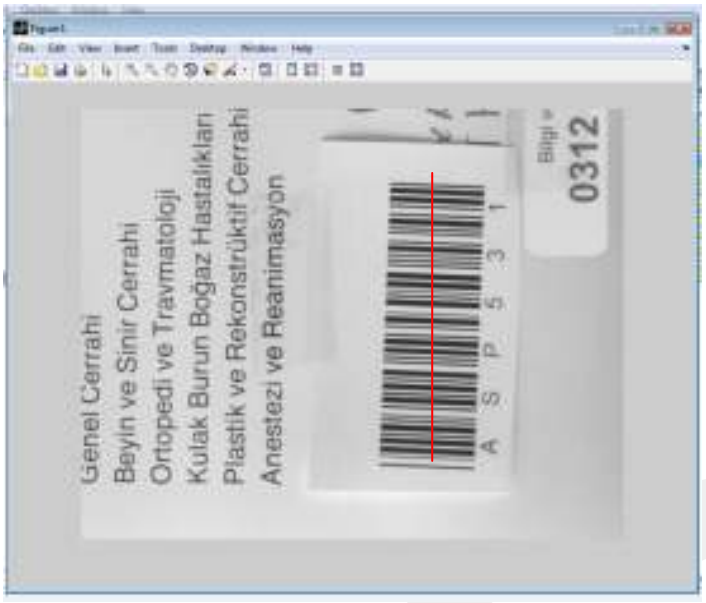


Figure 16. A Grayscale Image Including The Barcode

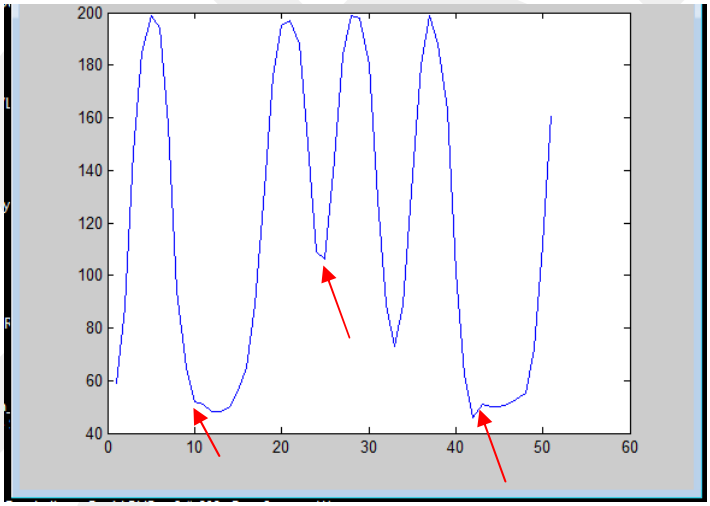


Figure 17. A Scan Line From Image. The Red Arrows Indicates Noise In Image

As seen in figure 17 there are some points in the image indicating noise which will affect evaluating threshold value. To eliminate these noises we have implemented a

moving average lowpass filter to make the edges more curvy which will lead to eliminating the noise. This moving average is done according to this equation:

$$Y[i] = \frac{1}{M} \sum_{j=0}^{M-1} x[i+j] \qquad (2)$$

Where x[ ] is the input signal, y[ ] is the output signal, and M is the number of points used in the moving average equation. After applying the moving average lowpass filter to our noisy signal the new signal became more curvy and less noisy as shown in figure 18 :

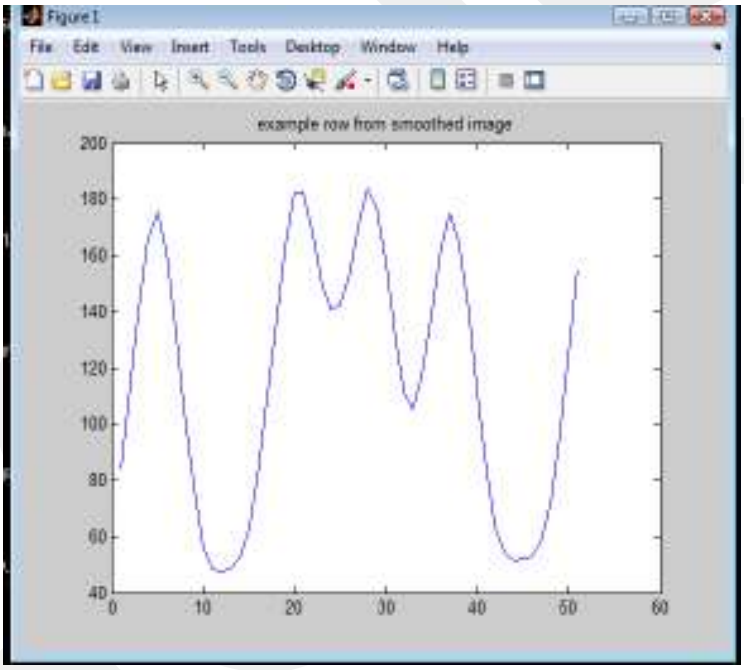


Figure 18. The New Signal With Noise Eliminated

Although the barcode image has been smoothed and the noisy signals became more curvy, but still locating an accurate threshold value difficult. As shown in figure 19, each peak in the signal has a different length and though if we applied thresholding to separate bars from spaces by locating a threshold value in the center of a long peak (red

dots), then the two short peaks (blue dots) will be accepted as on bar which causes loss in data of the barcode and thus a faulty detection.

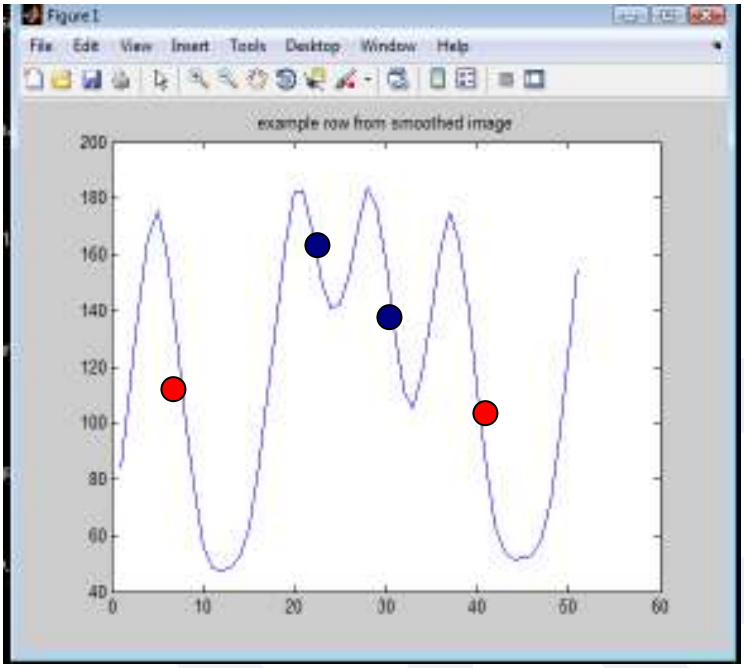


Figure 19. Long And Short Peaks

To overcome the problem of eliminating the noise in the image, we will apply a filtering method by scanning the grayscale image in both dimensions (horizontally and vertically) and calculate the second derivative of them according to row and column. This is done by first calculating the second derivative of the row of the grayscale image then examining it's zero crossing. This is done by counting each time when the absolute value of the second derivative according to the row of the image is greater than 4 when moving from peaks to valleys and counting each time when the same absolute value is greater than 0.2 when moves from valleys to peaks.

We want the moving from ones to zeroes more flexible and easy while moving from zeroes to ones will be hard because the barcode is considered clear and the remaining artifacts will be discarded using this method. The result of these operations will lead to this image:

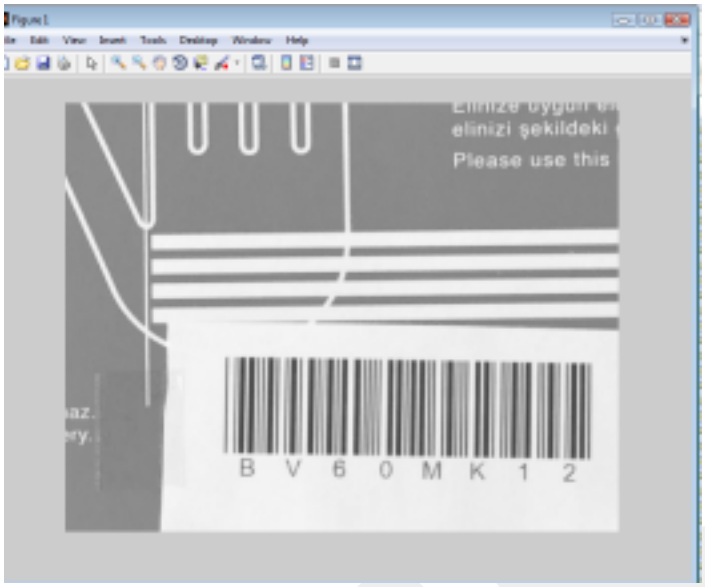


Figure 20. Original Grayscale Image

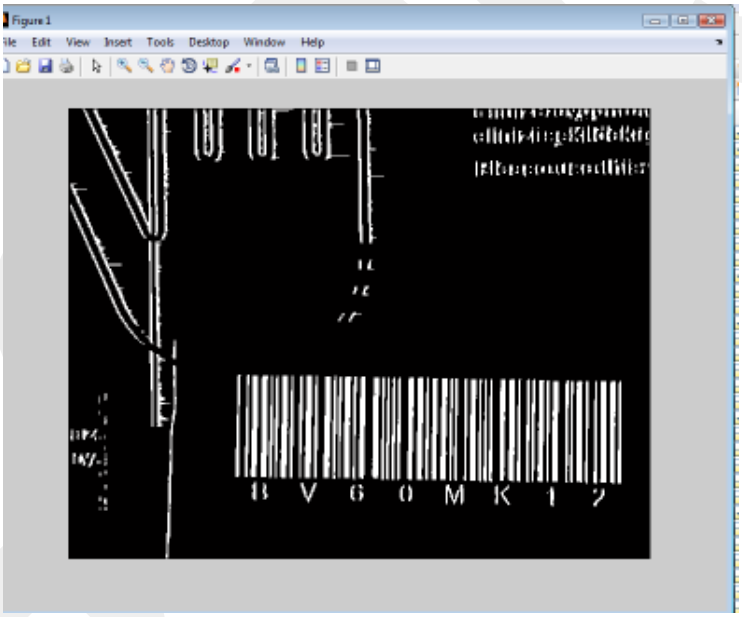


Figure 21. The Gray Image According To Row Dimension

After that we apply a function in the MATLAB software (bwconncomp) in order to calculate the connected components in the binary image and then filtering the binary image by checking the eccentricity of the connected components.

The eccentricity is the ratio of the distance between the foci of the ellipse and its major axis length. The value is between 0 and 1. (0 and 1 are degenerate cases; an ellipse whose eccentricity is 0 is actually a circle, while an ellipse whose eccentricity is 1 is a line segment.) This property is supported only for 2-D input label matrices.

Eccentricity of connected components are checked weather they are greater than 0.98 which will produce another filtered binary image representing the image horizontally :

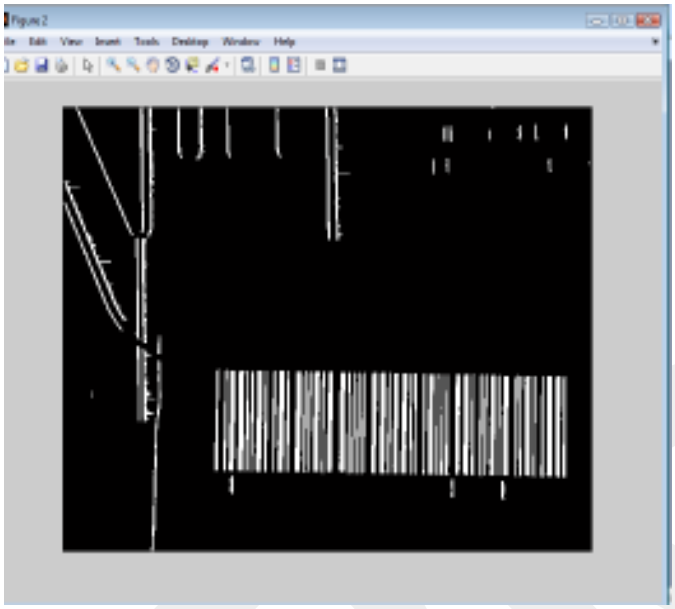


Figure 22. Horizontally Filtered Binary Image

These steps are applied to the image but this time all the operations will done according to the column of the original grayscale image resulting in these images:

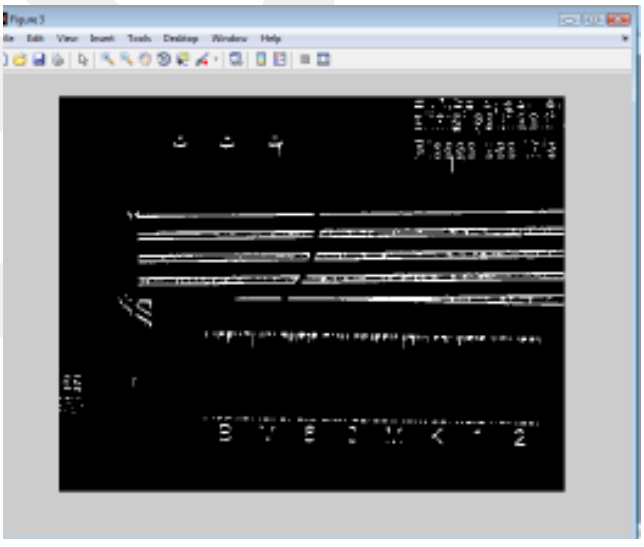


Figure 23. The Gray Image According To Column Dimension

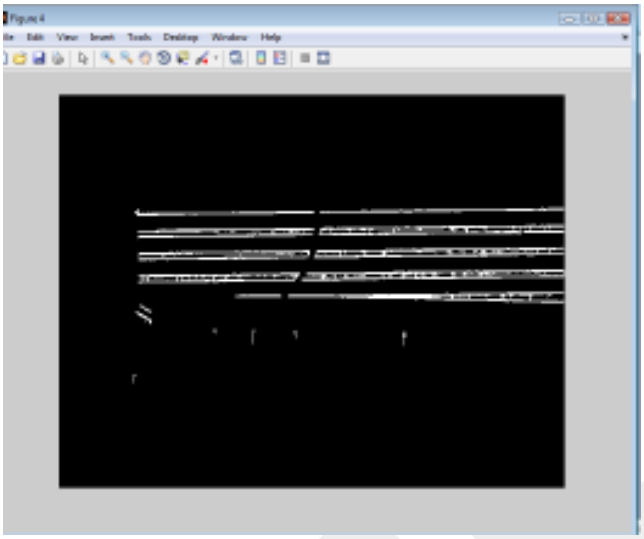


Figure 24. Vertically Filtered Binary Image

Since we did not know the direction of the barcode weather it is vertical or horizontal therefore we apply an OR operation to combine the information extracted from the two binary images into a single binary image without losing barcode data as shown in the following figure:

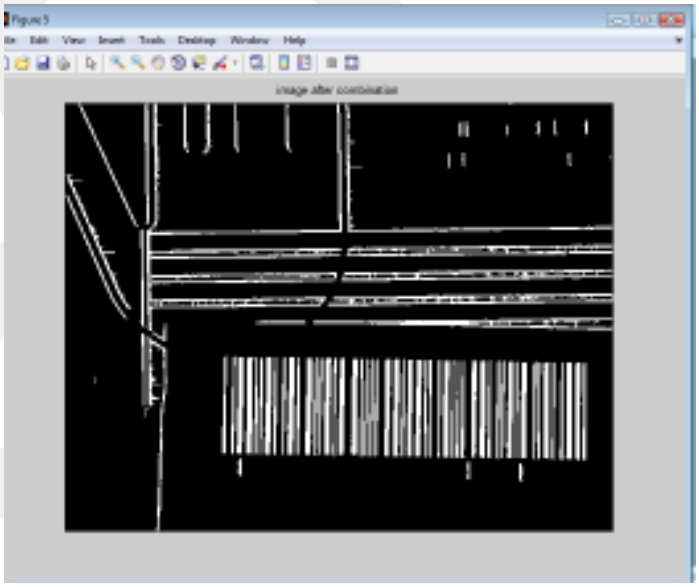


Figure 25. The Binary Image After Combination

4.1.3 Hough Transform

Hough transform is applied in this stage of the image preprocessing to read the barcode in the best way. While performing Hough transform, two parameters are used in this step which are rho and theta. The hough transform signal is represented in the following figure:

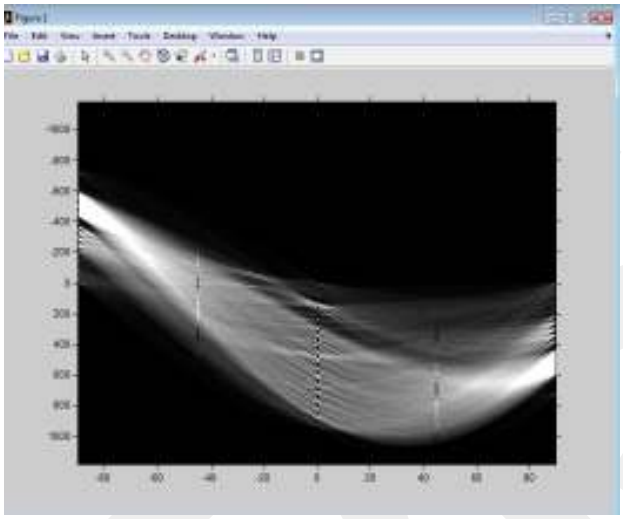


Figure 26. The Hough Transform Representation

To read the barcode, for each theta value we will calculate the first and second derivative of the rho. This is calculated by dividing the summation of the absolute value of rho's first derivative values by the summation of the absolute value of rho's second derivative values [11] . This is illustrated by the following equation:

$$C = \sum |r`| \; / \sum |r``| \qquad \text{For each } \theta \text{ value} \qquad (3)$$

Where C is the maximum theta value, r is the rho value and θ is the theta value.

The result of this equation will lead us to the maximum θ value which will be the ideal θ to read the barcode from the image. The resulted signal representation shown in the following figure:

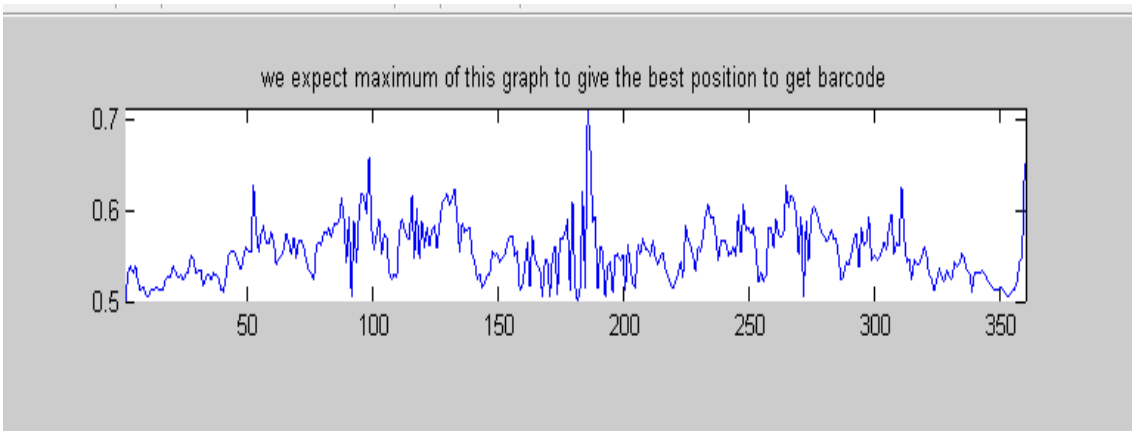


Figure 27. The Maximum Of Theta

Since barcode nature is having so many lines and when it is captured from a short distance it's lines in the image will have so many pixels, therefore the summation of the absolute value of the rho's first derivative values are expected to be big while the mentioned values of rho's second derivative values will not be big.

The resulted matrix of applying hough transform is considered and then we will take the corresponding column of this to the θ and read it.

Graph 1. Scanning Barcode In Theta Direction

This will produce our basic barcode. As shown :

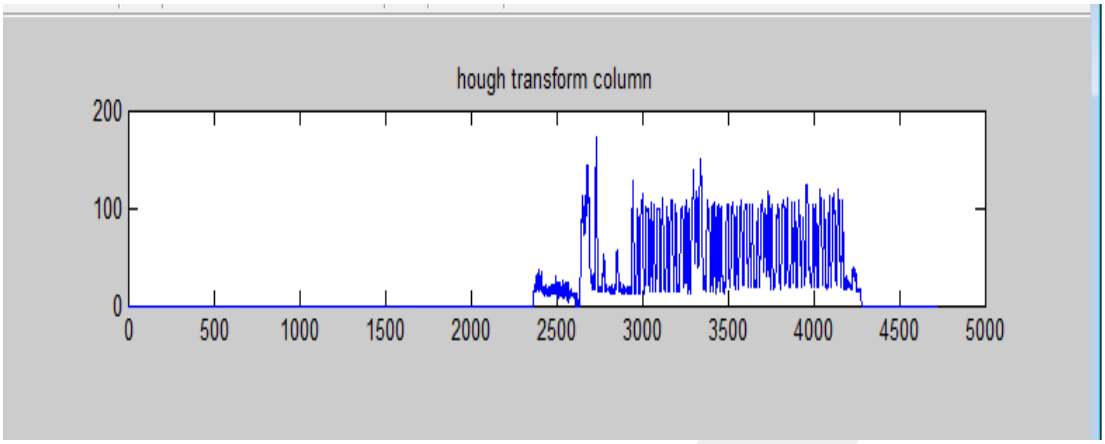


Figure 28. Hough Transform Column

During preprocessing, analyzing the obtained maximum value of hough transform, will increase the probability of viewing barcode's angle. Starting from this angle we will increase the theta value 90° and scan it by taking a side view of the barcode.

Graph 2. Scanning Barcode From Top To Bottom

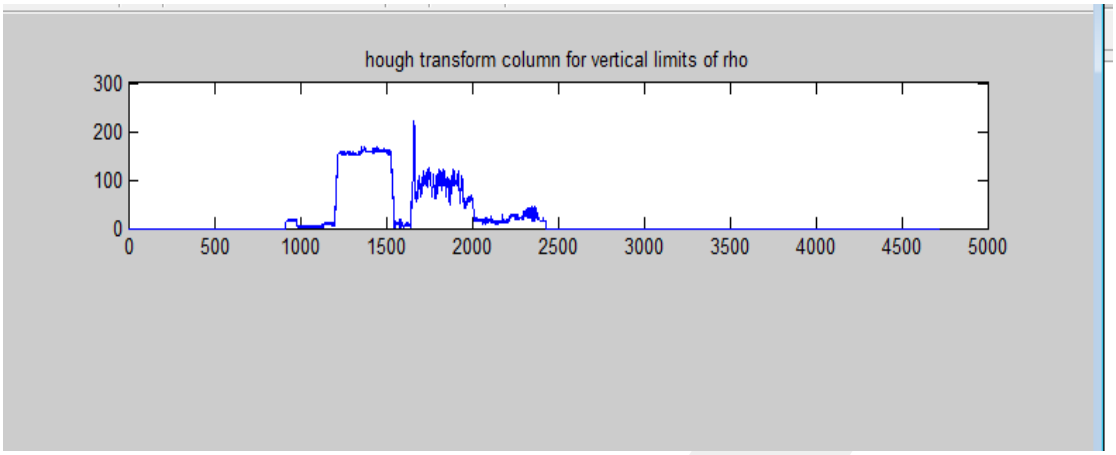


Figure 29. Hough Transform Column For Vertical Limits Of Rho

4.2 DETECTING BARCODE

This is the second important stage of image based barcode reading. It includes several operations following image filtering. The main step is done by first clearing all the noise existing above and below the barcode and secondly finding the beginning and end of the barcode and erasing the remaining noise and lines exists on both ends of it.

Now we will identify barcode's beginning values rho1 and ending rho2 values. To do this we will benefit from the fact that the derivative of the barcode vertically is 0 therefore the scanning line of hough transform when it is vertical passing through the barcode it will produce a stable and high value, even when the existed noise in the image exceeds. This is illustrates in the figure below:

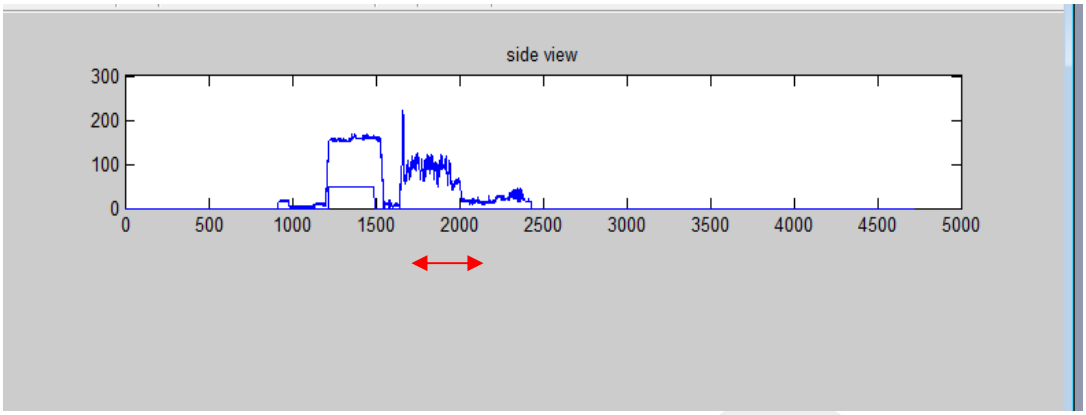


Figure 30. Side View Representation

The probability of the continuity of this high value is very low, because previously applied eccentricity filter has removed all the massive objects in the image also the applied moving average high pass filter has eliminated approximately all the sharp maximum signals keeping one maximum signal. The maximum value of the result of this filter (output = A) is considered in the barcode plateau and computing the threshold value by multiplying the maximum value of (A) by 0.8. After applying this threshold on the output (A) a binary array will be produced. In this array, the first point of non-zero area is denoted by (rho1), while the last point of non-zero area is denoted by (rho2). According to this labeling, the lines of (rho1, $\theta + 90°$) , (rho2, $\theta + 90°$) are named (L1) and (L2) respectively. The data in the image except that inside of (L1) and (L2) will be completely converted into 0's. The resulted image is shown in the next page.

Figure 31. Clearing Upper And Lower Boundaries Of The Barcode

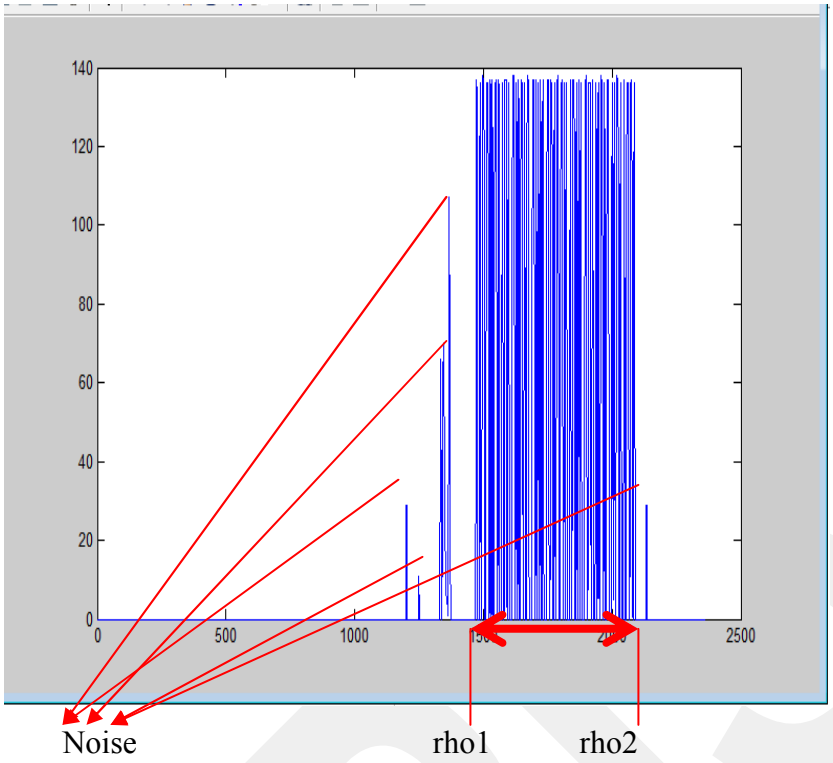


Figure 32. Representation Of Rho1 & Rho2

There are two main reasons of not using a scan line directly over the barcode to read it's data, these are:

1. After applying previous filters on the image the barcode area still not very smooth and by applying hough transform to the image this roughness will be lost in the overall total leaving a more smooth and net image with barcode to be able to read it easily.

2. Hough transform will increase the probability of reducing the noise existing in the right and left sides of the barcode, because for a noise to be able to opposes reading a barcode this obstacle must exceed the two lines (L1 , L2) also noise's lines must be parallel to the lines of the barcode. Therefore we apply hough transform for the second time.

This time we will apply the second hough transform by considering θ angle. The result of applying hough transform will be a one dimensional array which we denoted by [B]. After producing array [B], thresholding is performed by multiplying the maximum value of array [B] by a threshold value (0.8) producing a new array. This array has the barcode's crude data. To detect the barcode we must get rid of the zero's in the beginning and end of the array producing array [C]. Array [C] is the final detected

barcode data as 0's and 1's with counted beginning and end points. The figure below represents the extracted barcode data:

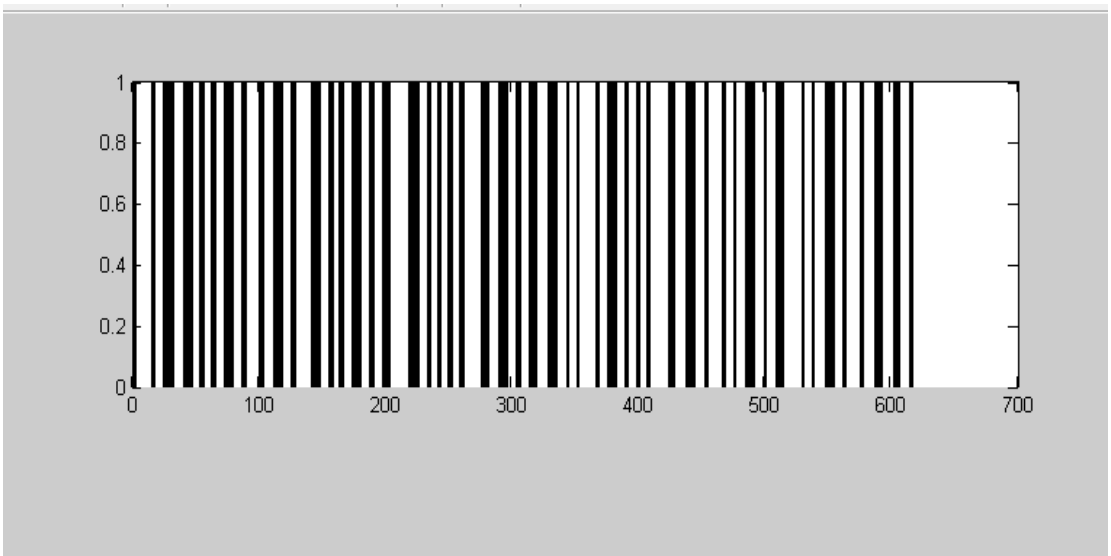


Figure 33. The Extracted Barcode

4.3 BARCODE DECODING

4.3.1 Code 39 Barcode

Code 39 is considered as a standard for many government barcode specifications, including the U.S. Department of Defense. The American National Standards Institute (ANSI) has defined code 39 a standard MH10.8M-1983.

The main property of this type of barcodes that they are alphanumerical barcodes with unlimited size of data. It's character set includes the digits 0-9, the upper case letters A-Z, and the following special characters: space, minus (-), plus (+), period (.), dollar sign ($), slash (/), and percent (%). Another important property about this type of barcodes that it always starts and ends with a special (*) character. Each character of code 39 barcode consists of 9 elements including 5 bars and 4 spaces. Each of includes 3 wide and 6 narrow elements which gave it the name 3 of 9 barcode. The characters are separated by an inter- character gap which is the same width as narrow space.

The main advantage of code 39 is that it does not need to generate a check digit and hence it can easily be inserted into existing printing system by adding a barcode font to the system or printer and then printing the raw data in that font.

4.3.2 Locating Bars And Spaces

This stage is composed of a series of operations in order to separate bars from spaces and making them readable to extract barcode information from them. After filtering the image several times the image had been enhanced and the barcode was detected clearing all the noise and objects bounding the barcode.

The last step in detecting barcode was creating an array [C] containing all barcode data stored in the bars and spaces of the barcode.

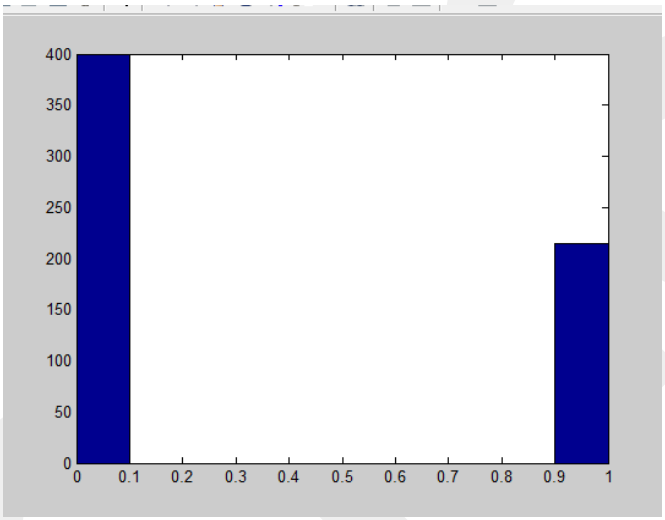


Figure 34. Histogram Of The Barcode Containing Bars & Spaces

The histogram of the barcode is illustrated above which is a double type of 1 x 615 because it is still having both bars and spaces. Our goal is to separate the bars from spaces by creating two arrays and scanning the barcode from the beginning to end of the barcode. This is done by counting each time it faces weather a bar or a space in a loop and storing the number of pixels that every following up bars and spaces occupy. After analyzing the distribution of 0's and 1's widths separately the result would be two histograms, one for bars and the other for spaces including wide and narrow types. By counting the maximum values of narrow spaces and narrow bars respectively stacks will be created. Considering these stacks the separating values between these stacks will be detected. Finally depending on these values two histograms will be created showing the narrow bars, wide bars respectively in the (Bar Width Histogram) and narrow spaces, wide spaces respectively in the (Space Width Histogram).

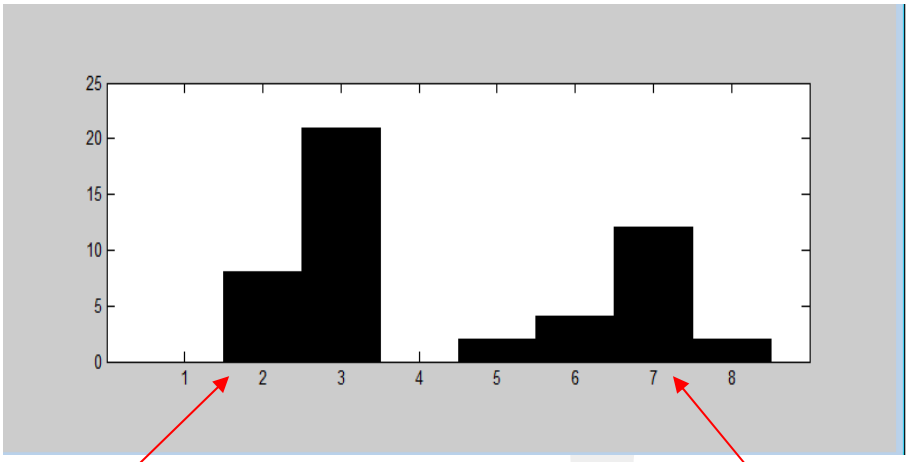


Figure 35. Bar Width Histogram

According to our experimental barcode image after executing our project in MATLB software the result of bar histogram array will be as follows: [0,9,21,0,2,4,12,2]. These numbers refers to the number of the repetition of one's and zeroes on each pixel. For example on pixel number (2) ones are repeated 8 times and so on.

This histogram also indicates the actual number of bars in our experimental barcode. This is performed by summing the elements of the histogram , this equation illustrates the operation:

$$n(b) = \sum_{i=1}^{m} (x(i) + x(i+1) + \ldots + x(m)) \qquad (4)$$

where n(b) is the number of bars or spaces , x is the element of the histogram, m is the length of the histogram. In our example this equation will be executed as follow:

n(b) = 0 + 9 + 21 + 0 + 2 + 4 + 12 + 2

= 50

These steps also repeated to the space width histogram which indicates the number of narrow spaces and wide spaces respectively then the total number of the spaces will be identified. The figure below shows the space width histogram.

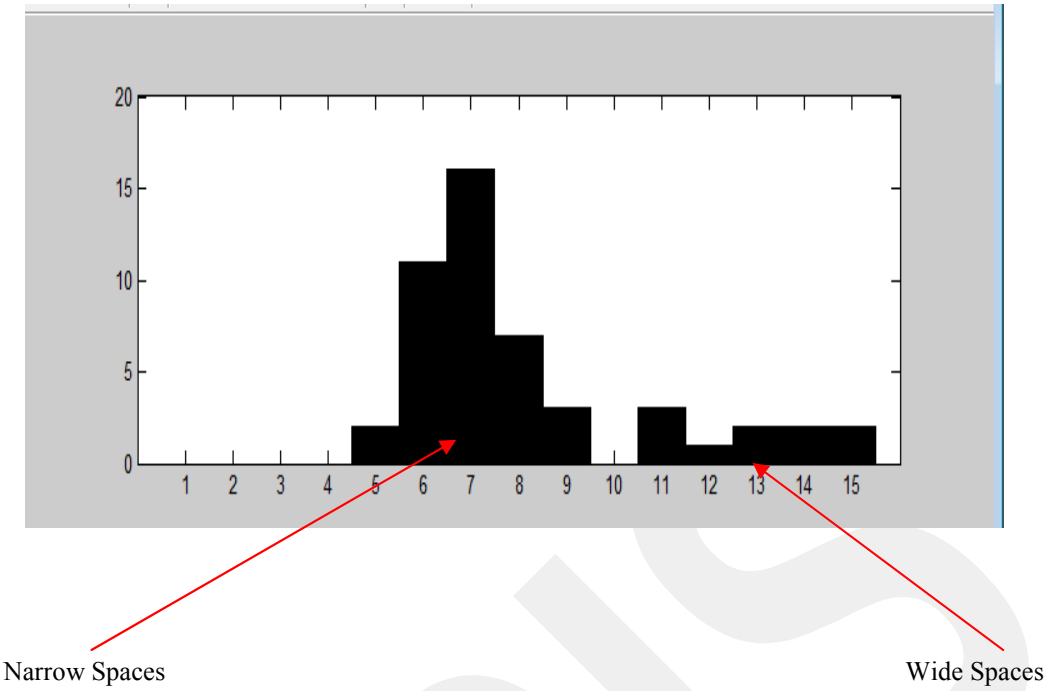


Figure 36. Space Width Histogram

4.3.3 Decoding

Barcode decoding is the last stage of this project which has an important role in reading the data stored in the 39 code barcode. After determining the maximum value of both bars and spaces several operations will be performed to be able to locate the actual positions of bars and spaces and exactly how many pixels they allocate.

```
barMax = max(barHist);
barMaxIndex =(find(barHist ==barMax));
barMaxIndex = barMaxIndex(end);
%we expect the value of barMaxIndex(1) to show the narrow bar
width median
i = barMaxIndex;
k = 1;
while( barHist(i) >= barHist(i+1))
    i = i+1;
    k = k+1;
end
BarWidthThreshold = barXout(i);
BarLowerThreshold = BarWidthThreshold- (2*(BarWidthThreshold -
barXout(barMaxIndex)))-1;
BarSecondMaxIndex = find (max(barHist(i:length(barHist))) ==
barHist);
BarSecondMaxIndex = BarSecondMaxIndex(end);
BarUpperThreshold = 1 + BarWidthThreshold +
(2*(barXout(BarSecondMaxIndex)-BarWidthThreshold ));

spaceMax = max(spaceHist);
spaceMaxIndex = find(spaceHist == spaceMax);
spaceMaxIndex = spaceMaxIndex(end);
i = spaceMaxIndex;
k = 1;
while( spaceHist(i) >= spaceHist(i+1))
    i = i+1;
    k = k + 1;
end
spaceWidthThreshold = spaceXout(i);
spaceLowerThreshold = spaceWidthThreshold-
(2*(spaceWidthThreshold - spaceXout(spaceMaxIndex)))-2;
spaceSecondMaxIndex = find (max(spaceHist(i:length(spaceHist)))
== spaceHist);
spaceSecondMaxIndex = spaceSecondMaxIndex(end);
spaceUpperThreshold = 8 + spaceWidthThreshold +
(2*(spaceXout(spaceSecondMaxIndex)-spaceWidthThreshold ));
```

Figure 37. Allocating The Actual Locations Of Bars & Spaces

Decoding is performed by converting the array [C] which was produced in barcode detecting stage into a binary array. During this, each wide bar is represented by [11], narrow bar as [1], wide space [00], and narrow space as [0] respectively. Following that each element will be compared with the ASCII characters of each element in the barcode. The ASCII table and the barcode encoding is more clear in this table:

Table 4.1 The ASCII Characters Table

ASCII Code	ASCII CHAR	WIDTH ENCODING	BARCODE ENCODING	ASCII Code	ASCII CHAR	WIDTH ENCODING	BARCODE ENCODING
48	0	NNNWWNWNN	101001101101	77	M	WNWNNNNWN	110110101001
49	1	WNNWNNNNW	110100101011	78	N	NNNNWNNWW	101011010011
50	2	NNWWNNNNW	101100101011	79	O	WNNNWNNWN	110101101001
51	3	WNWWNNNNN	110110010101	80	P	NNWNWNNWN	101101101001
52	4	NNNWWNNNW	101001101011	81	Q	NNNNNNWWW	101010110011
53	5	WNNWWNNNN	110100110101	82	R	WNNNNNWWN	110101011001
54	6	NNWWWNNNN	101100110101	83	S	NNWNNNWWN	101101011001
55	7	NNNWNNWNW	101001011011	84	T	NNNNWNWWN	101011011001
56	8	WNNWNNWNN	110100101101	85	U	WWNNNNNNW	110010101011
57	9	NNWWNNWNN	101100101101	86	V	NWWNNNNNW	100110101011
65	A	NNWWNNWNN	110101001011	87	W	WWWNNNNNN	110011010101
66	B	NNWNNWNNW	101101001011	88	X	NWNNWNNNW	100101101011
67	C	WNWNNWNNN	110110100101	89	Y	WWNNWNNNN	110010110101
68	D	NNNNWWNNW	101011001011	90	Z	NWWNWNNNN	100110110101
69	E	WNNNWWNNN	110101100101	45	-	NWNNNNWNW	100101011011
70	F	NNWNWWNNN	101101100101	46	.	WWNNNNWNN	110010101101
71	G	NNNNNWWNW	101010011011	32	SPACE	NWWNNNWNN	100110101101
72	H	WNNNNWWNN	110101001101	36	$	NWNWNWNNN	100100100101
73	I	NNWNNWWNN	101101001101	47	/	NWNWNNNWN	100100101001
74	J	NNNNWWWNN	101011001101	37	+	NWNNNWNWN	100101001001
75	K	WNNNNNNWW	110101010011	42	%	NNNWNWNWN	101001001001
76	L	NNWNNNNWW	101101010011	n/a	*	NWNNWNWNN	100101101101

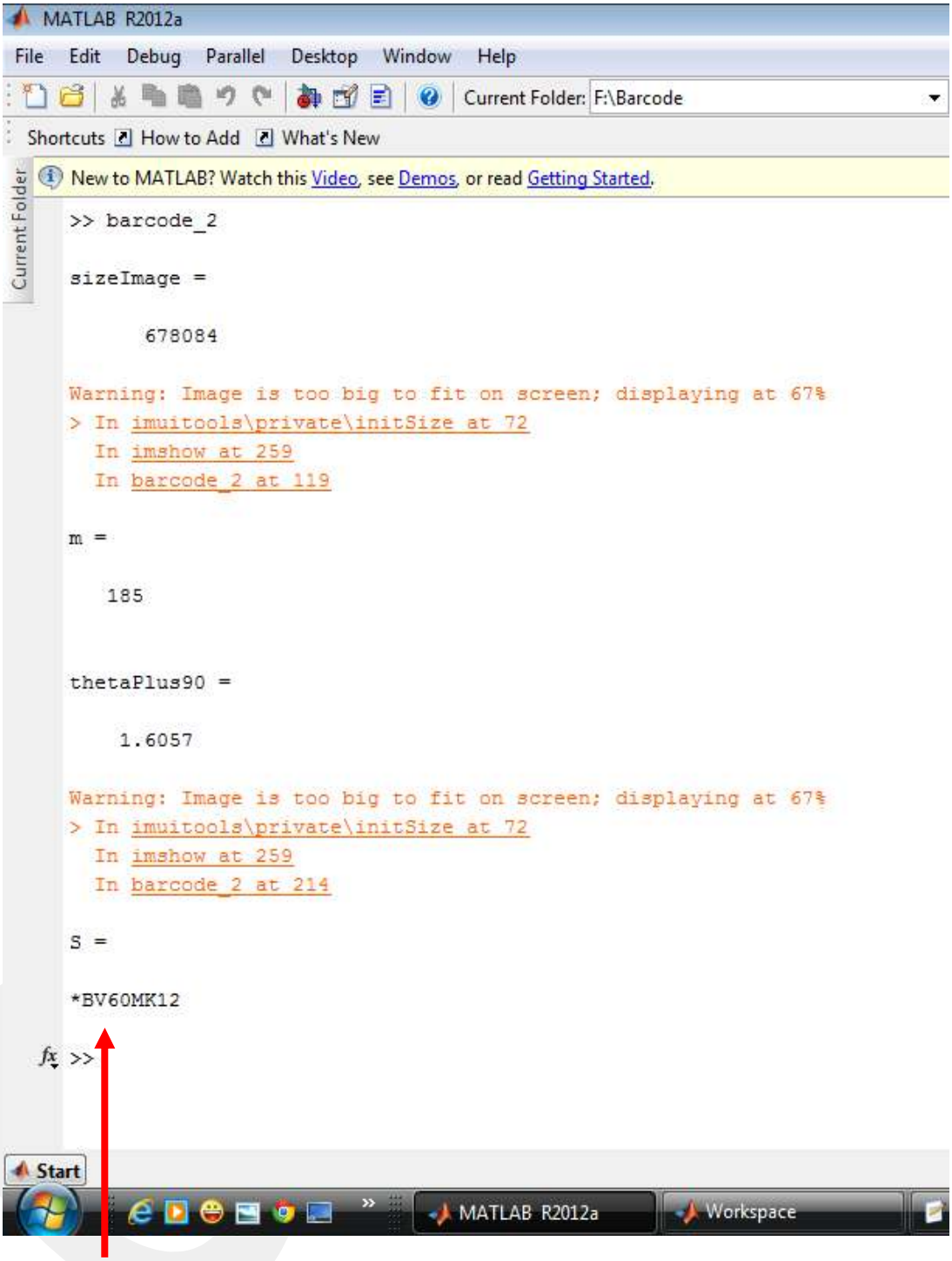
Where N= narrow , W= wide.

By checking each element in the barcode and compare it with it's corresponding ASCII character list, each bar and space of the barcode will converted to a readable form of characters and numbers showing the data stored in it. This is performed by creating a loop in MATLAB and convert each bar/space into it's binary value. Since each

character in code 39 barcode consists of 5 bars and 4 spaces and 3 of them are always wide then this loop will check every sequential 12 pixels and compare it with the values of ASCII characters and save it's corresponding value in an array. This check continues until all the pixels values are converted into symbols.

In our image after running the project the returned data is * BV60MK12. After comparing it with the original barcode image it retrieved to us the correct data.

Figure 38. The Original Test Image1 With Barcode



Barcode Data

Figure 39 . Decoded Barcode Data Of Test Image1

CHAPTER V

EXPERIMENTAL RESULTS

Our project has been implemented using MATLAB R2012a software , a Microsoft® Windows Vista™ Home Basic with a processor Intel(R) Core(TM)2 Duo CPU T5870 @ 2.00GHz, 2001 Mhz, 2 Core(s), 2 Logical Processor(s).

We have tested this project on many images to check it's authenticity and we saw that it works on images that have a high resolution and good quality with barcodes which are in the right direction, diagonal, and even upside down. Our algorithm fails on blurry images, images with multiple barcodes, vertically or horizontally curved barcodes, and barcodes on wrinkled papers. While implementing the project we had counted the elapsed time required to process each image at a time. The below table represents time table:

Table 4.2 Execution Time Table

Image Dimensions	Elapsed Time In Seconds
963 x 758	55.2282 sec
934 x 726	52.9385 sec
940 x 1083	85.7400 sec
930 x 919	51.720 sec
1072 x 712	70.325 sec
787 x 504	51.513 sec
623 x 437	48.971 sec
665 x 618	50.26 sec
600 x 800	52.3122 sec
1071 x 568	65.513 sec

Some of the test images have been resulted as shown:

Figure 40. Test Image2

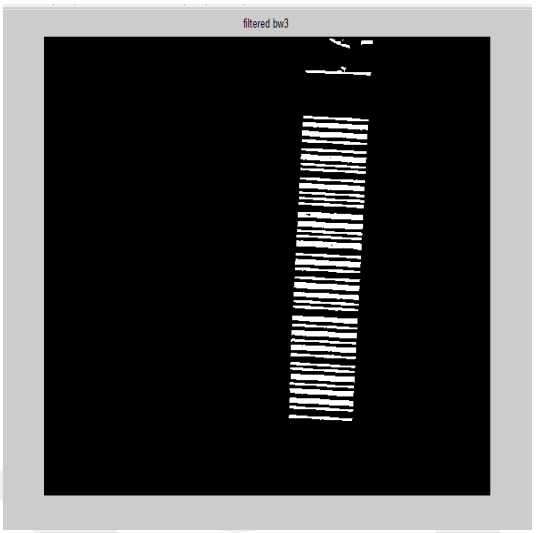
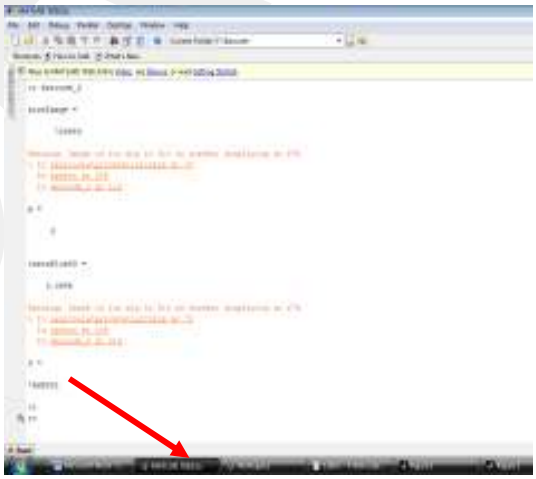


Figure 41. Filtered Image

Figure 42. Extracted Barcode



ASP531

Figure 43. Decoded Barcode Data

Figure 44. Test Image 3

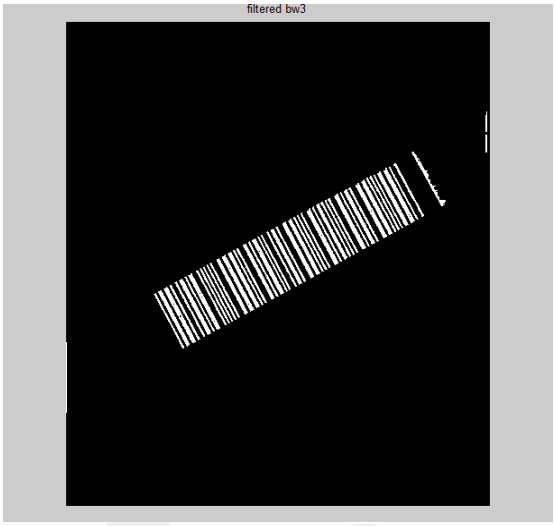


Figure 45. Filtered Image

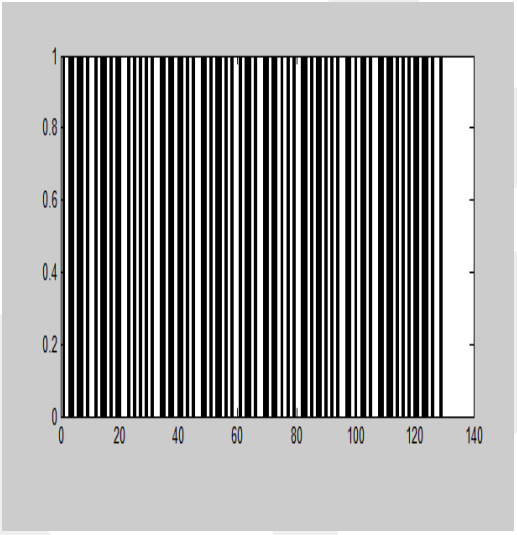
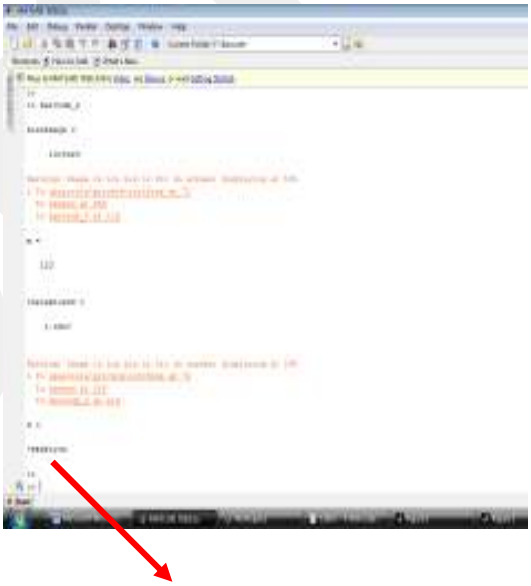


Figure 46. Extracted Barcode



TEST1234

Figure 47. Decoded Barcode Data

Figure 48. Test Image 4

Figure 49. Filtered Image

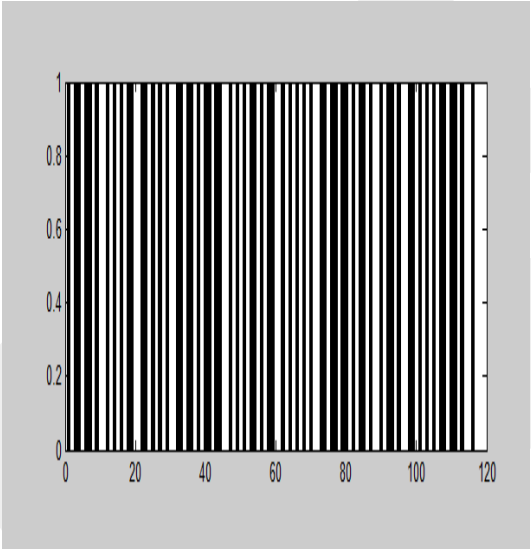


Figure 50. Extracted Barcode

DX34GF6

Figure 51. Decoded Barcode Data

Our algorithms for barcode detection and decoding also used on images containing barcodes but failed to detect and decode them because of blurriness, missing parts of barcode, existing in curved surfaces or wrinkled papers, and a case of multiple barcodes

existing in the same image. The image below explain the cases where our project fails to process them:

Figure 52. Blurry Image

Figure 53. Partial Missing Barcode

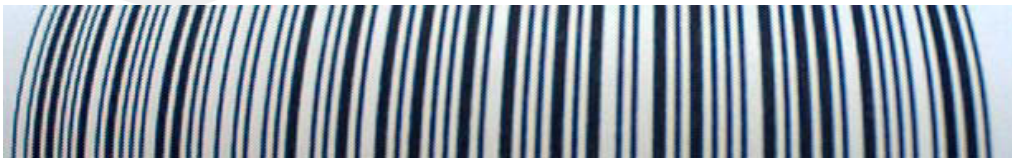


Figure 54. Vertically Curved Barcode

Figure 55. Barcode On Wrinkled Paper

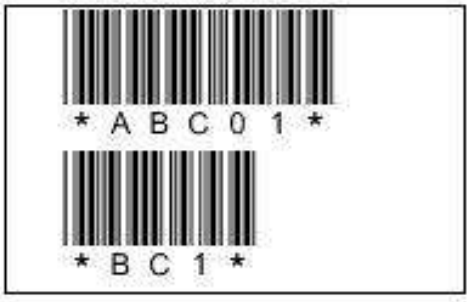


Figure 56. Multiple Barcodes In Same Image

CHAPTER VI

CONCLUSIONS AND FUTURE WORK

This theses presents an image based barcode reader system, explaining it's performance and implementation. The goal of this project is to implement a system which is able of detecting barcodes of type code 39 barcode (code 3 of 9) from images and decoding them to show the barcode information stored in it.

Tested images were captured by a NIKON D90 camera machine, with barcode samples in different angles. The project is divided into three main parts: preprocessing the image, barcode detecting and decoding. Preprocessing and detection sections were the most hard parts of the project, the images passed through several filtering techniques to prepare the image for detecting.

Preprocessing include first smoothing the image using the moving average lowpass filter then scanning the image both horizontally and vertically with measuring the second derivative of both rows and columns. Then using the connect region and eccentricity properties codes in MATLAB. Because we do not know the exact location of the barcode yet we applied an OR operation to combine the results of these operations.

Hough transform has an important role in detecting the barcode. It is used first in detecting the maximum value of detected lines in the image according to theta value which will increase the probability of viewing barcode angle.
Thresholding is used in this stage then the hough transform is used for the second time in clearing the upper and lower boundaries of the barcode by locating two lines and keeping only the noise inside these two lines. The result of this will be a binary array having the information of the barcode.

Decoding is performed by scanning barcode from start to end point to separate bars from spaces and count the number of pixels that each bar/space occupies and then converting them to their corresponding binary values (1 for bar, 0 for space). Finally the binary values are compared with their corresponding ASCII characters and the barcode data are displayed.

Future Work

As we mentioned before, our project works only on images with high resolution and decodes only one type which is code 39 barcode. To develop this project several techniques are required in the future to detect barcodes not only in very clear images but in blurry images too. The time average of running this project is 58.45 sec which could be reduced to have a better performance.

Also considering other types of linear barcodes other than code 39 barcode will be a good step to get a better usage of this work.

REFERENCES

[1] Tropf, A. and Chai, D. (2006) "Locating 1-D Bar Codes In DCT-Domain". Acoustics, Speech and Signal Processing. IEEE Conference Publications. Volume 2.

[2] Janapriya, R. ,Kularatne, L., Pannipitiya, K., Gamakumara, A. and Silva, C. (2003) "A Low Cost Optical Barcode Reader Using A Webcam". Engineering Research Unit (ERU) Symposium, Sri Lanka.

[3] Muniz, R. , Junco, L. and Otero, A. (1999) "A Robust Software Barcode Reader Using The Hough Transform,".Proceedings 1999 International Conference on Information Intelligence and Systems,. p 313

[4] Öktem, R. (SIU2004, 28-30 April) "Barcode Localization Algorithm Using Binary Morphology in Wavelet Domain - Dalgacık Bölgesinde İkili Morfoloji Kullanarak Çubuk KodYersenimi".

[5] Öktem, R., Çetin, A. E. (SIU2005, 16-18 May 2005) " Barcode Localization By Image Processing – İmge İşleme Yoluyla Çubuk Kod yersenimi".

[6] Otsu, N. (1975) "A Threshold Selection Method from Gray-Level Histograms". Volume 11. pp. 285–296.

[7] Madej, D. (2003) "A Wavelet Based Speckle Noise Filtering In A Laser Scanner". In: Proceedings Of The 11th International Conference on Software, Telecommunications and Computer Networks. pp. 787–791.

[8] Kresic-Juric, S., Madej, D. and Santosa, F. (2005) "Applications Of Hidden Markov Models In Bar Code Decoding.". 27: 1665-672.

[9] Carlson, B., Joseph, E. and Lu, K. (2007) "Automatic Focusing System For Imaging-Based Bar Code Reader" . Symbol Technologies, Inc.

[10] Gonzalez, R. and Woods, R. (2008) "Digital Image Processing ". University of Tennessee.

[11] Sağdıçoğlu, B. (2012) . MATLAB lecture notes.

[12] MathWorks – MATLAB and Simulink for Technical Computing <http://www.mathworks.com>

[13] Wikipedia, the free encyclopedia. <http://en.wikipedia.org/wiki>

[14] Barcodes Inc, 200 W Monroe St, Chicago IL 60606. <http://www.barcodesinc.com/articles/types-of-barcodes-2.htm>

[15] Google. 2012. <http://books.google.com.tr/books>

[16] Barcoding Incorporated <http://www.barcoding.com/upc>

APPENDIX

CURRICULUM VITAE

PERSONAL INFORMATION

Surname, Name: FAWZI, Thuraya

Nationality: Iraqi

Date and Place of Birth : 1 July 1982, Kirkuk-Iraq

Marital Status: Married

Phone: +90 530 041 11 32

Email: thurayasami@yahoo.com

EDUCATION

Degree	**Institution**	**Year Of Graduate**
M.S.	Çankaya University, Computer Engineering	2012
B.S.	Kirkuk Technical College, Software Engineering Department	2004
High School	Kirkuk High School, Kirkuk	2000

FOREIGN LANGUAGES

English, Turkish, Arabic