A COMPUTER VISION BASED BARCODE READING SYSTEM

A MASTER'S THESIS

in

Electrical & Electronics Engineering

Atılım University

by

MEHMET İLHAMİ SAFRAN

AUGUST 2008

A COMPUTER VISION BASED BARCODE READING SYSTEM

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

OF

ATILIM UNIVERSITY

BY

MEHMET İLHAMİ SAFRAN

IN PARTIAL FULLFILMENT OF THE REQUIREMENTS FOR

THE DEGREE OF

MASTER OF SCIENCE

IN

THE DEPARTMENT OF ELECTRICAL & ELECTRONICS

ENGINEERING

AUGUST 2008

Approval of the Graduate School of Natural and Applied Sciences

Prof. Dr. Abdurrahim Özgenoğlu

Acting Director

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

Asst. Prof. Dr. Ali Kara

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.

Asst. Prof. Dr. Ruşen Öktem

Supervisor

Examining Committee Members

Asst. Prof. Dr. Ruşen Öktem

Assoc. Prof. Dr. Elif Derya Übeyli

Asst. Prof. Dr. Elif Uray Aydın

....

....

Date: August 13, 2008

I declare and guarantee that all data, knowledge and information in this document has been obtained, processed and presented in accordance with academic rules and ethical conduct. Based on these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Mehmet İlhami Safran

Signature:

ABSTRACT

A COMPUTER VISION BASED BARCODE READING SYSTEM

Safran, Mehmet İlhami

M.S., Electrical & Electronics Engineering Department

Supervisor: Asst. Prof. Dr. Ruşen Öktem

August 2008, 65 pages

This study focuses on barcode recognition from images of barcode stamped products, acquired by different cameras. First step in recognition of barcode data is achieving barcode localization. For this purpose, a Fast Hough Transform Approximation has been developed. With this method, barcode localization and the angular orientation of the barcode is obtained. In order to decipher barcode data, the scan line obtained by barcode localization and angular orientation information are used. However, the image resolution has to be good enough so that the bar code data could be read properly. In this work, image quality is improved with methods of B-Spline Smoothing and super-resolution techniques. Later, barcode data is deciphered via match filtering**.** The proposed barcode reading system is tested and results are discussed.

Keywords: Barcode, Hough Transform, Fast Hough Transform, B-Spline Smoothing, super-resolution.

ÖZ

BİLGİSAYAR TABANLI BİR BARKOD OKUMA SİSTEMİ

Safran, Mehmet İlhami

Yüksek Lisans, Elektrik - Elektronik Mühendisliği Bölümü

Tez Yöneticisi: Yrd. Doç. Dr. Ruşen Öktem

Ağustos 2008, 65 sayfa

Bu çalışmada, değişik kameralarla elde edilmiş ürün imgelerinde barkod bilgisinin tanımlanması üzerinde durulmaktadır. Barkod bilgisinin tanımlanması için ilk olarak Barkod yerseniminin elde edilmesi gerekmektedir. Barkod yersenimi için, genellikle doğru bulmada kullanılan Hough dönüşümüne hızlı bir yaklaşıklık geliştirilmiştir. Geliştirilen bu yöntem kullanılarak barkod yersenimi ve barkod'a ait açısal yönelim bilgisi elde edilmiştir. Barkod bilgisinin okunmasında ise, barkod'a ait yersenim ve açısal yönelim bilgileri kullanılarak elde edilen tarayıcı doğru kullanılmaktadır. Ancak, Çubuk kod bilgisinin doğru okunabilmesi için imge çözünürlüğünün yeterince iyi olması gerekmektedir. Bu çalışmada, görüntü kalitesi B-spline yumuşatma ve süper çözünürlük teknikleri ile iyileştirilmektedir. Daha sonra uyuşma filtresi uygulanarak barkod bilgisi çözümlenmiştir. Barkod okuma sistemi test edilmekte ve test sonuçları tartışılmaktadır.

Anahtar Kelimeler: Barkod, Hough Dönüşümü, Hızlı Hough Dönüşümü, B-Spline Yumuşatma, süper çözünürlük

To My Parents

ACKNOWLEDGMENTS

This study has been carried out at Electrical-Electronics Engineering Department of Atılım University as a part of the PUSULAL Project during the years 2006-2008. The PUSULAL Project, 105E130, has been supported by TÜBİTAK (The Scientific and Technological Research Council of Turkey).

I would like to express my sincere gratitude to my advisor Asst. Prof. Dr. Ruşen Öktem for her guidance, suggestions and supports throughout this research. Also I would like to extend my appreciation to my friends, my colleagues for their valuable support and fellowship.

I am forever indebted to my family who have been supporting me in every aspects of life.

TABLE OF CONTENTS

LIST OF TABLES

TABLE

LIST OF FIGURES

FIGURE

LIST OF ABBREVIATIONS

PDA	-	Personal Digital Assistance
MP	-	Mobile Phone
HT	-	Hough Transform
FHT	-	Fast Hough Transform
B-Spline	-	Basis Spline
FHTA	-	Fast Hough Transform Approximation
SNR	-	Signal to Noise Ratio
RSS	-	Residual Sum Squares

CHAPTER 1

INTRODUCTION

Barcodes are very popular and are seen on almost every consumer products. Barcodes provide reliable data storing and quick input into a computer system. In this manner, price and product description can be obtained very fast. There exist various one dimensional barcode symbologies such as EAN13, EAN8, EAN128, UPC-A, UPC-E etc. However, mainly barcode consists of thick or thin parallel lines which are parallel to each other in the forms of rectangles.

Traditionally, a laser barcode reader is used to read the embedded code. However, recently, image processing based barcode reading systems started to gain importance which provide more information than laser barcode readers at a time [1]. Similarly, the use of Personal Digital Assistance (PDA) and Mobile Phone (MP) are becoming increasingly widespread. They can be used to access the internet and take pictures and videos. They can be even used as a barcode reader [2]. Thus, consumer or user can capture an image of a barcode by using their PDA/MP. Such devices, which are programmed to decipher the barcode image, will be able to obtain the barcode information. By connecting online, consumer or user can access product information such as price, product description and price comparison.

1.1 Scope of the Thesis

In this work, our aim is to develop an image processing based barcode recognition system in the scope of the project "PUSULAL: Görme Özürlüler için Alışveriş Merkezlerinde Yardım Pusulası / A Compass to Aid the Visually Impaired in Shopping Centers". Pusulal project aims to offer a navigation aid for visually impaired people to guide them through the aisles of a shopping centre and access information about products. The information access system includes a camera to read barcodes of a product and wireless communication to inquire information about that product through the barcode from the centre's database. In PUSULAL project, the mission of the barcode reading system is to decipher the barcode information from barcode images. Barcode images are acquired by visually-impaired people with a camera which captures images from different sides of the product (e.g. front, back, top and bottom sides). Block diagram of the system is shown in Figure 1. The system is divided into two main parts in order to increase the system performance. The first main part is localization of the barcode and the second part is decoding the information embedded in the barcode.

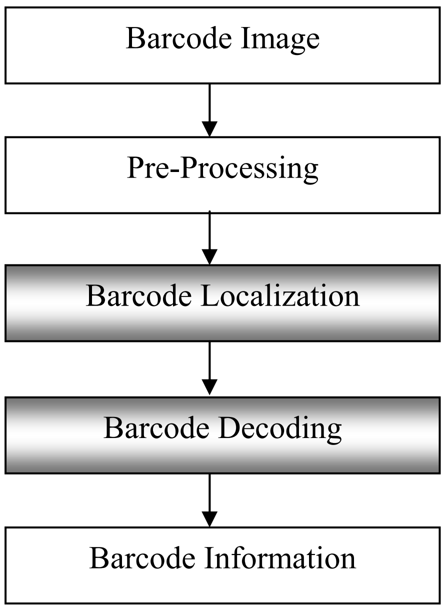


Figure 1. Block diagram of the barcode recognition system

Barcode Localization is a kind of image segmentation problem. Hence, some line and curve fitting algorithms such as Hough Transform (HT), Fast Hough Transform (FHT) and Basis Spline (B-Spline) were examined and Fast Hough Transform Approximation (FHTA) has been developed with the intention to minimize the computational time of the HT/FHT. The FHT and FHTA are proposed and tested over various images for barcode localization.

Image resolution has key role in barcode decoding. In this work, a super-resolution technique is used for increasing the image resolution. Barcode decoding is achieved by estimating the midpoints, hence widths of each line. Simulation results over test images are also presented.

1.2 Organization of the Thesis

Chapter 2 presents a literature survey on image processing based barcode recognition systems. Chapter 3 is an overview of the image segmentation and HT, FHT, FHTA and B-Spline techniques. Barcode Localization via the FHT and FHTA are introduces and experimental results are listed in Chapter 4. In Chapter 5, Barcode Decoding technique is presented. Chapter 6 discusses barcode localization, decoding and overall system performance with experimental results. Finally, in Chapter 7 some concluding remarks are stated on the performance of the localization, decoding and reading system.

CHAPTER 2

LITERATURE SURVEY ON IMAGE PROCESSING BASED BARCODE RECOGNITION SYSYTEMS

Use of image processing based barcode recognition systems are expected to increase the performance in barcode reading. Moreover, this enables PDA and MP users to be able to obtain barcode information via their devices. There exist some commercial image processing based barcode reading softwares. “Softek Barcode Reader” [3] and “DTK Barcode Reader SDK” [4] are two examples developed by DTK Software and Softek Software companies. However, in the literature, there exist few research works on that field. For the purpose of increasing the system performance and decreasing the computation time, the problem is generally divided into two parts; Barcode Localization and Decoding. The first part of the system finds the location of the barcode. The second part decodes the barcode information embedded in the barcode. As a result of localization, decoding algorithm only considers the barcode area but not the whole image.

In the literature, part of the works merely considers Barcode Localization [5, 6, 7], some only consider Barcode Decoding [8, 9, 10] and the others consider the whole problem [11, 12, 13].

In this Chapter, review of the Barcode Localization and Decoding techniques in the literature are presented. Limitations and drawback of those systems are mentioned.

2.1 Barcode Localization

Multichannel Gabor filtering technique has been used for many texture classification and segmentation tasks. In [5], this technique is used for barcode localization. Firstly input image is filtered by using Multichannel Gabor filter in eight orientations (0°, 22.5°, 45°, 67.5°, 90°, 112.5°, 135° and 157.5°). Thus, eight Gabor filtered images or feature images are obtained. Then, local energy of the feature images is examined by using either unsupervised method (K-Mean Clustering) or supervised method. As a result of this operation, barcode location and barcode orientation are found. In [11, 12], Gradient approach is used. The gradients of the images are computed along several directions (0°, 45°, 90° and 135°). Then, high density areas are processed.

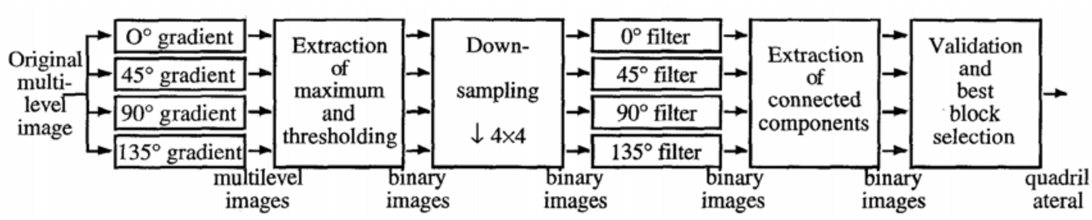


Figure 2. Block diagram of the barcode localization method [11]

Figure 2 illustrates the main block diagram of the barcode localization procedure in [11]. However, these methods are not examined in detail for different barcode samples at different orientations and different quality. Furthermore, their time complexity is so high that they are not used in real time applications.

Hough Transform (HT) is applied in many segmentation problems such as lines and regular shape detections. In [13], HT is also applied to barcode localization. After an edge detection operation, all non-zero edge pixels are mapped to Hough plane using line equation. Muniz *et. al.* [13] assumed that all barcodes lie on ninety degrees because they only study prescription images. Thus, Hough plane is examined and ninety degrees lines are captured. Then, barcode area is detected as the region of vertical lines. This work is presented as a case study and is not applicable to different product images at different orientations. In addition, computational time of this algorithm is not enough for real time applications because it requires many multiplications. [6] proposes a less time complex solution for barcode localization. Original image is mapped to wavelet domain by using a wavelet transform technique so vertical, horizontal and diagonal features are obtained. Then binary morphology (erosion and dilation) and size thresholding are applied to locate the barcode area. However, in this technique orientation of the barcode is not obtained. Wavelet transform based barcode localization technique is improved in [7]. In this technique, wavelet features are combined with logical OR operation for the purpose of obtaining a binary image. Binary morphology is applied to binary image with the intention to locate the barcode. Furthermore, in order to find the orientation of the barcode, free angle thresholding is developed and applied.

2.2 Barcode Decoding

After finding the location of the barcode, the remaining problem is to decipher the barcode. In order to decode the barcode information barcode resolution has a key role in decoding. It is assumed that barcode resolution quality is high enough in [8, 11, 12, 13]. High quality barcode samples are obtained from low resolution ones by using super-resolution techniques in [9, 10].

To decode barcode information, widths of the dark and light regions on the barcode must be known. Peak locations of the dark and light regions are found with statistical pattern recognition in [8]. Barcode information is deciphered based on the distance between the peak locations. In [11], simple thresholding is applied to all pixels. If the pixel value is less than threshold, it is replaced by zero. Then, number of zeros is counted to obtain widths of the barcode lines. Detection of the transition between the two barcode regions is used in [13]. Transition values are obtained by zero-crossings of the second derivative of the barcode. Then, distance measurement is performed between the light and dark barcode lines and barcode is decoded using the distance information. Finally, neural network is used for decoding in [12]. However, the performance of these methods depend severely on the image quality.

In [9, 10] a super-resolution method is proposed which enables the reconstruction of a higher quality barcode sample from a low resolution or low quality sample. [9] only considers the improvement of the barcode image but does not try to decipher it.

For this purpose Fourier Transform phase information and the truth that each scan line of a barcode codes the same information are used for improvement. In [10] super resolution algorithm generates a perfect resolution barcode sample from estimated start and end points and tilt angle and tries to estimate to minimize the mean square error between the generated and original samples.

CHAPTER 3

SEGMENTATION

Image segmentation is a low-level image processing task that aims at partitioning an image into homogeneous regions in terms of the features of pixels extracted from the image [14]. The main idea of the image segmentation is to simplify and change the representation of an image into something that is more meaningful and easier to analyze. Thus, image segmentation is an important research area in digital image processing with several applications in machine vision, product quality inspection, medical diagnosis, the analysis of remotely sensed images, etc.

Many techniques and algorithms have been developed for image segmentation. However, there is not unique solution to the image segmentation problem; they often have to be combined with domain knowledge in order to effectively solve an image segmentation problem.

There are a variety of segmentation criteria that involve models at a larger scale. Typically, one wants to decompose an image or a set of tokens which could be pixels, isolated points, sets of edge points, etc. into components that belong to one or another simple family. Finding such groups is often called fitting that is a part of the segmentation process, because it uses a model to produce compact representations that emphasize the relevant image structures [15].

Fitting technique is extremely useful in many applications such as object identification and approximation. For example, if the objects are characterized by presence of straight lines, it means that straight lines report the segmentation of the image.

This chapter reviews the line fitting and curve fitting techniques. Hough Transform, Fast Hough Transforms and our developed Fast Hough Transforms Approximation and also popular curve fitting technique Basis Splines are presented.

3.1 The Hough Transform

The Hough Transform (HT) is commonly used for the detection of the regular shapes such as lines, curves and ellipse on binary images [16]. In HT, the simplest slope intercept form of the line equation (1) is used for the detection of the straight lines in the image.

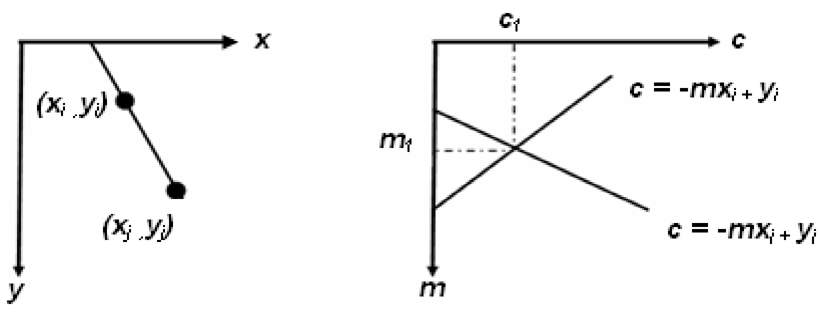
$$y_i = m \cdot x_i + c \tag{1}$$

where, *m*, *c*, *(x, y)* refer to slope, intersection of the y-axis and edge pixel coordinate pair, respectively.

Any line can be characterized by the parameters *m* and *c*. Each of the possible lines passing through point *(x, y)* has unique *(m, c)* value in parameter space (2);

$$c = -m \cdot x_i + y_i \tag{2}$$

Every point in the image space *(x, y)* corresponds to a line in parameter space *(m, c)* and each parameter space *(m, c)* corresponds to a line in image space *(x, y).* Thus, every nonzero edge pixel coordinate *x* and *y* are grouped according to the equation (2). The purpose is to find the number of edge pixels satisfying the same *m* and *c*. Figure 3 illustrates the relation between image space and parameter space.



a) b)

Figure 3. Illustrations of a) image space b) parameter space

3.2 The Fast Hough Transform:

The HT is mapping from an image space into parameter space. One nonzero point is mapped into many points in parameter space. Thus, parameter space contains many mapped points. In HT formulation, time complexity is very high because it requires many multiplications. Therefore, HT cannot be suitable for high resolution and/or high dimensionality images.

Li *et. al.* wanted to decrease the time complexity and storage capacity of HT implementation so they have developed a new approach which is called Fast Hough Transform (FHT) [17]. In FHT, solution points in the *(m, c)* pairs are found by using clusters of them. In order to achieve that the parameter space is recursively divided into four main quadrants. Then, the distance between the center of the quadrant and the lines is measured. If the distance is greater than a threshold value, which is a large enough number of lines cross a given quadrant, the quadrant is divided again and the same procedure is applied until the quadrant size reaches 1x1. When quadrant size is equal to 1x1, *(m, c)* value gives the solution.

However, this approach has some practical problems such as perpendicular line detection, erroneous solution etc. These problems are eliminated in [18] with the following improvement in the algorithm

i. 2Nx2N (-N to N) parameter plane is generated for a NxN image and parameter plane is divided into four quadrants
ii. Each quadrant has center coordinate values (C_m and C_c)
iii. Distance d_i are measured between every nonzero (x_i, y_i) pairs and C_m , C_c, with the following equations

$$d_i = A_{2i} + A_{1i} \cdot C_m + A_{0i} \cdot C_c \quad (3)$$

$$A_{0i} = \frac{1}{\sqrt{1+x_i^2}}, A_{1i} = x_i \cdot A_{0i},\ A_{2i} = -y_i \cdot A_{0i} \quad (4)$$

iv. To minimize the number of calculations, a presence vector is defined. This vector indicates whether a line of the parameter space crosses a given quadrant. It consists of four elements. Each of elements represent a quadrant vote

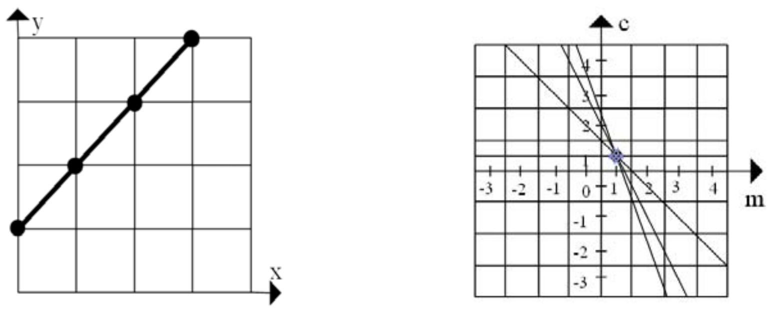
v. If the distance is above the threshold, the presence vector element is increased by 1

vi. The given quadrant is divided again and the same procedure is applied if large enough number of lines crosses it (i. e. presence vectors corresponding element vote is greater than the threshold).

vii. The recursive application of this process continues until quadrant size is 1x1.

viii. If the quadrant size is equal to 1x1, (C_m , C_c) is a solution of the *(m, c)* pairs.

Figure 4 illustrates the image and parameter spaces. Image space represents the binary image. It is seen that *(m, c)* pair takes part in the first quadrant and this quadrant is divided again.



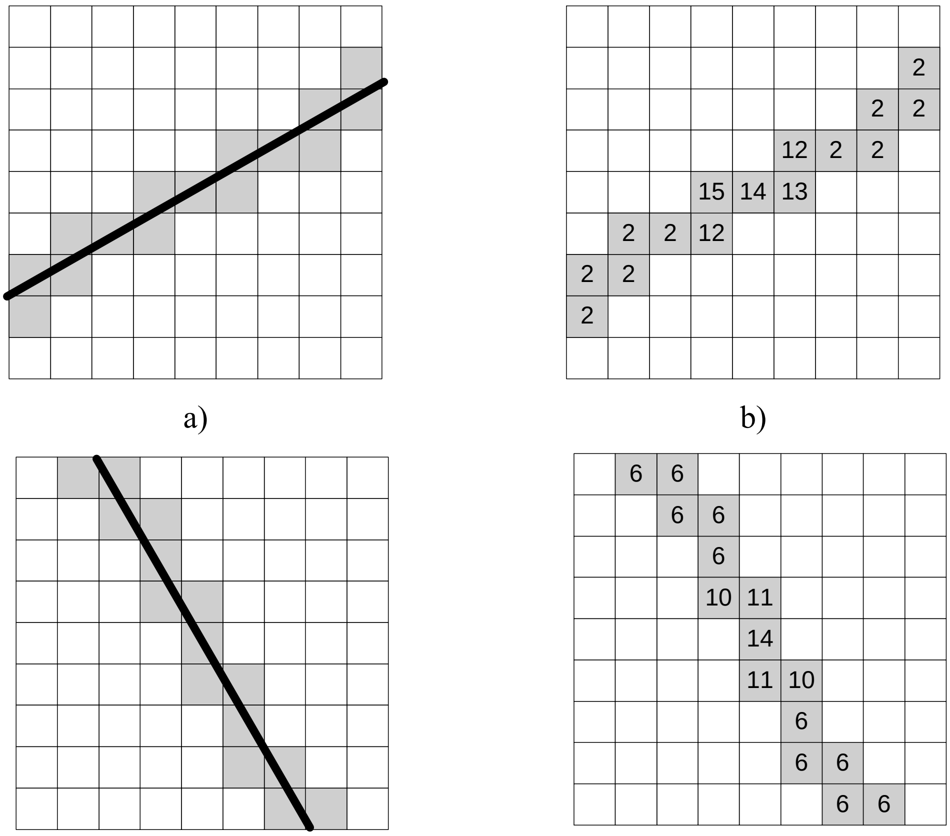
a) b)

Figure 4. Illustrations of a) image space b) parameter space

3.3 The Fast Hough Transform Approximation

The Fast Hough Transform Approximation (FHTA) has been developed by us with the intention to decrease the time complexity of the HT/FHT.

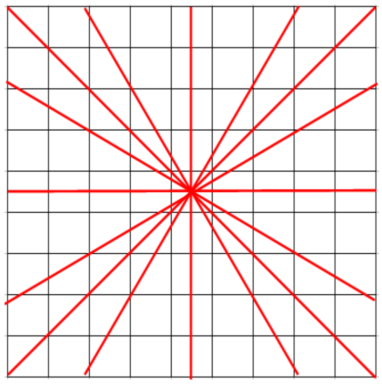
The FHT basically depends on slope, *m*, and intersection of y-axis, *c*. Note that the purpose of the HT/FHT is to find the *(m, c)* pairs of the lines. However, in FHTA we seek only known slope values which are denoted by 8 degrees or orientations (0°, 30°, 45°, 60°, 90°, 120°, 135° and 150°). Each slope or orientation line is converted into digital form by using label values. A specific collection of those label values denote the pixel locations through which a line passes. As an example Figure 5(a) presents the pixels through which 30° line passes. Those pixels are labeled as {2, 12, 13, and 14} as in Figure 5(b). Similarly Figure 5(c) presents the pixel through which 120° line passes. Those pixels are labeled as {6, 10, 11 and 14} as in Figure 5(d).



a) b) c) d)

Figure 5. a) 30° line b) 30° label values c) 120° line d) 120° label values

The remaining six lines or orientations are defined by label values similarly and they are combined to form a 9x9 mask as in Figure 6. This procedure is presented in Appendix A. The mask will be used for defining 8 orientations of the lines in the image. Table 1 shows how 8 orientations and their label values are formed.



a)

7	6	6	5	5	5	4	4	3
8	7	6	6	5	4	4	3	2
8	8	7	6	5	4	3	2	2
1	8	8	10	11	12	2	2	1
1	1	1	15	14	13	1	1	1
1	2	2	12	11	10	8	8	1
2	2	3	4	5	6	7	8	8
2	3	4	4	5	6	6	7	8
3	4	4	5	5	5	6	6	7

b)

Figure 6. 9 x 9 digital mask a) 8 lines b) 8 lines label values

Table 1. Eight orientations and corresponding label values.

Degree	Label values
0	$1 \vee 13 \vee 14 \vee 15$
30	$2 \vee 12 \vee 13 \vee 14 \vee 15$
45	$3 \vee 12 \vee 14$
60	$4 \vee 11 \vee 12 \vee 14$
90	$5 \vee 11 \vee 14$
120	$6 \vee 10 \vee 11 \vee 14$
135	$7 \vee 10 \vee 14$
150	$8 \vee 10 \vee 13 \vee 14 \vee 15$

Binary input image is processed with the mask to obtain eight different binary images. Each of them represents lines captured at one of 8 orientations.

3.4 Basis Splines (B-Spline)

A spline function consists of polynomial pieces on subintervals joined together with certain continuity conditions [19]. Spline functions are used for curve fitting applications. Formally, suppose that n + *1* points $t_o, t_1, \ldots, t_n$ have been specified and they satisfy $t_o < t_1 < \ldots\ldots < t_n$. These points are called knots or breakpoints. A spline function of degree k, which is greater than zero, having knots $t_o < t_1 < \ldots < t_n$ is a function S such that

i. *On each interval* $[t_{i-1}, t_i)$, *S is polynomial of degree* $\leq k$

ii. *S has a continuous (k-1)st derivative on* $[t_0, t_n]$

Therefore, the main advantage of the piecewise polynomials is that large number of data points can be fit with the low degree polynomial. Similarly, Basis Splines (B-Splines) form a basis for spline functions in a given degree. Use of B-Splines increases the efficiency and stability of the spline functions. Spline basis function is recursively defined by [20]

$$B_i^k(x) = \frac{t_{i+k+1} - x}{t_{i+k+1} - t_{i+1}} B_{i+1}^{k-1}(x) + \frac{x - t_i}{t_{i+k} - t_i} B_i^{k-1}(x) \qquad (k \geq 1) \tag{5}$$

Basis form of Spline functions *S* becomes

$$S(x) = \sum_i \theta_i B_i^k(x) \quad , \qquad \theta \text{ is real} \tag{6}$$

3.4.1. Cardinal (Cubic) Splines

Spline Functions with equally spaced knots have been called Cardinal Splines [19]. They have been the most popular Splines used in approximation theory. The main advantage of the Cardinal Splines is that there is only one cardinal spline in a given degree. The basis Cardinal Splines function of order $k = 3$ and the knot sequence $t = (-2, -1, 0, 1, 2)$ is given by the equation

$$B^3(x) = \begin{cases} 3|x|^3 - 6|x|^2 + 4, & 1 \leq |x| \leq 2, \\ -|x|^3 + 6|x|^2 - 12|x| + 8, & 1 \leq |x| \leq 2, \\ 0, & 2 \leq |x|, \end{cases} \tag{7}$$

Any cardinal spline function can be written as

$$S(x) = \sum_{=-\infty}^{\infty} \theta_i B_i^k(x - i) \tag{8}$$

CHAPTER 4

BARCODE LOCALIZATION VIA SEGMENTATION

Assume that a given image contains a barcode but how to locate the barcode area? This is a barcode localization problem. The barcode localization problem can be considered as a segmentation problem where a particular area of interest has to be extracted. This segmentation can be done on the edge detected image by usage of the line fitting techniques.

Mainly, barcode consists of dark bars and light spaces. Therefore, barcode are characterized by presence of straight lines and straight lines report the barcode area of the barcode image. In order to exploit this observation FHT and FHTA are used for the barcode localization. An input image must be binary so pre-processing is performed on the input image prior to barcode localization. The pre-processing consists of conversion of color image into gray scale image and edge detection. In this work, we used canny edge detection [20]. Barcode Localization procedure is presented in Figure 7 as a block diagram.

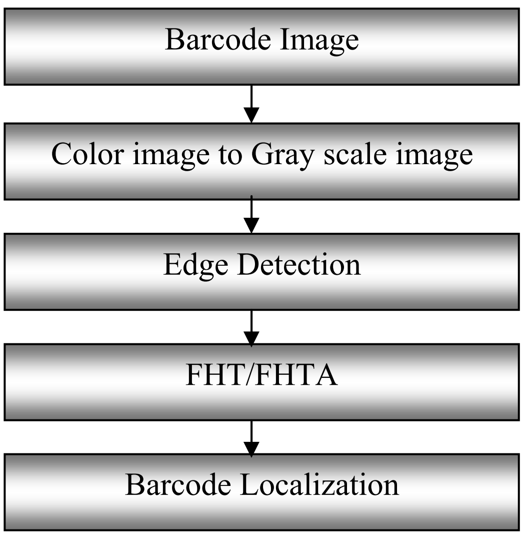


Figure 7. Barcode localization procedures

In this Chapter, our solution to barcode localization problem is examined and tested to evaluate its performance. Following the barcode localization methods in Chapter 3, the FHT and FHTA are chosen to obtain barcode localization and they are tested on test images. Some test images are shown in Figure 8 and Figure 9. The algorithms are implemented and executed on INTEL E6400 computer with 1024 MB DDR Ram.

In the evaluation of the experimental results, computational time of the barcode localization is used. Comparison between the localization via FHT and FHTA is presented in Section 4.3.

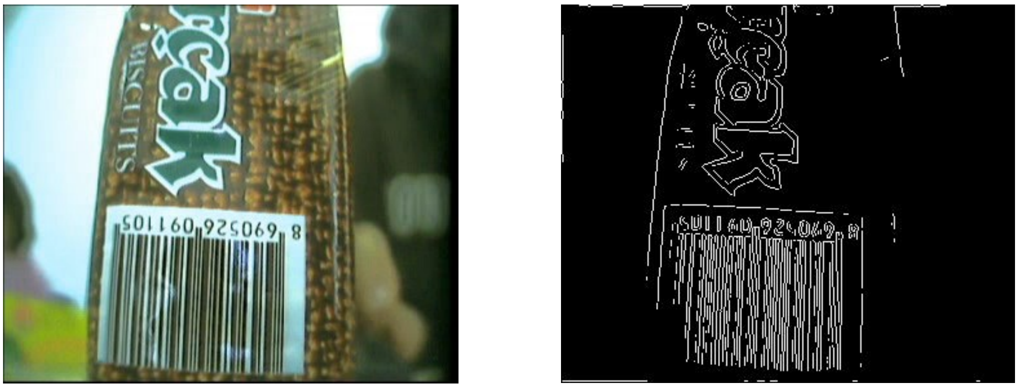


Figure 8. a) 282x358 sized test image 1 b) edge map

Figure 9. a) 1224x1632 sized test image 2 b) edge map

4.1. Barcode Localization via FHT

The basic idea of the FHT is to find the slope and intersection value *(m, c)* of lines. For this purpose parameter space is divided into four main quadrants and the distances between lines to quadrants center coordinate are measured. The flowchart of the FHT algorithm is presented in Figure 10. Algorithm of the barcode localization via FHT is listed as a pseudo code in detail.

main_module(barcode_image)
{
load_edge_detected_barcode_image(barcode_image);
calculate_fast_hough_transform_coefficent();
divide_hough_plane (binary_image, coefficient, $C_{m,}$ C_c);
find_barcode_orientation_and_rotate();
find_barcode_area();
}

main_module is the main function of the localization algorithm. Input argument of the function is barcode image and output is the barcode area. The loader must load the input image.

Firstly main module calls *load_edge_detected_barcode_image* function. The function takes on pre-processing. That is, prior to the FHT, binary image is obtained by using edge detection methods. If the input image is not gray scale, it is first converted into gray scale format.

Secondly, *calculate_fast_hough_transform_coefficent* calculates the FHT coefficients for nonzero pixel coordinate pairs. These values are used in distance measurements in FHT.

Thirdly, *divide_hough_plane* function is the most important part of the main function and pseudo code algorithm for it is as follows;

```
divide_hough_plane (binary_image, coefficient, Cm, Cc)
{
if  binary_image_size ==1
    send_line_coordinate();
else
    measure_distance();
      if distance_satisfy_desired_value
          divide_slope_orgin_and_intersection_orgin
          divide_binary_image_size
          divide_hough_plane (binary_image, coefficient, Cm, Cc);
      end
end
}
```

This function divides the parameter space into four quadrants. Each quadrant has some slope coordinate C_m and intersection of y-axis coordinate C_c. Initial value of the C_m and C_c are both zero. *divide_hough_plane function* calls itself in order. If distance between lines and (C_m ,C_c) stays below threshold value, this quadrant is divided again recursively and (C_m ,C_c) pairs are changed depending on the size. This procedure keeps until image size is equal to 1x1. If image size equals to 1x1 the function returns the line coordinates.

Fourthly, *find_barcode_orientation_and_rotate* function finds the orientation of the barcode and the image is rotated as the barcode is in 90 degrees orientation by using the orientation information. Because *find_border* function only works when barcode lie on the 90 degree.

Finally, *find_border* function applies the binary morphology (erosion and dilation) in order to eliminate the noisy points. After that, profiling of the row and column respectively returns the barcode area.

Profiling is the process of converting a two-dimensioned data into a one-dimensioned curve by projecting it vertically or horizontally. The one dimensional curve obtained as a result of the projection, is formed of clusters with different amplitudes. By applying thresholding to the formed clusters, the separate clusters are cleaned and thus the start and end coordinates of the barcode are obtained.

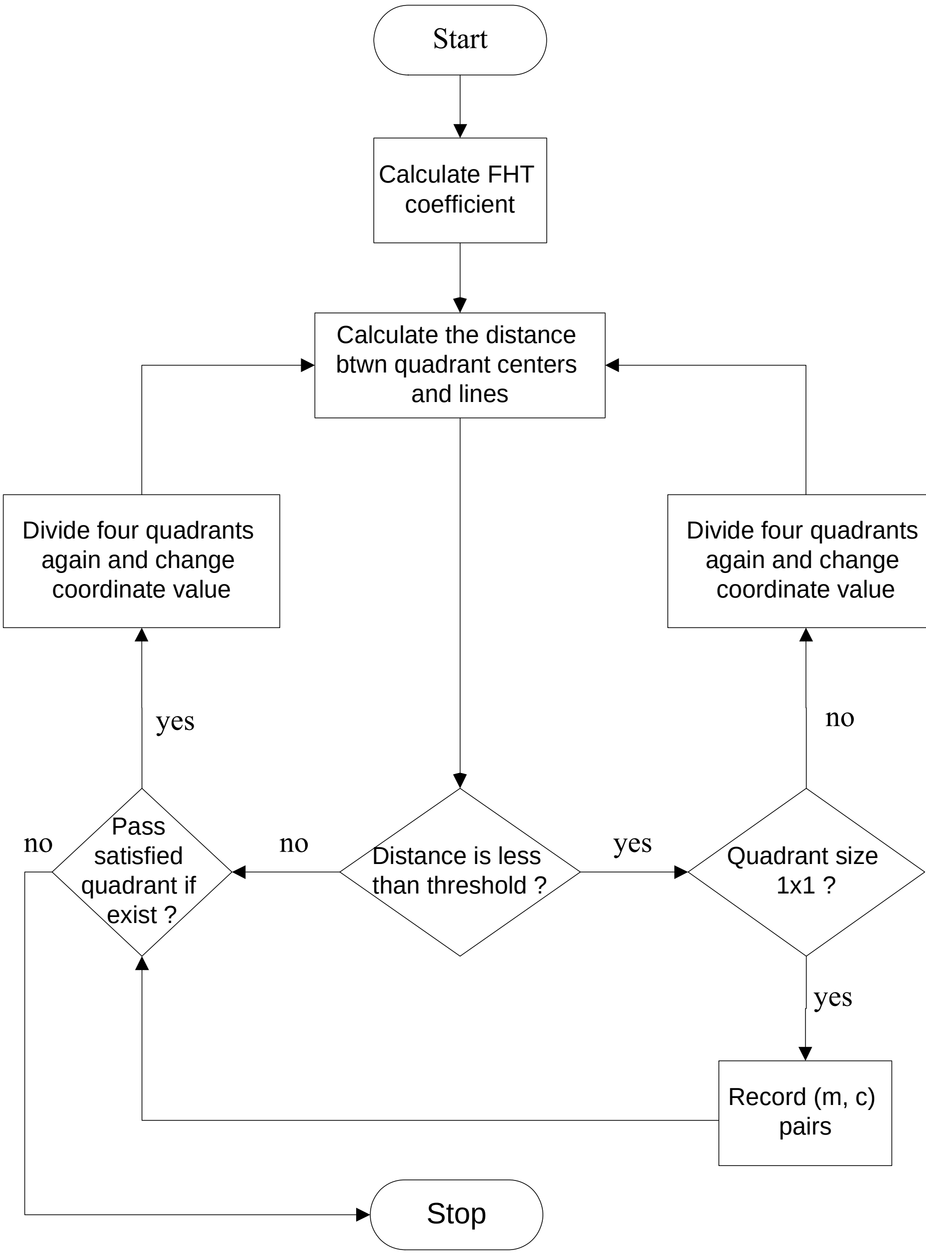


Figure 10. The flow chart of FHT algorithm

The test images of Figures 8 and 9 are utilized to assess the performance of barcode localization via FHT. Before the simulation, the image dimension must be changed to be the multiples of two. Otherwise, algorithm never reaches 1x1 image size. During the simulation, threshold value of the distance is expressed in terms of quadrant size n as $0.6{\cdot}n$. Barcode localization, binary morphologies and projections results are displayed in Figures 11 and 12.

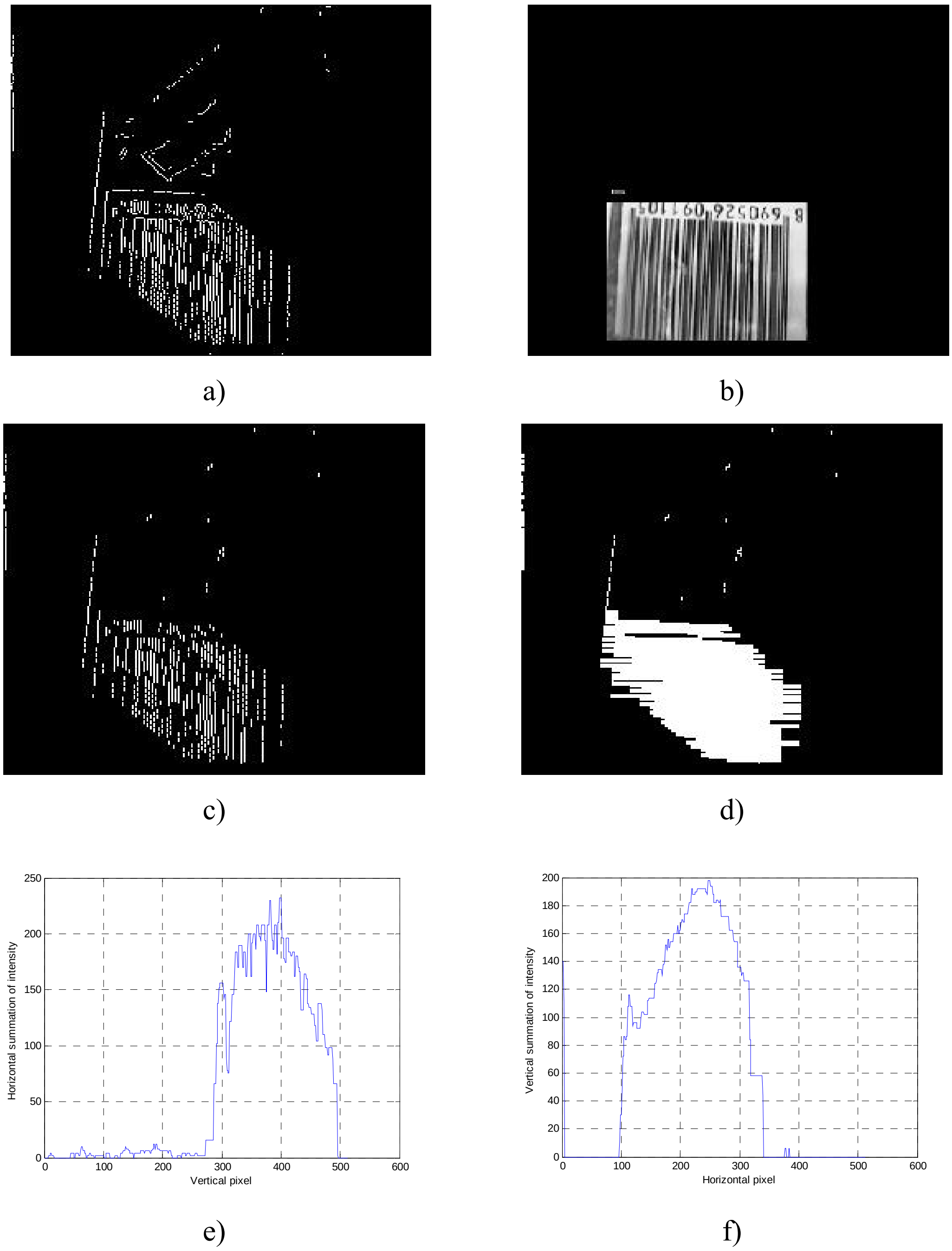


Figure 11. Test image 1 a) FHT result b) barcode localization c) binary morphology opening d) closing e) horizontal projection f) vertical projection

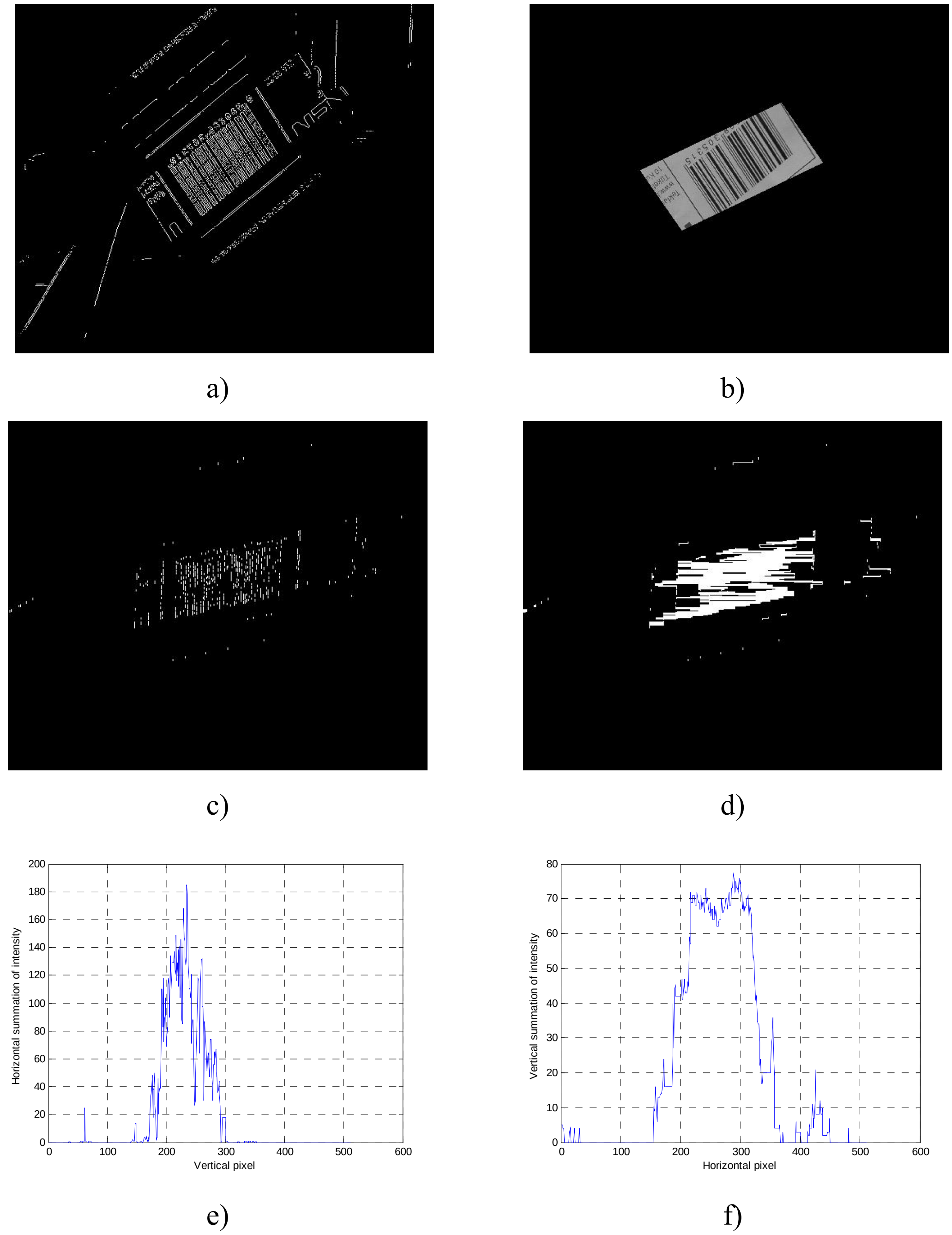


Figure 12. Test image 2 a) FHT result b) barcode localization c) binary morphology opening d) closing e) horizontal projection f) vertical projection

Barcode area is determined via binary morphology and then with the methods of horizontal and vertical profiling.

Dilation and erosion are two basic morphological operations. Dilation is gradually enlarge the the boundiries of regions of foreground pixels grow in size while holes within those regions become smaller. Erosion shrinks areas of the foreground pixels and holes within those areas become larger [21]. Dilation and erosion is generally used for bridging gaps and eliminating irrelevant detail, respectively. A number of pixels are added or removed from the image which depends on kernel size and orientation. If these operations are combined with different orders, it is called opening or closing. The opening is erosion followed by the dilation and the closing is dilation followed by the erosion with the same kernel. In this work opening and closing was used to get rid of the isolated pixels. Horizontal orientation and 5 unit size kernel was used for opening. Vertical orientation and 20 unit size kernel was used for closing. Results of opening and closing are presented in Figure 11(c-d) and Figure 12(c-d).

The horizontal and vertical projection results for test images of Figure 8 and Figure 9 are presented in Figures 11(e-f) and 12(e-f). Barcode areas possess themselves uninterrupted high amplitude region in profiling curves because of intense, close lines. The results of the computation time are presented in Table 2.

Table 2. Execution time of FHT

IMAGE	Elapsed Time(Sec)
Test Image 1	583.34
Test Image 2	390.69

4.2 Barcode Localization via FHTA

Remember that the purpose of FHTA is to locate the region of lines on the image. A 9x9 sliding mask is used for this purpose. Binary morphology and profiling explained in Section 4.1 are applied to get the barcode region. Algorithm of the barcode localization via FHTA is listed as a pseudo code in detail. The flowchart of the FHT algorithm is presented in Figure 13.

```
Main module (input_image)
{
Load_edge_detected_image (input_image);
find_lines (binary_image);
find_barcode_area (barcode_lines);
}
```

main_module is the main function of the localization algorithm. Input argument of the function is a barcode image and output is the barcode area coordinates.

Firstly, main module calls *load_edge_detected_image* function. This function performs pre-processing by converting a color image into gray scale one, and then by generating binary image of edges.

Secondly, find_*lines* function uses 9x9 mask of Figure 6 and the edge map. The mask is slided on the edge map. For every nonzero edge pixel enclosed by the mask, shifting by the corresponding label values is performed. Hence, at most 15 bit planes (the maximum label value is 15) are created for each masking operation. Next, bit planes are combined by logical OR operation according to Table 1 to form 8 bit planes each corresponding to one of eight orientations. If the number of non-zero pixels of a bit plane exceeds a threshold, the central pixel is said to be a point on a line with the orientation of the corresponding bit plane. Note that, an edge pixel may be a point on several lines with different orientations. This procedure is repeated until the mask is shifted on the whole image. As a result, eight binary images are obtained. Each of them contains lines which are at one of (0°, 30°, 45°, 60°, 90°, 120°, 135° and 150°) orientations. In the end, the function compares the eight binary images. The one which includes more number of lines is set as the output image.

Finally, *find_border* function applies binary morphology (opening and closing) in order to eliminate noise and isolated pixels. Barcode area coordinates are extracted by vertical and horizontal profilling. If the solution set is not on 90 degrees, it is rotated to 90 degrees depending on the orientation information.

The test images of Figures 8 and 9 are utilized to assess the performance of barcode localization via FHTA. Before the simulation, test image is resized into 512x512 with the intention to compare the test results by FHT. Barcode localization, binary morphologies and projection results are displayed in Figures 14 and 15.

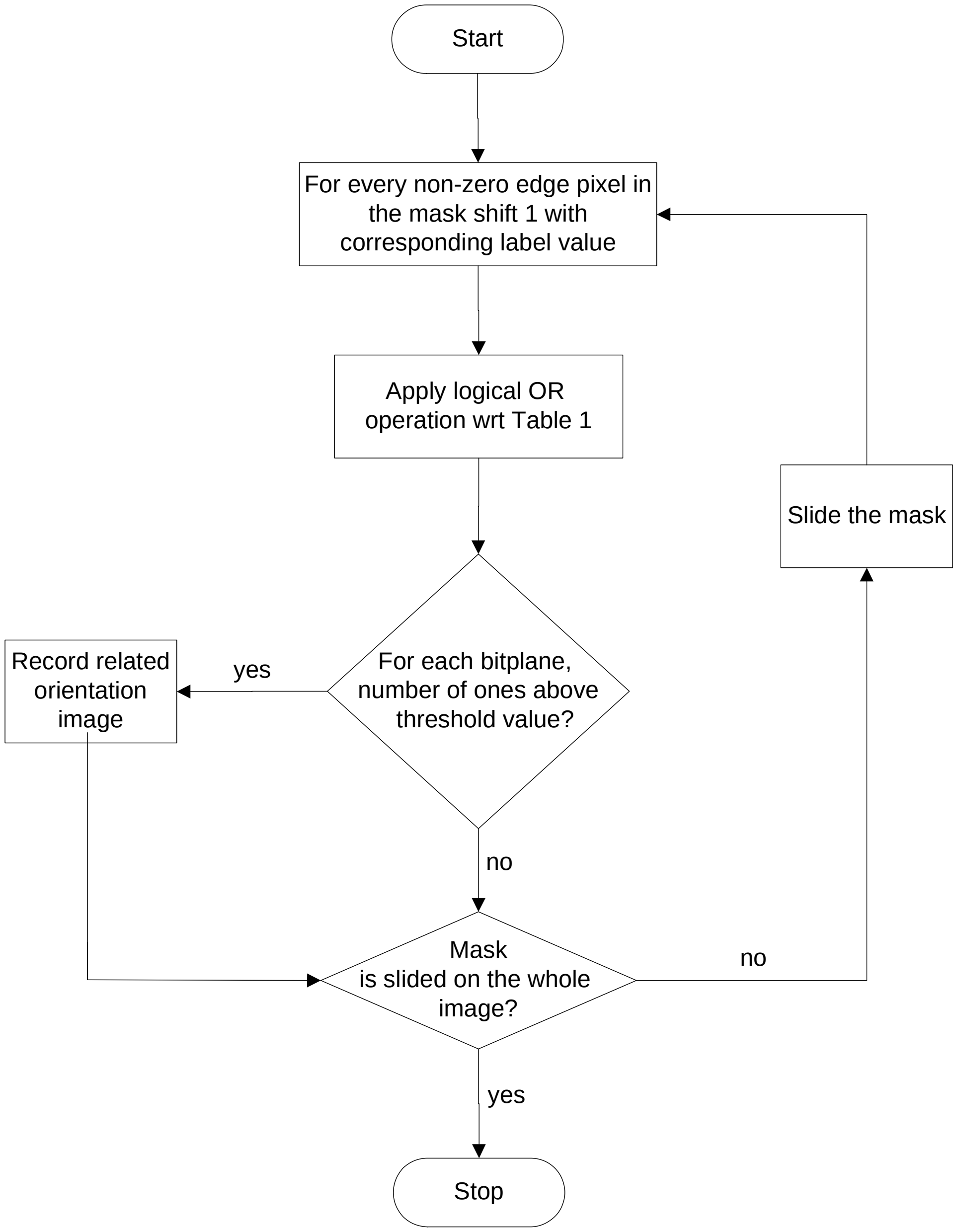


Figure 13. The flow chart of FHTA algorithm

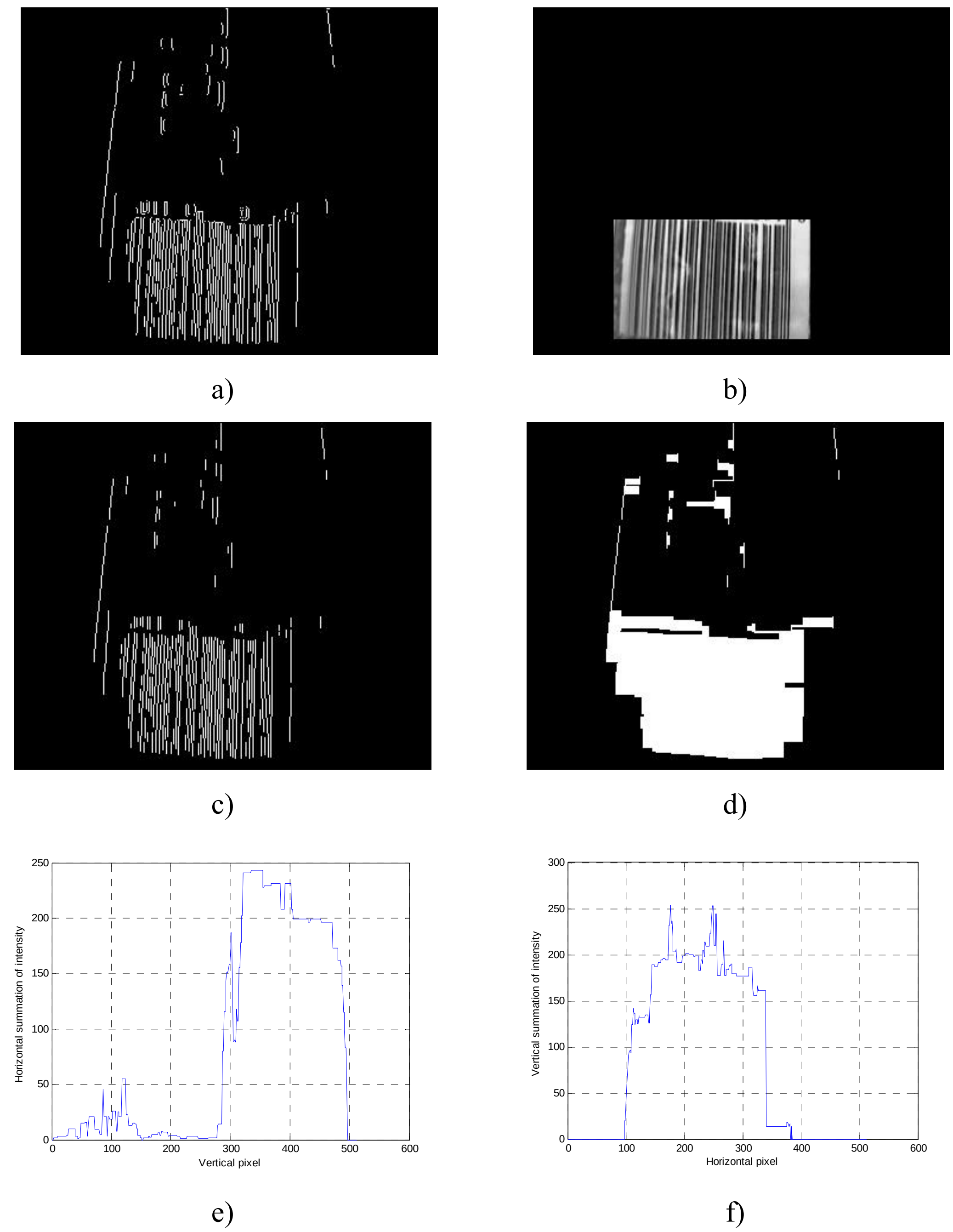


Figure 14. Test image 1. a) FHTA result b) barcode localization c) binary morphology opening d) closing e) horizontal projection f) vertical projection

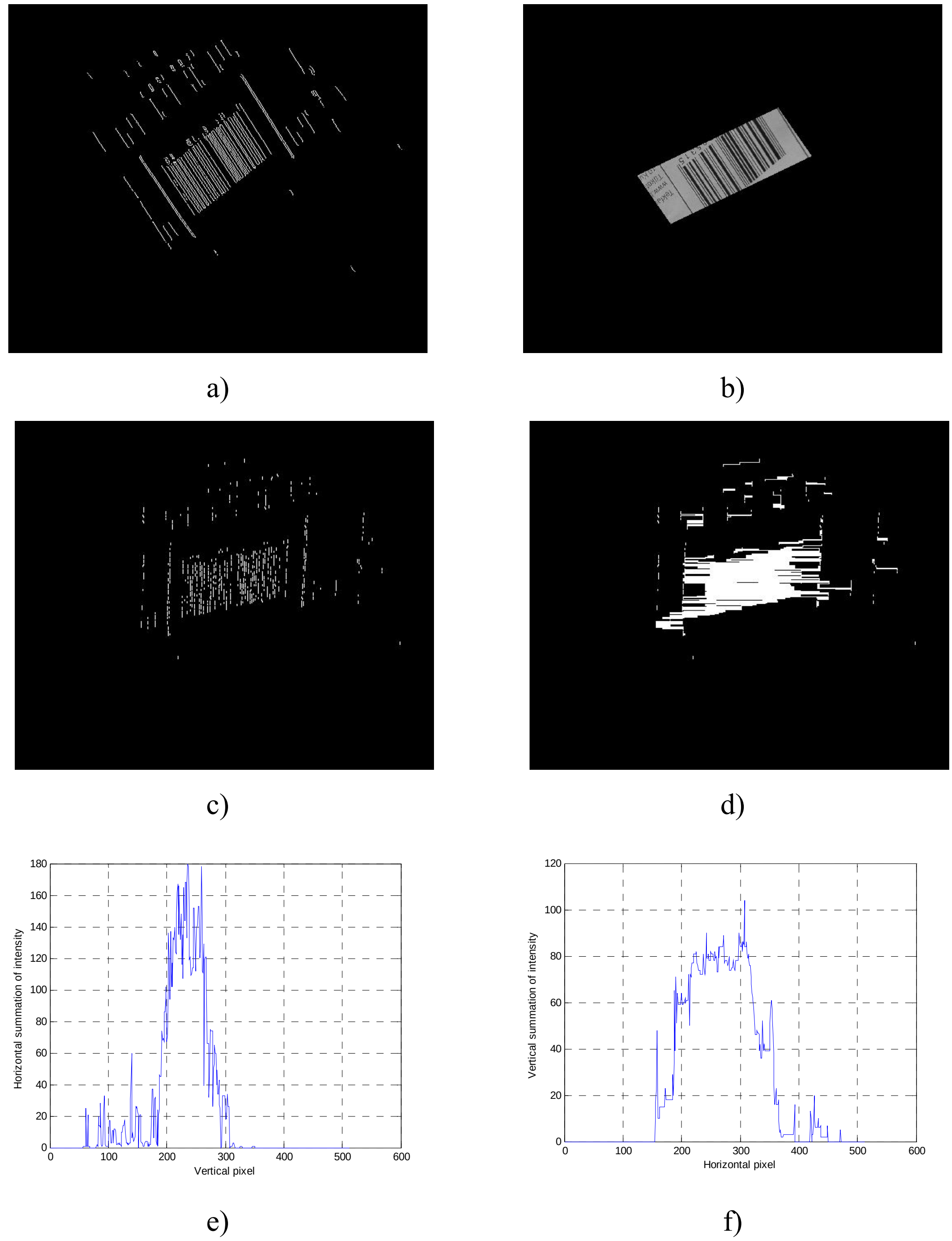


Figure 15. Test image 2. a) FHTA result b) barcode localization c) binary morphology opening d) closing e) horizontal projection f) vertical projection

As in the FHT tests, the coordinate values of the barcode area are detected by binary morphology followed by profiling. For profiling and binary morphology, the same kernel and threshold values that we have used in FHT are used. Execution times of localization are presented in Table 3.

Table 3. Execution time of FHTA

IMAGE	Elapsed Time(Sec)
Test Image 1	17.142
Test Image 2	17.195

4.3 Experimental Result and Comparison between FHT and FHTA

Our experiments on several images which include barcodes at different orientations show the efficiency of the method we have introduced. Images are captured with two different cameras which are Sony-Ericsson K750i cellular phone and wireless webcam. Figures 8 and 9, display examples of the test set. For each of the two test images, barcode localization is achieved by FHT/FHTA followed by profiling and binary morphology.

FHT is implemented both according to the method proposed in [18] and to our proposed approximation. When the two methods are compared in terms of execution time, FHT that has been introduced by Li *et. al.* [17] and also Zapate *et. al.* [18] is as follows,

- $O(P \cdot \log_2 N)$ multiplication
- $O(P \cdot \log_2 N)$ addition
- $O(P)$ thresholding

for P edge points and N being image size. On the other hand, complexity of FHTA includes;

- $O(P \cdot k_i)$bit shift,
- $O(P)$ logical OR
- $O(P)$ addition and
- $O(P)$ thresholding

for k_i is number of the ith window edge points.

Tables 2 and 3 show that the execution time of FHT is much higher than of FHTA when the execution times of the two test images in Figures 8 and 9 are compared.

This is also confirmed by the order of process analysis above, where FHTA does not require any multiplications. To sum up, the efficiency of the two methods are equivalent but the execution time of the method we have introduced is much shorter and it detects not only the barcode but also the orientation of the barcode.

CHAPTER 5

BARCODE DECODING

5.1 Barcode Fundamentals

Data embedded in barcodes are coded by a coding standard. There exist many different one-dimensioned coding standards such as EAN13, EAN8, EAN128, UPC-A, UPC-E. However, EAN13 barcode standard is the most common one in Turkey and throughout Europe. EAN13 standard consists of 13 digits. At the end and the beginning of the barcode is the start and finish codes which consist of two black lines/dark bars and a white line/light space and in the middle is the center code [24]. All these codes are classified as guard bands (Figure 16). Moreover, the last digit on the barcode is the control code or parity code of the barcode.

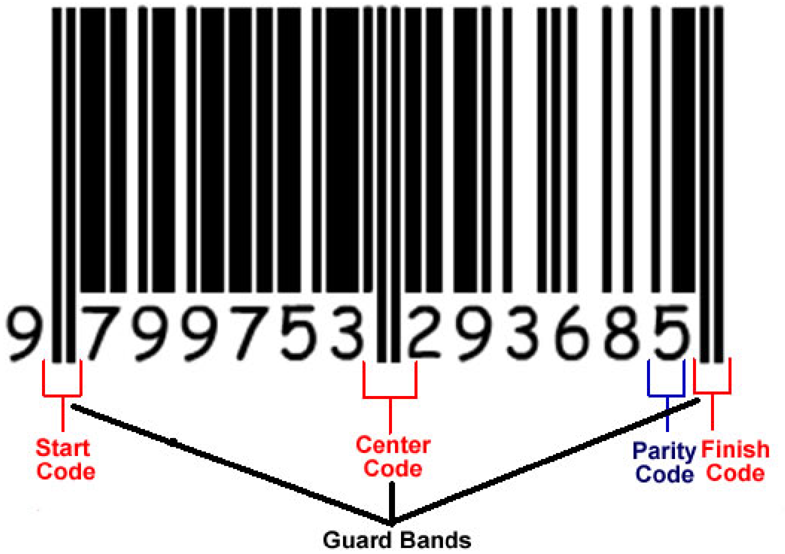


Figure 16. EAN13 barcode standard

The thickness of the dark bars and the light spaces in comparison with each other determines the digit they represent. The thinnest bar/space represents one unit whereas the thickest bar/space represents four units. Every digit except for the start digit consists of two dark bars and two light spaces (seven units in total) and they are expressed according to their line thickness in figure Figure 17.

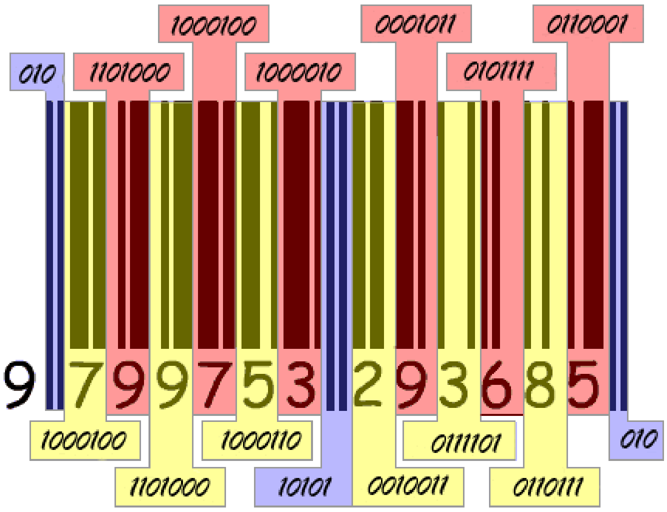


Figure 17. The representation of the barcode data through binary number system
1 represents light spaces, 0 represents dark bars

The center code is the reference point when decoding the barcode. The left part of the center code is classified as the left side whereas the other part is classified as the right side. The start digit holds the reference point and then the six digits on the left side are arranged as odd or even, according to Table 4. For instance, in Figure 16 the six digits are arranged as Odd-Even-Even-Odd-Even-Odd since the start digit is 9. After the line up of the left side is found out using Table 4, the left side is decoded regarding whether the digit to be decoded is odd or even, using Table 5. There is no odd-even line up for the right side. Hence only Table 5 is used for the right side decoding. After the barcode is decoded the accuracy of the transcribed data could be checked through these steps:

1. The barcode which consists of 13 digits except for the control code is arranged as Even-Odd-Even-Odd (The first digit should be Even).

2. The sum of the odd numbers in the line up is calculated and the total is multiplied with 3.
3. The sum of the even numbers is calculated.
4. The sums found in step 2 and 3 and the control code are added.
5. The result should be divisible by 10.

Table 4. The line up of the left side according to the start digit.

FIRST DIGIT	**SECOND DIGIT**	**LEFT SIDE**				
		1. DIGIT	**2. DIGIT**	**3. DIGIT**	**4. DIGIT**	**5. DIGIT**
0	Odd	Odd	Odd	Odd	Odd	Odd
1	Odd	Odd	Even	Odd	Even	Even
2	Odd	Odd	Even	Even	Even	Even
3	Odd	Odd	Even	Even	Even	Odd
4	Odd	Even	Even	Odd	Even	Even
5	Odd	Even	Even	Odd	Even	Even
6	Odd	Even	Even	Even	Odd	Odd
7	Odd	Even	Odd	Even	Odd	Even
8	Odd	Even	Odd	Even	Even	Odd
9	Odd	Even	Even	Odd	Even	Odd

Table 5. The representation of digit values through binary number system.

DIGIT	**LEFT SIDE**		**RIGHT SIDE**
	ODD	**EVEN**	
0	1110010	0100111	0001101
1	1100110	0110011	0011001
2	1101100	0011011	0010011
3	1000010	0100001	0111101
4	1011100	0011101	0100011
5	1001110	0111001	0110001
6	1010000	0000101	0101111
7	1000100	0010001	0111011
8	0110111	0001001	0110111
9	0001011	0010111	0001011

5.2 Problem Definition

The decoding process requires the widths of the barcode lines to be known. In order to do this, the midpoints of the dark bars and light spaces could be used since the images at hand may be noisy or low quality. However, quite often barcode area obtained via barcode localization may not include only the barcode itself. There may be some areas which are not actually a part of the barcode or that area may not completely enclose the barcode. The extracted barcode area must be checked if start and finish are included. First, a scan line is chosen from the region that has been detected as the barcode area as in Figure18. This scan line is used to find the start and finish coordinates of the barcode and so a new scan line is formed which includes the data of the barcode only and excludes unneeded information.

a)

b)

c)

Figure 18. a) Barcode image b) barcode localization c) 10 scan lines extracted from the detected region

On the scan line which includes only the barcode itself could be found the midpoints of the barcode bars and spaces. However, enhancing the image quality of the scan line (B-Spline Smoothing) will improve the system performance considering the fact that image quality is critically important in the process of decoding. In order to be able to handle imperfect (low resolution, noisy etc.) barcode image samples, decoding process performs the following steps; B-Spline smoothing, detection of bar and space intervals, super-resolution enhancement, and finally matching through the correlation.

5.3 Decoding Technique

5.3.1 Image Approximation with B-Spline Smoothing

B-spline fitting is a kind of interpolation technique and is used for obtaining continuous approximations of discrete data. However, interpolation is not robust enough in presence of noise in observed data. Thus, B-spline smoothing is preferred in the existence of noise to enhance fitting. B-spline smoothing based estimation is introduced by Reinsch in [25] as follows;

Consider a set of sample values $\mathbf{z} = (z_1, z_2, z_3, ...z_4)^T$, corresponding to data at grid x_i. Assume that the samples are corrupted with noise such that

$$z_i = f(x_i) + v_i \tag{9}$$

where v_i represents noise and $f(x)$ can be supposed as a smooth function. For normally distributed measurement error, the natural measure of fit to data is the weighted residual sum of squares (RSS)

$$RSS(f) \stackrel{\Delta}{=} \sum_{i=1} w_i (z_i - f(x_i))^2 \tag{10}$$

where w_i's are weights associated with the i'th sample

Spline Smoothing measures the lack of smoothness based on the following Equation

$$R_2(f) = \int (\frac{\partial^2 f}{\partial x^2})^2 dx \tag{11}$$

To recover $f(x)$ from z_i, a $\hat{f}(x)$ can be constructed by minimizing

$$\sum_{i=1}^{N} w_i (z_i - f(x_i))^2 + \lambda \int (\frac{\partial^l f}{\partial x^l})^2 dx \tag{12}$$

where λ is the smoothing parameter. It controls the trade of between fit to data and smoothness.

To reformulate (12) in matrix form, let $\mathbf{z}$ be the observation vector and $\mathbf{f}$ be the estimation vector. The matrix of cardinal basis function evaluated at grid points ($i = 1,.....,N$) can be written as

$$B(x) = \begin{bmatrix} B(x_0 - 0) & B(x_0 - 1) & \cdots & \cdots & B(x_0 - N) \\ B(x_1 - 0) & B(x_1 - 1) & \cdots & \cdots & B(x_1 - N) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ B(x_N - 0) & B(x_N - 1) & \cdots & \cdots & B(x_N - N) \end{bmatrix} \tag{13}$$

(12) can be rearranged in matrix notation as

$$(\mathbf{z} - \mathbf{f})^T (\mathbf{z} - \mathbf{f}) + \lambda (\mathbf{f}'')^T (\mathbf{f}'') \tag{14}$$

where $\mathbf{f} = \mathbf{B\theta}$, $\boldsymbol{\theta}$ is the vector of basis functions weights. Then Equation (14) becomes

$$(\mathbf{z} - \mathbf{B\theta})^T (\mathbf{z} - \mathbf{B\theta}) + \lambda (\mathbf{B}''\boldsymbol{\theta})^T (\mathbf{B}''\boldsymbol{\theta}) \tag{15}$$

Minimizing (15) with respect to $\boldsymbol{\theta}$,

$$\frac{\partial}{\partial \boldsymbol{\theta}} \left((\mathbf{z} - \mathbf{B\theta})^T (\mathbf{z} - \mathbf{B\theta}) + \lambda (\mathbf{B}''\boldsymbol{\theta})^T (\mathbf{B}''\boldsymbol{\theta}) \right) \tag{16}$$

$\hat{\boldsymbol{\theta}}$ becomes

$$\hat{\boldsymbol{\theta}} = (\mathbf{B}^T\mathbf{B} + \lambda\mathbf{B}''^{\mathbf{T}}\mathbf{B}'')^{-1}\mathbf{B}^T\mathbf{z} \tag{17}$$

Then,

$$\hat{\mathbf{f}} = \mathbf{B}(\mathbf{B}^T\mathbf{B} + \lambda\mathbf{B}''^T\mathbf{B}'')^{-1}\mathbf{B}^T\mathbf{z} \tag{18}$$

$\hat{\mathbf{f}}$ provides a smoothed approximation of $\mathbf{z}$ evaluated at x_i grid points.

5.3.2 Applying B-Spline Smoothing Method to Barcode Scan Line

B-spline smoothing (BSS) is applied to a scan line of the barcode by using cardinal basis. The noise of the scan line is eliminated with this process. BSS also provides a function representation to data, from which the derivative of data can easily be obtained through derivative of basis functions. Figure 19 presents a barcode scan line data, B-spline smoothed data output and the difference between the scan line and smoothed data. Smoothing parameters is 0.5 for this example. The smoothing parameter λ controls the tradeoff between fit to data and smoothness. For instance, when λ approaches zero we attain only cubic spline interpolation and we attain linear regression when it approaches infinity. We choose λ to be 0.5 empirically for our application after a set of experiments.

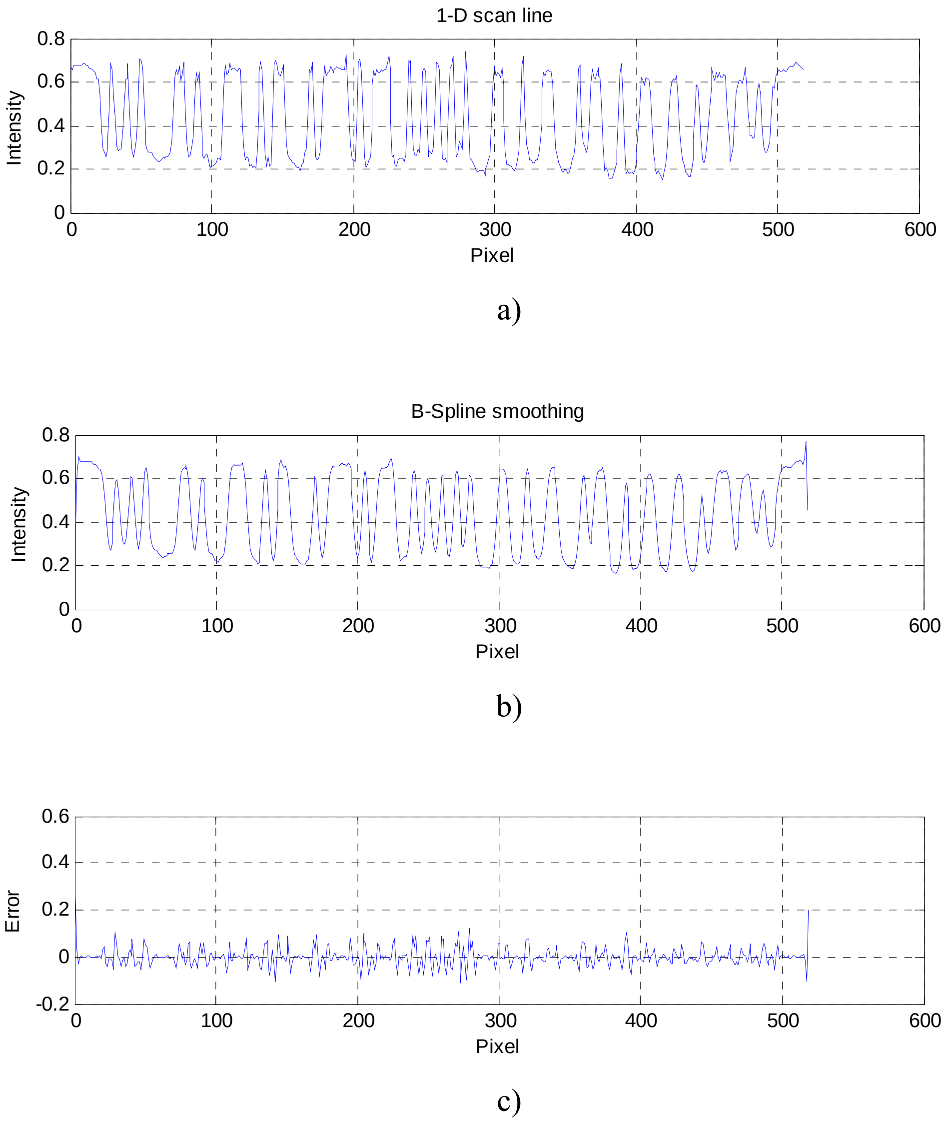
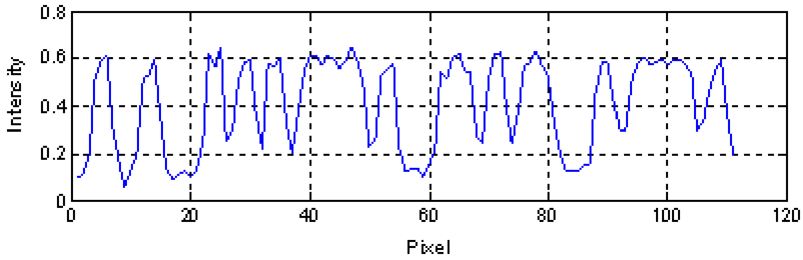


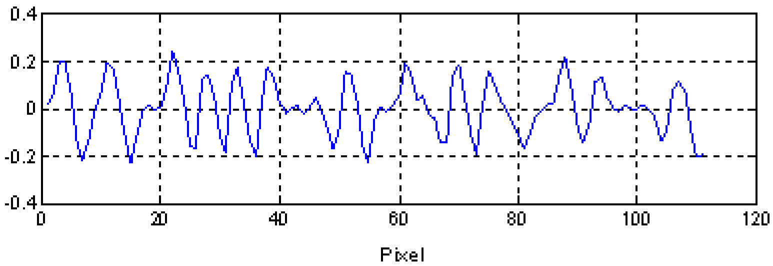
Figure 19. a) 1-D Scan line b) B-spline smoothing result (λ =0.5) c) error between the scan line and B-spline smoothed data

5.3.3 Estimation of Peak Locations

Since the barcode is made up of bars and spaces, the one-dimensional scan line we have achieved will constantly produce maxima and minima. The maxima are supposed to occur at the midpoints of the light spaces and minima are supposed to occur at the midpoints of the dark bars. Location of maxima and minima can be found as the zero-crossing points of the first derivative of the smoothed scan line. However, even with the smoothed data, existence of local variations degrades the performance as can be seen in Figure 20.



a)



b)

Figure 20. a) A scan line of the barcode b) the derivative of the scan line

When examining Figure 20(b), it is seen that the derivative of the barcode intersects zero at several points because of local variations instead of intersecting it at only one point corresponding to a bar/space. This results in inaccuracy in the calculation of midpoint values. In order to overcome this problem, a thresholding band is used. On the thresholding band, it is considered that there are two lines parallel to the axis x for –M, +M thresholding value. M value has to be bigger than local variations. In this way, the peak maximum and minimum values will intersect thresholding band at two points. For each maximum and minimum value, the values intersecting thresholding band are calculated. By using the calculated values, a line is drawn between the points that intersect M thresholding band. The starting point of the line we have drawn is the second value of the minimum intersection whereas the finish point is the initial value of the maximum intersection or the starting point is the second value of the maximum intersection whereas the finish point is the initial value of the minimum intersection. Then we detect the point where the line intersects zero. Hence, the midpoints of the bars and spaces on the barcode are found. This procedure is illustrated in Figure 21.

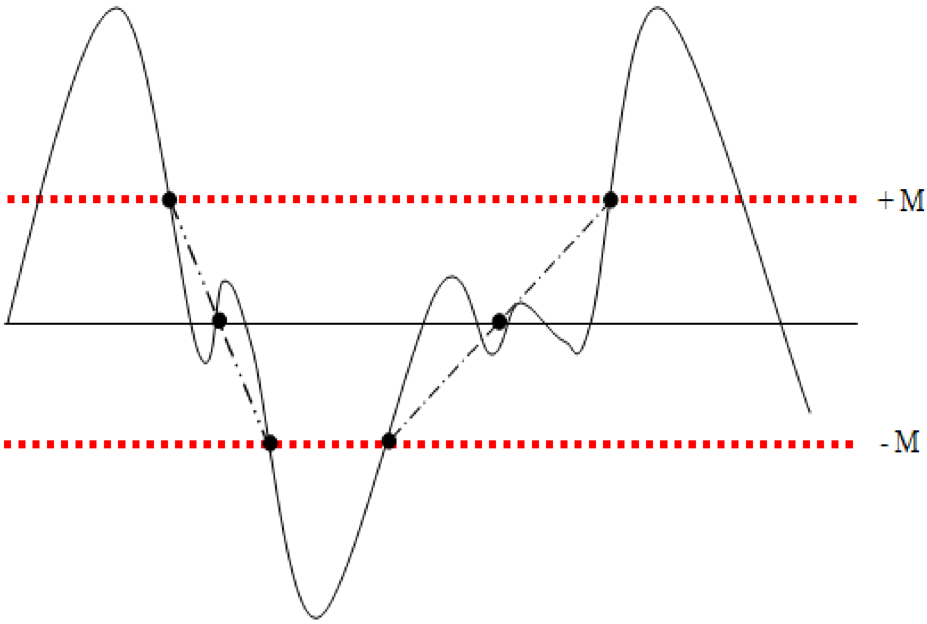


Figure 21. Thresholding Band

The selection of the M threshold value varies according to the image quality. Thus a fix threshold value is not used. By considering that the peak values of the barcode are fixed (total of the bars and spaces is 59), 0.045 is selected which provides the M value for the best image quality. If this M value can catch sufficient peaks, midpoints are calculated. If not, this value is decreased 0.001 until correct number of peak values is obtained.

5.3.4 Super Resolution Enhancement for the Barcode

Super resolution method is proposed for the reconstruction of a higher quality barcode sample from a low resolution/low quality sample. To obtain high quality barcode sample from low quality one, barcode geometric features and the imaging parameters are used [10]. The geometric features and imaging parameters consist of a vector containing the distance of start and end points of each bar from a reference point and the tilt angle. The start and end points identify the width of the bars. Note that using these parameters a barcode can be reconstructed at any resolution. The tilt angle and pixel size must be known for calculating how locations of bars are displaced from one scan line to another.

In this approach, barcode area is defined as a 2-D continuous function $f(x,y)$ which represents the intensity value of the barcode at location (x,y), i.e.,

$$f(x,y)=\begin{cases}1, & (x,y)\in B_k \\ 0, & otherwise\end{cases} \tag{19}$$

where B_k represent the region bounded by the kth bar. Another function $f_s(x,y)$ is defined as a discrete function acquired from $f(x,y)$ such that

$$f_s(x,y)=\int_{j}^{j+1}\int_{i}^{i+1} f(x,y)dxdy \tag{20}$$

where $(i,j)=(0,0),(0,1),\cdots,(1,0),\cdots,(N,M)$ as shown in Figure 22. $f_s(x,y)$ is the area of the region in pixel (i,j) bounded by the bar passing trough that pixel. It is assumed that each pixel is a square region (1unit x 1unit area). A feature vector **d**, with size *2n+1,* is defined. **d** is made up of odd and even indexed elements and a tilt angle. The odd indexed n elements denote the distance from the point $(0,0)$ to the point where the left boundary of B_k (kth) bar intersects the top of the pixel at $j=0$. Similarly the even index n elements is from the point $(0,0)$ to the point where the right boundary of B_k (kth) bar intersect the top of the pixel at $j=0$. Finally, the last element (*2n+1*th) is the tilt angle (α).

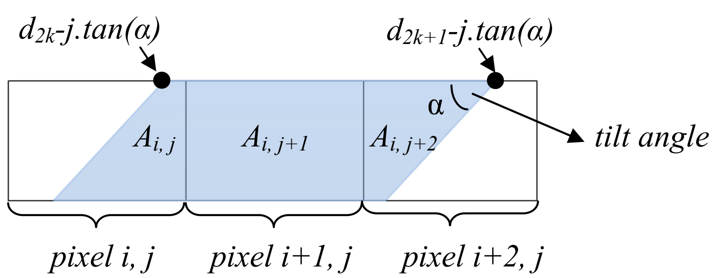


Figure 22. Construction of $f_s(x,y)$

For each scan line, boundaries of B_k are shifted by a factor of $\tan(\alpha)$, as shown in Figure 22 and boundaries of B_k can be calculated. Thus, missing information can be compensated by using other scan line information.

The mathematical model is described in [10]. Before the mathematical model some parameter definitions are listed below;

- $f_l(i,j)$ is an undersampled discrete version of $f(x,y)$,
- $\mathbf{d_0}$, $\mathbf{d_i}$ as the initial and *i*th iteration respectively,
- $\mathbf{f_s^{d_i}}$, $\mathbf{f_l}$ as the vectors whose elements are $f_s^{d_i}$ and f_l, respectively , at locations where f_l data available,
- $\mathbf{f_s^{d_i}}$ as the barcode image constructed from $\mathbf{d_i}$
- $\nabla\mathbf{f_s^{d_i}}$ as the vector whose elements represent the change in $\mathbf{f_s^{d_i}}$, when $\mathbf{d_i}$ is moved in positive x direction by $\boldsymbol{\delta}\mathbf{d}$,

The goal is to converge to a $\mathbf{d_i}$ which represents barcode lines' start and end points such that the norm of the error in Equation (21) is minimized

$$\|\mathbf{e}\| = \left\|\mathbf{f_l} - \mathbf{f_s^{d_i}}\right\| \tag{21}$$

$\mathbf{d_i}$ is found by the following iteration since the equation is nonlinear

$$\mathbf{f_s^{d_{\mathit{i}+1}}} = \mathbf{f_s^{d_i}} + \boldsymbol{\delta}\mathbf{d}.\nabla\mathbf{f_s^{d_i}} \tag{22}$$

$$\mathbf{f_l} = \mathbf{f_s^{d_i}} + \boldsymbol{\delta}\mathbf{d}.\nabla\mathbf{f_s^{d_i}} \tag{23}$$

Using Equations (22) and (23), error is defined as

$$\mathbf{e} = \boldsymbol{\delta}\mathbf{d}.\nabla\mathbf{f_s^{d_i}} \tag{24}$$

$\boldsymbol{\delta}\mathbf{d}$ is expressed as

$$\boldsymbol{\delta}\mathbf{d} = [(\nabla\mathbf{f_s^{d_i}})'\nabla\mathbf{f_s^{d_i}}]^{-1}.(\nabla\mathbf{f_s^{d_i}})'.\mathbf{e} \tag{25}$$

The iteration stops when Equation (26) is satisfied (i.e. for $\varepsilon = 0.005$)

$$\left\|\mathbf{f_s} - \mathbf{f_s^{d_i}}\right\| < \varepsilon \tag{26}$$

and feature vector $\mathbf{d}$ can be represented in iteration forms

$$\mathbf{d_{i+1}} = \mathbf{d_i} + \boldsymbol{\delta}\mathbf{d} \tag{27}$$

Sometimes Equation (26) does not converge to a desired value. In order to overcome this problem, the epsilon (ε) value calculated for each iteration is compared with the value calculated in the previous iteration and the feature vector information of the smaller epsilon is stored. If the epsilon value increases more than consecutively, the $\mathbf{d}$ vector that was kept is given as an output. This algorithm returns not only a higher quality barcode but also the widths of bars and spaces widths (feature vector $\mathbf{d}$). The following section describes how barcode is decode from the feature vector $\mathbf{d}$.

5.3.5 Match Filtering

Remember that each division consists of two bars and two spaces for a barcode. Each digit of the scan line corresponds to a division. Binary expression of each digit are listed in Table 5. Note that each division of a scan line and binary expression of each digit can be considered as one dimensional signal. For this purpose, we normalize both the scan line data and binary representation of a digit (Table 5) to [1, -1] interval. Since the width of the expressions on Table 5 is seven units, digit correspondences are resized according to the detected division length. Next, each division is correlated with all possible digit correspondences to find a correlation coefficient for each digit. Correlation coefficient is expressed as

$$\boldsymbol{\rho} = \boldsymbol{\gamma} \cdot \boldsymbol{\upsilon}^T \tag{28}$$

where $\boldsymbol{\rho}, \boldsymbol{\gamma}$ and $\boldsymbol{\upsilon}$ refer to correlation coefficient, detected division and a digit representation, respectively.

The expression with the highest correlation coefficient is determined as the equivalent digit of that division. Figure 23 shows the comparison of the signals of an interval of which division equivalent is 3.

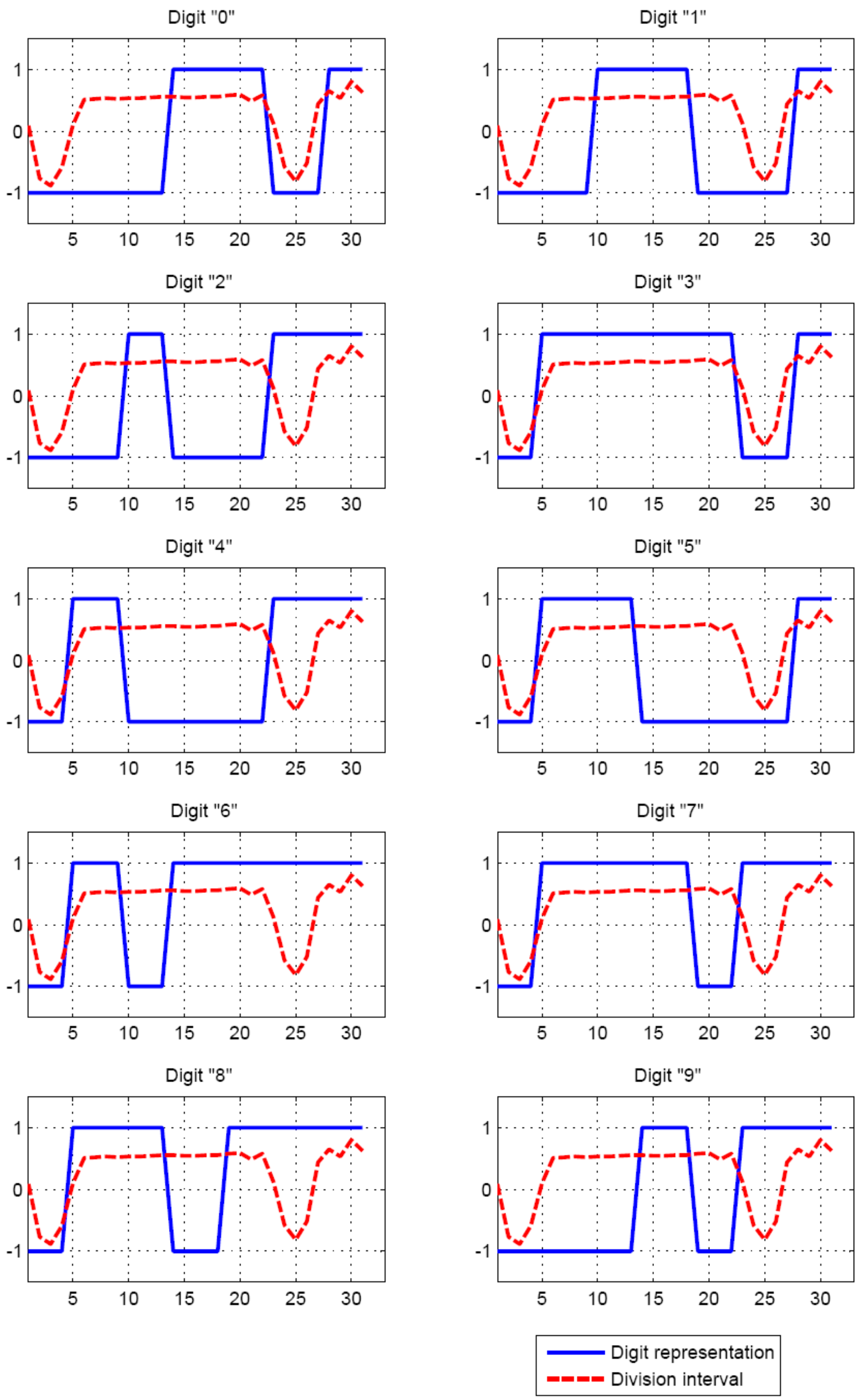


Figure 23. The correlation between possible digit equivalents and a division interval.

When we consider that the barcode data is transcribed in two different ways – the left side and the right side – and that every division on the left side is coded in two different ways, it is seen that there exists the risk of false decoding as the codes used are very similar to each other. For instance, when Table 5 is examined, digit 8 is represented 0110111 in odd case and digit 1 is represented 0110011 in even case. The order of binary representations for two digits is similar except one binary number. On the left side there exists at most ten possible orders (zero to nine). Thus, ten decoding solutions are calculated according to Table 4. The right side is correlated with the potential digits. This way, ten different barcode decoding data are attained. The control code on the barcode is used to find which of the ten different decoded data is the code that has been embedded in the barcode.

5.4 Simulation Results

In order to test the robustness of the barcode decoding technique, scan lines obtained from the barcode images are used. Three of the images that we have used for the test are presented in Figures 24, 25 and 26.

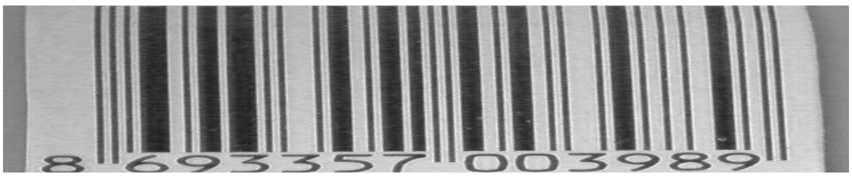


Figure 24. Decoding test image 1

Figure 25. Decoding test image 2

Figure 26. Decoding test image 3

A scan line obtained from the barcode image in Figure 24 is also presented in Figure 27(a-b). By using this scan line, the midpoint values of the bars and spaces in the barcode are obtained. The result of the B-spline smoothing which is applied before finding the midpoint values and the threshold band are displayed in Figure 27(c) and (d), respectively. By using the result of the smoothing, the midpoints of the bars and spaces and thus the start and end borders of the bars and spaces are also found.

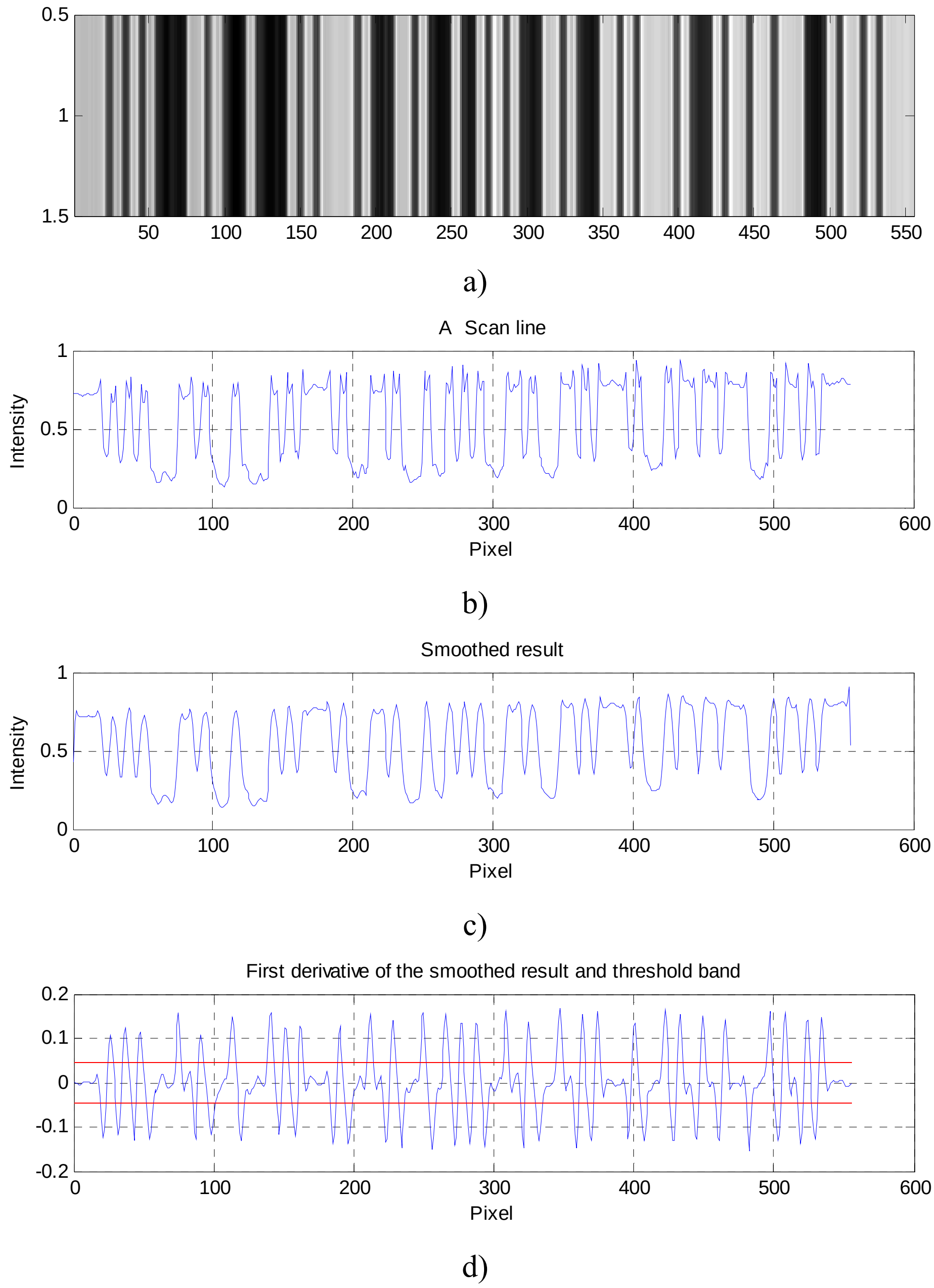


Figure 27. Decoding test image 1 a) a scan line image b) a scan line c) smoothed data d) first derivative of the smoothed result and threshold band

The obtained border information of the bars and spaces are used in super-resolution enhancement and the image quality is increased. An example of super-resolution enhancement is presented in Figure 28. By using the new border information obtained by super-resolution, the digit divisions (as two bars and two spaces) belonging to right and left sides are determined. The decoding process is performed according to the 10 possible orders. The results of the example of Figure 28 are displayed in Table 6. Then, each possible barcode value is calculated by using control code and the result with zero control code value returns the barcode information (6 9 3 3 5 7 0 0 3 9 8 9).

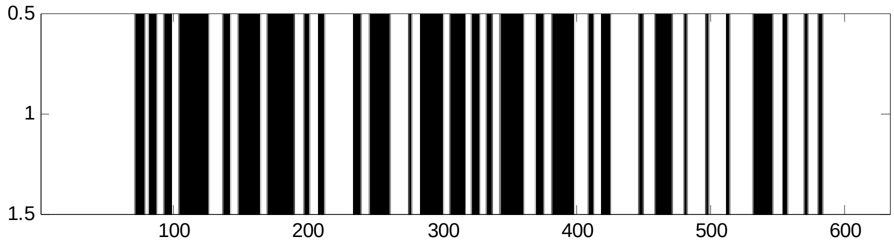
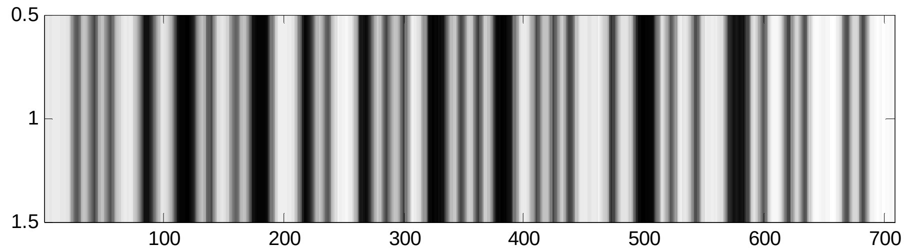


Figure 28. Super-resolution enhancement result for test image 1

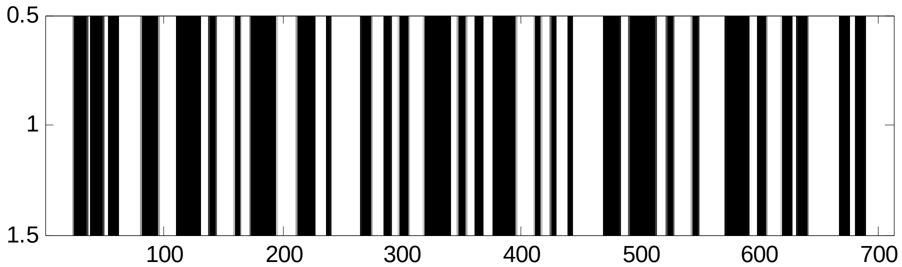
Table 6. All the possible barcode values for decoding test image 1

Start digit	**Left side**						**Right side**						**Solution check**
0	6	8	3	5	1	7	0	0	3	9	8	9	1
1	6	8	4	5	5	2	0	0	3	9	8	9	2
2	6	8	4	3	1	2	0	0	3	9	8	9	9
3	6	8	4	3	5	7	0	0	3	9	8	9	7
4	6	9	3	5	5	2	0	0	3	9	8	9	3
5	6	9	4	5	1	2	0	0	3	9	8	9	5
6	6	9	4	3	1	7	0	0	3	9	8	9	9
7	6	9	3	3	1	2	0	0	3	9	8	9	2
8	6	9	3	3	5	7	0	0	3	9	8	9	0
9	6	9	4	5	5	7	0	0	3	9	8	9	6

Figures 29 and 30 display scan line samples and super-resolution enhanced barcodes for test images 2 and 3. The barcode orientation used in super-resolution application is obtained by finding the starting location of a second scan line. When the original barcode samples and super-resolution results in Figures 29 and 30 are examined, it can be seen that super-resolution enhancement increases the image quality significantly. Thus, analysis of barcode is expected to be more accurate after this enhancement. The results of analysis are displayed in Tables 7 and 8. The performance of the barcode decoding unit are inspected in Chapter 6 in detail.



a)



b)

Figure 29. a) A scan line b) super-resolution result for test image 2

Table 7. All the possible barcode values for decoding test image 2

Start Digit	Left Side	Right Side	Solution Check
0	4 3 8 0 0 6	0 6 5 8 9 6	7
1	4 3 9 0 8 0	0 6 5 8 9 6	9
2	4 3 9 8 0 0	0 6 5 8 9 6	4
3	4 3 9 8 8 6	0 6 5 8 9 6	5
4	4 4 8 0 0 0	0 6 5 8 9 6	8
5	4 4 9 0 0 0	0 6 5 8 9 6	0
6	4 4 9 8 0 6	0 6 5 8 9 6	5
7	4 4 8 8 0 0	0 6 5 8 9 6	7
8	4 4 8 8 8 6	0 6 5 8 9 6	8
9	4 4 9 0 8 6	0 6 5 8 9 6	4

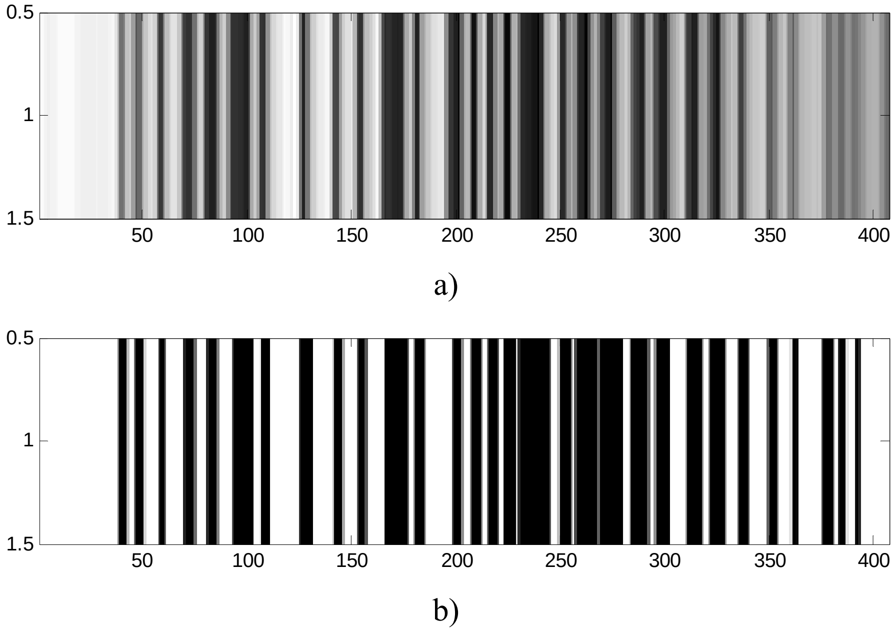


Figure 30. a) A scan line b) super-resolution result for test image 3

Table 8. All the possible barcode values for decoding test image 3

Start Digit	**Left Side**						**Right Side**						**Solution Check**
0	2	8	4	9	1	0	0	2	2	2	7	3	2
1	2	8	3	9	4	8	0	2	2	2	7	3	7
2	2	8	3	8	1	8	0	2	2	2	7	3	8
3	2	8	3	8	4	0	0	2	2	2	7	3	0
4	2	1	4	9	4	8	0	2	2	2	7	3	6
5	2	1	3	9	1	8	0	2	2	2	7	3	5
6	2	1	3	8	1	0	0	2	2	2	7	3	7
7	2	1	4	8	1	8	0	2	2	2	7	3	9
8	2	1	4	8	4	0	0	2	2	2	7	3	1
9	2	1	3	9	4	0	0	2	2	2	7	3	3

CHAPTER 6

COMBINATION OF BARCODE LOCALIZATION AND DECODING

Barcode reading system consists of barcode localization and barcode decoding sections. The localization forms the first part of the Barcode reading system. In this part barcode region is localized via FHTA which is both advantageous from the point of time complexity as we suggested in Chapter 3 and provides the barcode orientation information. Barcode decoding is performed through a series of steps; B-spline smoothing, estimation of peak locations, super-resolution enhancement and match filtering, which are explained in Chapter 5. This chapter discusses the performance of the overall system which is the combination of the two parts.

The integration of the two parts in the barcode reading system is achieved by extracting a scan line from the localized region and feeding it to second part. After obtaining the scan line, the barcode reading system removes the region which does not belong to the barcode. This process as executes as follows: the thresholding applied on the scan line and threshold crossing locations are noted. The threshold value is chosen by dividing the sum of the minimum and maximum intensity values with 2.2. A rough estimation of minimum bar width is obtained by using threshold crossing. This width information is used to discard non-barcode region and detect barcode start position.

This chapter includes test results of our proposed barcode reading system. Test images are obtained from two different cameras. For the performance of the barcode reading system, the test images are tested on "DTK Barcode Reader SDK" and "Softek Barcode Reader" and the results are compared.

6.1 Simulation Results

The efficiency of proposed barcode reading system on 1632x1224 and 352x288 sized images which are acquired by Sony-Ericsson K750i model cellular phone and a wireless camera, respectively. Tests are performed over 71 cellular phone and 27 wireless camera images. The robustness of the barcode reading system is related to the localization and decoding results directly. The efficiency of the system over 98 images is summarized in Table 9.

Table 9. Performance of the localization, decoding and barcode reading system

Image size	**Explanation**	**Localization performance**	**Decoding performance**	**Overall system performance**
1632x1224	# of images	59	33	30
	Percentage	82.89	46.48	42.25
352x288	# of images	20	1	0
	Percentage	74.07	3.70	0

Table 9 show that localization fails for 19 over a total of 98 images. The source of the problems are displayed in Table 10 for those images.

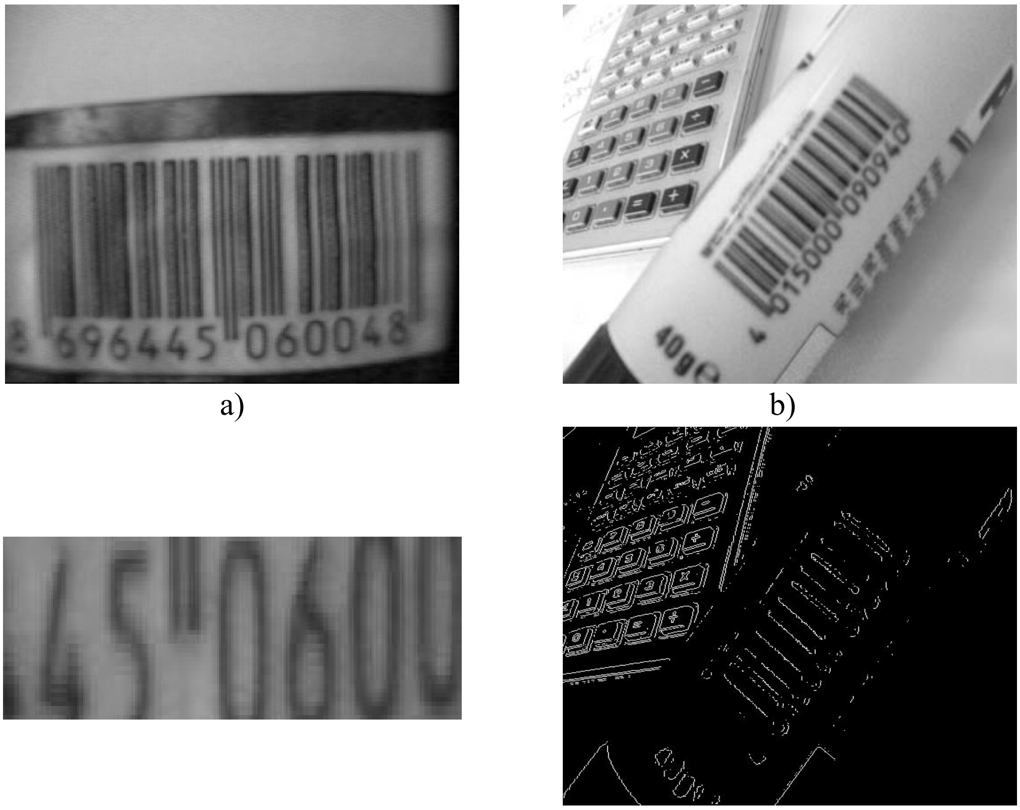
Table 10. Details of the problem encountered in localization

Image size	**Explanation**	**False angular orientation detection**	**Failure of edge detection**	**Incomplete barcode region extraction**	**Total**
1632x1224	# of images	6	1	5	12
352x288	# of images	2	0	5	7

Source of the problems can be summarized as follows

- There exist dense regions of high frequency components other than the barcode area.
- Failure of edge detection process.

For such reasons, localization cannot be done on 19 images. The 4 test images for which correct localization cannot be achieved are displayed in Figures 31 and 32.



a) b)

c) d)

Figure 31. 352x288 sized a-b) test images c-d) localization results

a) b)

c) d)

Figure 32. 1632x1224 sized a-b) test images c-d) localization results

The success of our barcode reading system in decoding of the barcode information is also displayed in Table 9. The decoding performance is tested by feeding extracted scan line as the input without considering the barcode localization results. Our proposed decoding method has decoded 34 images with success. The barcode information cannot be correctly decoded for the remaining 64 images. The problems encountered for these 64 images are presented in Table 11 in detail.

Table 11. The problem details encountered in decoding technique

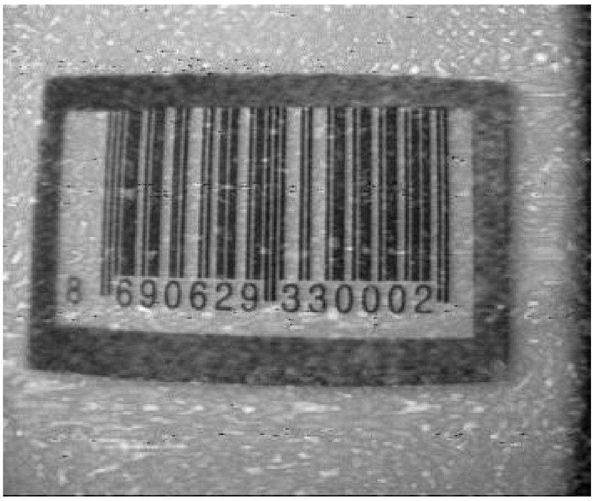
Image size	**Explanation**	**False detection of midpoints**	**False detection of borders**	**Incomplete scan line extraction**	**Total**
1632x1224	# of images	25	11	2	38
352x288	# of images	23	3	0	26

46.48% success is obtained for the images of 1632x1224 cellular phone images. Success rate for the wireless camera images stayed at 3.7%. Considering Table 11, our inferences can be listed as follows

- Peak location estimation may fail in presence of high blur.
- Although the bars are often black, in rare cases different color bars (i.e. blue) exist. In such cases, the threshold band application we use in peak location estimation fails.
- Peak location estimation fails when barcode resolution (i.e. number of pixels representing the thickest bar) is not high enough
- Although false detection of the borders of the images with higher quality images can be tolerated by super-resolution, the images obtained by the wireless cameras cannot be tolerated due to severely low quality.
- Misdetection of borders result in incorrect correlation based barcode reading.
- Barcode start and end positions are sometimes not correctly detected on the scan line.

Three of the images for which the barcode information cannot correctly be decoded, are displayed in Figure 33.

a)



b)

c)

Figure 33. a) Barcode with blue bar color b) image for which the SNR value is low c) image with high distance between the barcode and the camera

Finally, the performance of the barcode reading system (overall system) is directly proportional with the successes of the localization and decoding sections separately. Thus, the same problems encountered in localization and decoding are also seen in the barcode reading system. When the success of the barcode reading system is analyzed, it is seen that almost half of the cellular phone images can be correctly decoded whereas none of the wireless camera images can be decoded correctly.

6.2 Performance of the Barcode Reading System

The performance of the barcode reading system is observed by comparing it with the image process based barcode reading softwares which are developed commercially. The interfaces of the two softwares are presented in Figures 34 and 35.

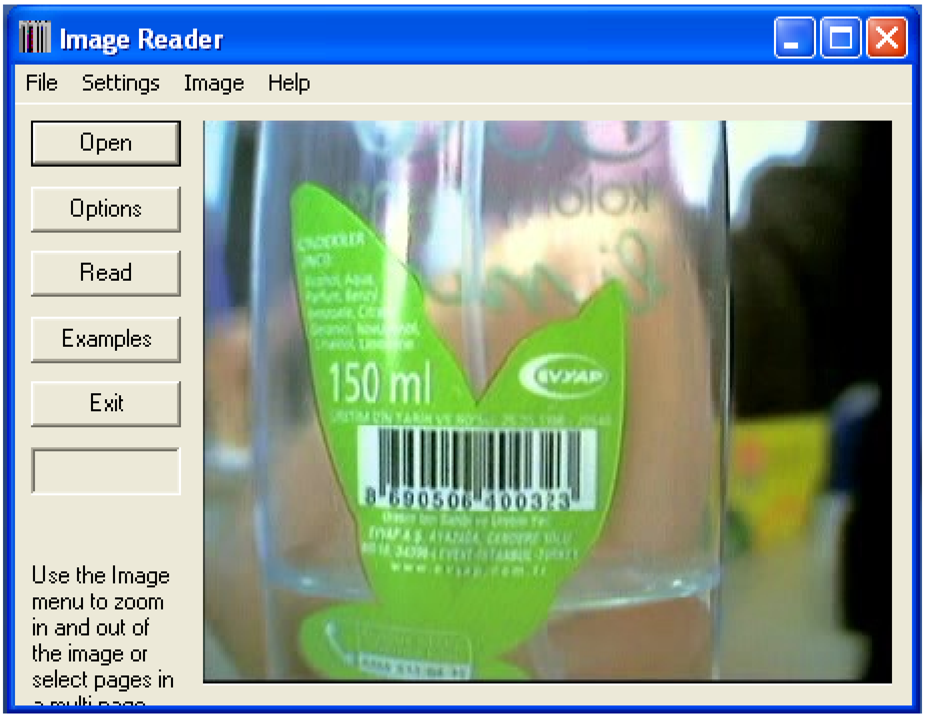


Figure 34. Softek barcode reader version 7.1.2

Figure 35. DTK barcode reader SDK version 3.7.5

The two softwares are tested over the same 98 cellular phone and wireless camera images. The test results related to the commercial softwares are displayed in Table 12.

Table 12. Test results of the commercial softwares and barcode reading system

Image size	Explanation	DTK barcode reader SDK	Softek barcode reader	Our proposed barcode reading system
1632x1224	# of images	35	12	30
	Percentage	49.30	16.90	42.25
352x288	# of images	7	0	0
	Percentage	25.93	0	0

When the performance of our developed barcode reading system is compared with the two other commercial softwares, it is observed that:

- The performance of the DTK barcode reader and the performance of our system are close to each other in cellular phone images. However, the performance of Softek barcode reader is worse than the others.
- For the wireless camera images, our system and Softek barcode reader cannot perform any decoding. DTK barcode reader SDK can achieve correct decoding for those images but with very low success rate, 26%.
- Our proposed system and Softek barcode reader cannot achieve correct decoding colored (i.e. blue) barcodes.
- DTK barcode reader cannot achieve correct decoding in case the barcode orientation is at 0 degrees. However, Softek barcode reader and our proposed system achieve correct decoding for such cases.

CHAPTER 7

CONCLUSIONS

In this thesis, an image processing based barcode reading system is presented and its performance issues are discussed. The aim of the work is to develop a system which provides the decoding of the barcode information from acquired barcode images. This developed system is a part of the Pusulal project.

In order to increase the robustness and decrease time complexity of the barcode reading system, the system is divided into two parts: barcode localization and barcode decoding. After analyzing the localization and decoding sections separately, these two sections are integrated into the barcode reading system.

A new approximation to Fast Hough Transform is developed for the barcode localization application. The robustness of this method is tested various barcode images, barcodes being at different orientations. The images are acquired by Sony-Ericsson cellular phone and a wireless camera. Besides obtaining successes of 82.47% and 74.07% (same with the previous FHT implementations), respectively for the two different cameras, it is observed that run time of the proposed approximation is significantly less than that of the previous FHT implementation.

Barcode decoding process is executed over a scan line extracted from the localized region. The midpoints of the bars and the spaces are estimated for the decoding process. Midpoint estimation is carried out over B-spline smoothed data, in order to avoid overestimation of local minima and maxima. Midpoints are used to detect bar/space borders, which are later used to reconstruct a higher quality barcode sample by least squares approximation. This step also provides more accurate border estimations, from which bar/space widths can be determined. Finally, barcode data is deciphered by use of bar/space widths and match filtering. The proposed method is also tested over cellular phone and wireless camera images, 46.48% and 3.7% success rates are achieved, respectively.

The efficiency of the barcode reading system which is the combination of the barcode localization and decoding techniques is compared with two different image processing based barcode reading softwares. The results show that our proposed system is 60% more successful than “Softek barcode reader” software and is 15% worse than “DTK barcode reader SDK” software. In our proposed system, estimation of peak location method does not work effectively under the low quality image conditions and different bar colors. As a result of this, midpoints and border information of the bars and spaces, which belong to the scan line, are obtained incorrectly on these kinds of images. For ten of the images which decoding could not be achieved, we succeeded to achieve recognition on 2 more images by entering border information for bars and spaces on scan line manually. On this ground, if the estimation of peak locations technique is developed, the barcode reading system will be more effective.

We achieved 42.25% better results than that of wireless camera at tests of the combined barcode localization and decoding algorithm with cellular phones. This is due to the extremely low quality (low SNR, low quality focus) images acquired by the wireless camera. Neither our proposed system nor “Softek barcode reader” software achieves any correct reading with those images. Even “DTK barcode reader SDK” algorithm, high quality commercial software achieves as low as 25.93% correct reading results with wireless camera images. So, although enhancement techniques help to improve bar/space location estimation, not big improvements can be obtained by very low quality images of severe blur and noise.

REFERENCES

[1] Kuroki, M., Yoneoka, T., Satou, T., Takagi, Y., Kitamura, T., Kayamori, N., “Bar-code Recognition System Using Image Processing”, Sixth International Conference On Emerging Technologies and Factory Automation, pp. 568-572, 1997.

[2] Ohbuchi, E., Hanaizumi, H., Hock, L.A., “ Barcode readers using the camera device in mobile phones ” International Conference on Cyberworlds, pp. 260 – 26, 2004.

[3] URL: http://www.dtksoft.com/products.php. Site last visited in August 2008

[4] URL: http://www.softeksoftware.co.uk/products.html. Site last visited in August 2008

[5] Jain, A. K., Chen, Y., ”Bar code localization using texture analysis”, Proceedings of the Second International Conference on Document Analysis and Recognition, pp. 41-44, 20-22 Oct. 1993.

[6] Öktem, R., “Barcode Localization Algorithm Using Binary Morphology in Wavelet Domain - Dalgacık Bölgesinde İkili Morfoloji Kullanarak Çubuk Kod Yersenimi”, SIU2004, 28-30 April 2004.

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

[8] Joseph, E., Pavlidis, T., **“**Bar code waveform recognition using peak locations” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 16, no. 6, pp. 630–640, June 1994.

[9] Bailey, D. G., “Super-resolution of bar codes “, Journal of Electronic Imaging, 10(1), pp. 213-220, January 2001.

[10] Öktem R., Öktem L., “A Superresolution Approach for Bar Code Reading”, Proc. Eusipco 2005, Antalya, TURKEY.

[11] Normand, N., Viard-Gaudin, C., “A two-dimensional barcode reader”, Proceedings of the 12th IAPR International Conference on Pattern Recognition, Vol. 3, pp. 201-203, October 9-13, 1994.

[12] S.-J. Liu, H-Y. Liao, L-H. Chen, H-R. Tyan, J-W. Hsieh, “Camera-based bar code recognition system using neural net“, Proceedings of 1993 International Joint Conference on Neural Networks, Vol. 2, pp. 11301-1305, Oct. 1993.

[13] Muniz, R., Junco, L.O., ”A robust software barcode reader using the Hough transform”, International Conference on Information Intelligence and Systems, pp. 313 – 319, 31 Oct.-3 Nov. 1999.

[14] Duda R. O., Hart P. E., “Pattern Classification and Scene Analysis”, John Wiley&Sons, New-York, 1973.

[15] Forsyth, D. A., Ponce J., “Computer Vision: *A Modern Approach*”, Prentice Hall, pp. 437-438, 2003.

[16] Gonzalez, R. C., Woods, R. E., “Digital Image Processing “, second edition, pp. 587-590, 2002.

[17] Li H., Lavin M. A., and Le Master R. J., “Fast Hough Transform: A hierarchical approach”, CVGIP, Vol. 36, pp. 139-161, 1986.

[18] Zapata, N., Guil, J. Villalba, “A fast Hough transform for segment detection”, IEEE Trans. Image Process., Vol. 4, no. 11, pp. 1541-1548, 1995.

[19] Kincaid, D., Cheney, W. “Numerical Analysis Mathematics of Scientific Computing”, Brooks Cole Publishing, pp. 315-333, 1991

[20] De Boor, C., “A Practical Guide to Splines”, Applied Mathematical Sciences, Vol. 27, Springer Verlag, 1978

[21] I. J. Schoenberg, “Cardinal Spline Interpolation”, Philadelphia, PA: Soc. Indus Apllied Math, 1973.

[22] Canny, J., “A Computational Approach to Edge Detection”, IEEE Transactions on Pattern Analysis and Machine Intelligence,Vol. PAMI-8, No. 6, pp. 679-698, 1986.

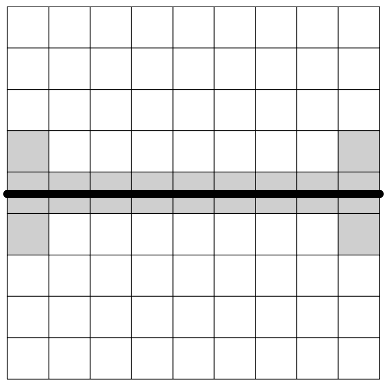
[23] Ritter, G. X., “Handbook of Computer Vision Algorithms in Image Algebra” Joseph N. Wilson CRC Press, pp.196, 1996.

[24] URL:http://www.barcodeisland.com. Site last visited in August 2008.

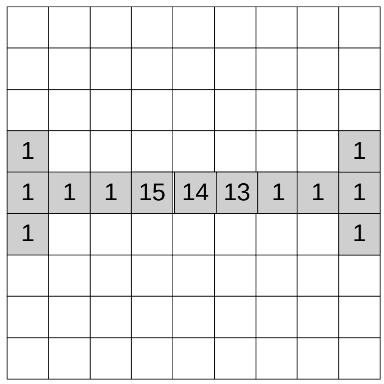
[25] Reinsch, C. H. “Smoothing by Spline Functions”, Numerische Mathematik 10, pp. 177-183, 1967.

APPENDIX

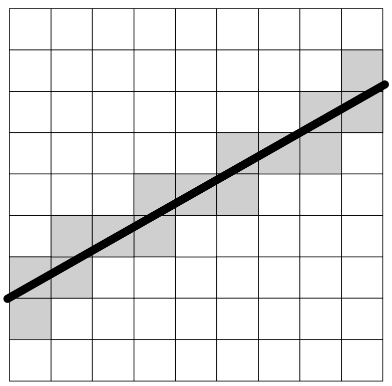
EIGHT ORIENTATIONS AND CORRESPONDING LABEL VALUES



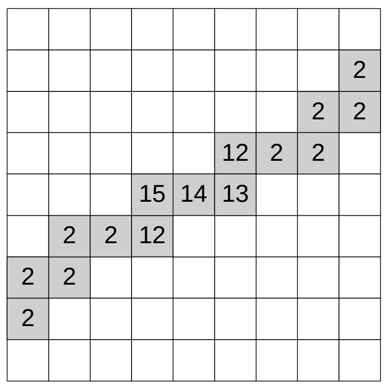
a) 0° line



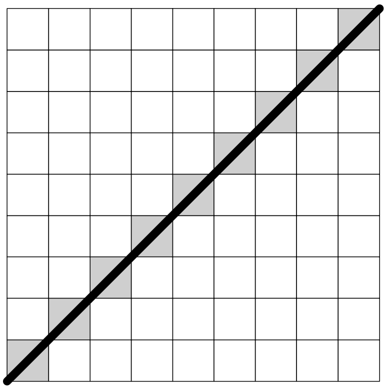
b) 0° label values



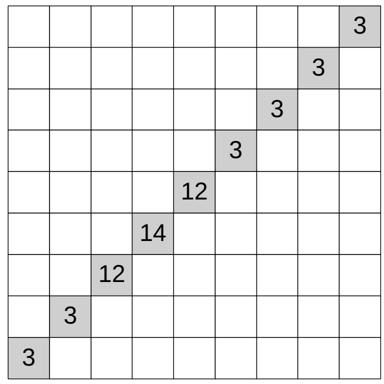
c) 30° line



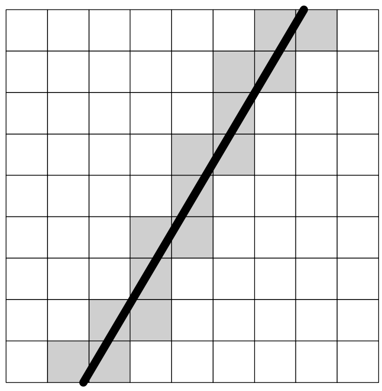
d) 30° label values



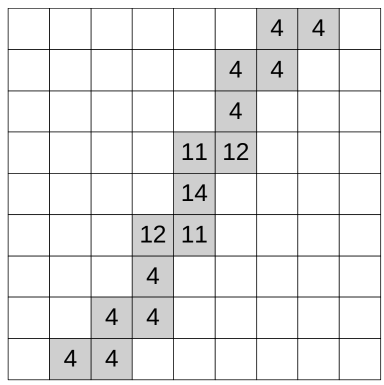
e) 45° line



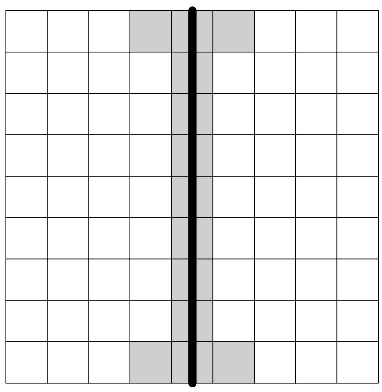
f) 45° label values



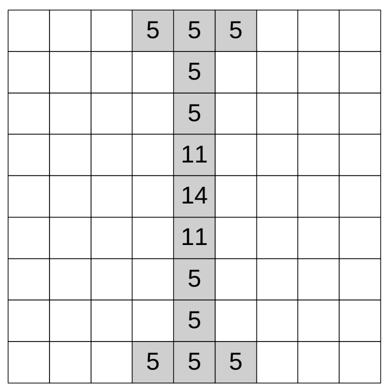
g) 60° line



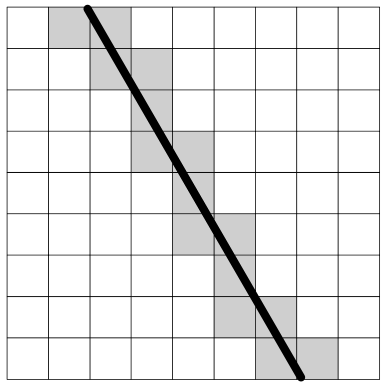
h) 60° label values



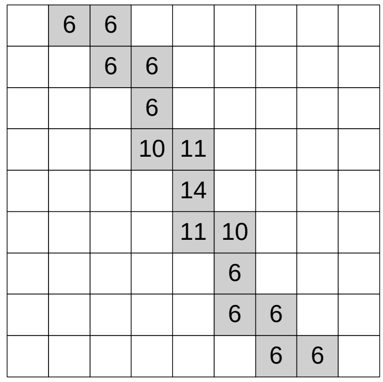
i) 90° line



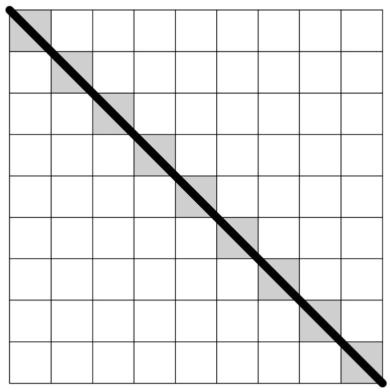
j) 90° label values



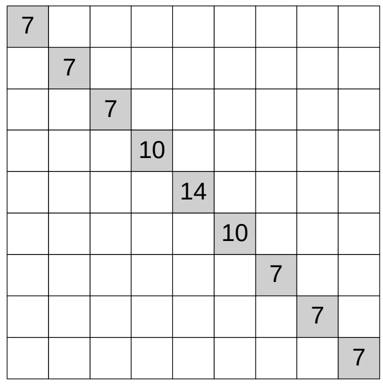
k) 120° line



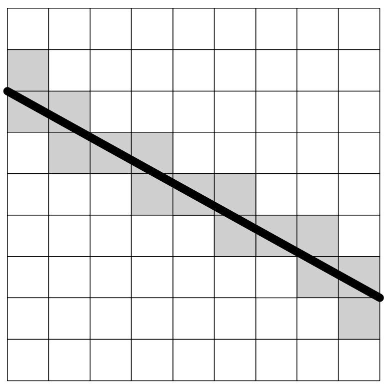
l) 120° label values



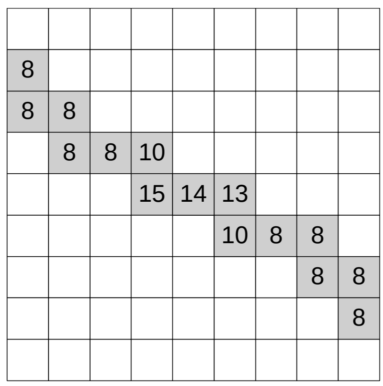
m) 135° line



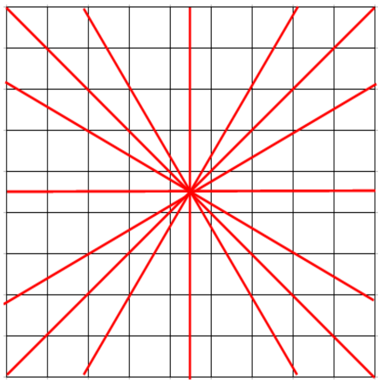
n) 135° label values



o) 150° line



p) 150° label values



q) 8 Orientations

7	6	6	5	5	5	4	4	3
8	7	6	6	5	4	4	3	2
8	8	7	6	5	4	3	2	2
1	8	8	10	11	12	2	2	1
1	1	1	15	14	13	1	1	1
1	2	2	12	11	10	8	8	1
2	2	3	4	5	6	7	8	8
2	3	4	4	5	6	6	7	8
3	4	4	5	5	5	6	6	7

r) 9x9 Mask with label values