

# A LOW COST OPTICAL BARCODE READER USING A WEBCAM

Ruwan Janapriya, Lasantha Kularatne, Kosala Pannipitiya, Anuruddha Gamakumara and Chathura de Silva.

Department of Computer Science and Engineering, University of Moratuwa, Sri Lanka.

{janapegr, kulakgli, pannpakg, gamagkma, crdsilva} @cse.mrt.ac.lk

## ABSTRACT

We have developed a low cost, WebCam based optical barcode reader, which can extract and decode the sequence on a cluttered background. It is composed of three functions: barcode localization from the raw image, transformation of the localized barcode and decoding the sequence with an intelligent algorithm. The localization method is based on detecting the areas with the maximum density difference in two normal directions. The transformation method, capable of identifying any orientation, is based on the Hough line detection method. The decoding method is based on the peak/valley detection method of the barcode waveform and a consistency checking method. The consistency checking method, a constraint network, employs artificial intelligence searching methods. The algorithms used in the barcode reader have been tested on hundreds of images with an accuracy of more than 99%.

**Keywords:** Barcode Localization, Barcode Recognition, WebCam, Hough Transformation, Peak Detection

## INTRODUCTION

Barcodes are a class of the simplest printed patterns that can be reliably recognized by a computer. These codes consist of sequence of parallel, light and dark stripes printed on papers. Information is encoded in the width of the stripes; sequences of a fixed number of stripes encode a character. The sequence that defines the character is unique for the barcode type. Each barcode is started with start bars and ended with stop bars. In some barcode types a checksum digit is included in order to validate the reading. Barcodes are used in wide variety of areas such as for industrial identification, for publications, for pharmaceutical products and for postal codes. There are several types of barcodes. The most commonly used types are EAN 13, UPC, ITF, Codebar, Code 128, ISBN, Pharma (pharmaceutical) and FIM (postal).

Since 1952, various kinds of scanners were developed to read barcodes. Most of these barcode readers use a laser beam to scan the barcode and give the resulting value. There are some disadvantages in these kinds of barcode readers. These disadvantages includes that the barcode has to be manually oriented towards the laser beam to get the barcode value, high cost and the harmfulness for the user from the exposure to the laser beam.

In the recent past several techniques and algorithms have been proposed for vision-based barcode reading. These Include two-step paradigm of image feature extraction method(1), Hough Transformation(2), Neural Networks Method(3)(4), Texture Analysis(5) and Mathematical Morphology(6) for barcode localization within the image and Selective Sampling method(7), EM Algorithm(8)(9) and Statistical Pattern Recognition(10) for decoding the sequence.

## **SmartCamBCR: A LOW COST BARCODE READER**

Due to the problems associated with the currently available barcode readers, the need for a sophisticated barcode reader is evident. This led to the development of *SmartCamBCR*, the first barcode reader to use a *web cam* (a small camera used for sending images over the Internet) as the capturing device and to use image processing and artificial intelligence techniques to obtain the barcode value. Since a web cam is a relatively inexpensive device, and most of the business applications already use computers, the cost of this barcode reader would be very much less than that of available barcode readers.

The proposed barcode reader is capable of identifying any barcode irrespective of its orientation and background, making it an entirely autonomous system. Being able to decode damaged or deformed barcodes, the reader is more useful for most of the real-time applications. Compared with other readers, with our reader we have obtained a comparatively high rate of successful readings with the barcodes damaged from scratches, stains and fading. The intelligent decoding technique has proven its robustness in situations where the presence of limited resolution depth (320×240) of the WebCam and the heavy blurring of the captured images when the WebCam is not focused. The peak detection method is far more superior to edge detection methods as it has a higher degree of immunity to noise. Furthermore, the peak detection method is not affected by the ink spread problem, which can be ensued in barcode printing process. Despite the high performance of the peak detection method, the images can be contaminated with noises considerably. The consistency checking with the constraint network rectifies the effect of the noise by checking the relationships of the widths of the nearby bars and taking into consideration of the decoding rules of EAN13.

We are proposing a real-time and autonomous system for the practical use. In this case, the system performance is inevitably influencing its success. Taking into consideration of the fact that image processing methods are consuming a large amount of processing time, we propose a method of controlling the real-time performance effectively. The system utilizes the full processing power only when necessary, gathering the knowledge of the barcode's details and behavior. After gathering of the knowledge, the system can use the gathered knowledge to predict the barcode's details and behavior in the future for a short period. This will reduce processor utilization. This method showed an improvement of over a factor of three, of the frame rate.

Even though the *SmartCamBCR* cannot read deformed barcodes, it could read partially damaged barcodes with high robustness. This is achieved by scanning through multiple lines of the barcode image. The illumination problem could be overcome by properly illuminating (controlling of the light level) the barcode image.

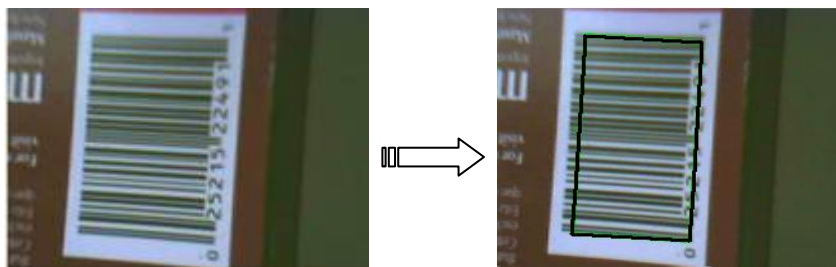
## THE PROPOSED METHODOLOGY

We propose a three-stage process, which effectively decodes the barcode contained in a given image as shown in the figure. First the barcode region is localized from a cluttered background, using a combination of gradient and morphological operators. Next the region of interest (ROI) is extracted after applying necessary transformation using Hough line detection(2) method. Finally the generated barcode waveforms are decoded by recognizing peak locations.

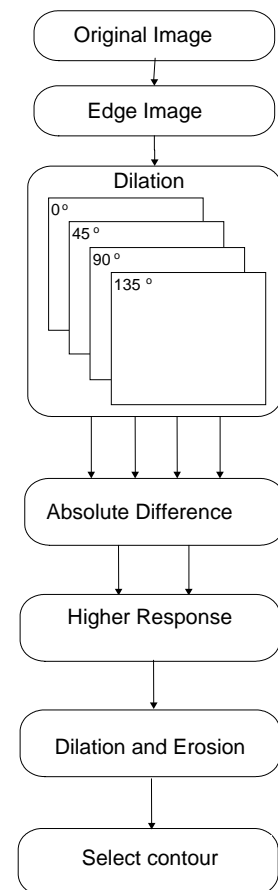
### Barcode Localization

Localization process is based on mathematical morphology. The two fundamental operators of mathematical morphology are erosion and dilation. Dilation, in general, causes objects to dilate or grow in size; while erosion causes objects to shrink. The amount and the way that they grow or shrink depend upon the choice of the structuring element. Also this process uses *Canny edge detection* mechanism to get the edge image from the given original image. In Canny method it uses a filter which is narrow as possible to provide suppression of high frequency noise and to provide good localization of the edges.

In our approach, as indicated in the figure, first original image is converted to edge image. Dilation is applied to that edge image taking structuring element as the line in four different angles,  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$  &  $135^\circ$ . Considering difference of two dilate images having structuring element's angular difference of  $90$ , two images can be taken in the next step. From those two images one can be selected, which has higher response. Then further dilation and erosion applied to that selected image using the respective structuring elements. From that image, the blob which corresponds to barcode area is selected using following formula [1]. Finally the barcode can be localized as shown in the figure 2.



**Figure 2: Barcode Localization**



**Figure 1: Flow of Localization Module**

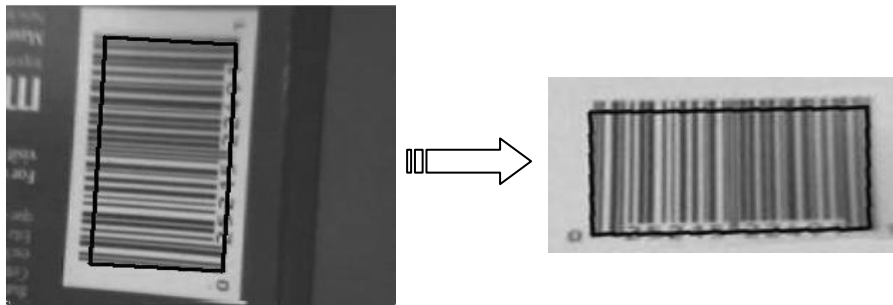
$$\frac{\text{Blob Contour Area}}{\text{Area}} > 0.65 \times \frac{\text{Area of the Minimum Area Rectangle}}{\text{Surrounded the Contour.}} \quad [1]$$

## Barcode Transformation

Transformation process is based on Hough line detection method. In this method, the original image plane is transformed into the Hough plane. In the Hough plane, each point in the original plane is represented by a line. Since a barcode is a set of bars or lines parallel to each other, this method can be used to find the angle of the bars of the barcode.

First barcode region given by the localizer is transformed into Hough plane. Now all possible candidates for lines in barcode region are converted to points having corresponding  $\theta$  and a  $\rho$ . These points are extracted by thresholding the transformed image taking a suitable threshold. From extracted points, an angle is selected which has the maximum span.

After detecting the orientation, the whole image is rotated in reverse direction by the angle determined above. If necessary, other transformations such as scaling are applied to the image. Considering the minimum area of the contour (this contour is given by the localization phase) as the region of the barcode, the final image is generated by cropping the original image into the bounding rectangle of the minimum area rectangle.



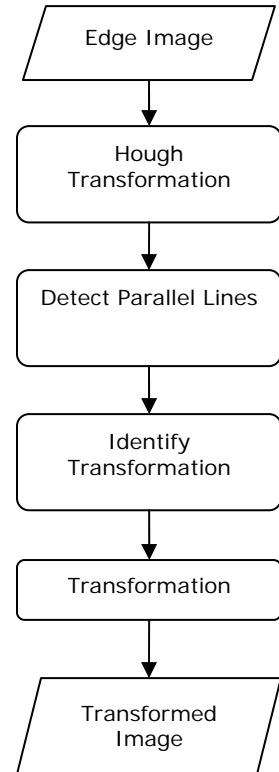
**Figure 4: Barcode Transformation**

## Barcode Decoding

The decoding method mainly consist of two parts; generation of the barcode waveform from the sequence and deciphering the barcode value from the waveform. For the first part, waveform generation, we have adopted the peak detection method [3] to recognize the barcode.

### Waveform Generation

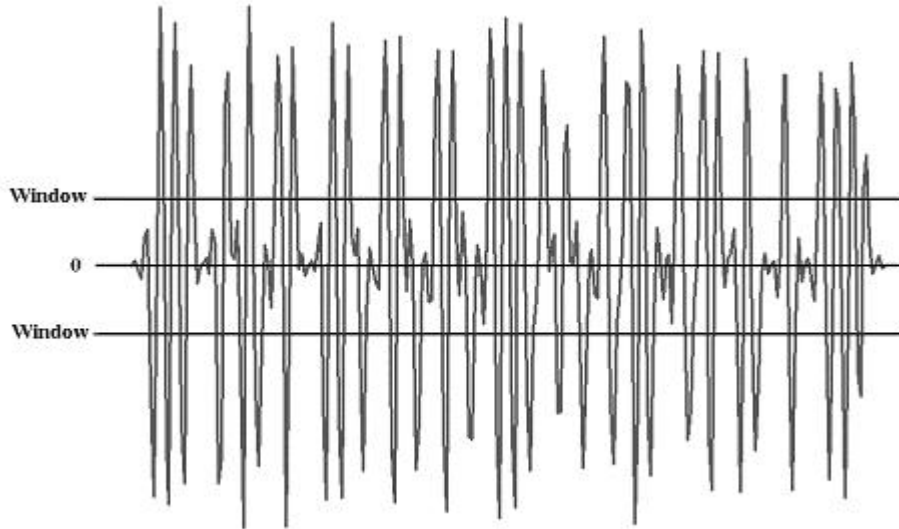
The zero crossings of the equation [2] give the peaks and valleys of the function  $g(x)$ . In equation [2],  $f(x)$  is the intensity function of the barcode waveform and  $\sigma$  is the standard deviation of the *Point Spread Function*.



**Figure 3: Flow Chart of Transformer**

$$g(x) = f(x) - \frac{\sigma^2}{2} f''(x) \quad [2]$$

The generated waveform of the function  $g(x)$  is illustrated in the figure 5. The steep peaks and valleys are corresponding to the white bars and black bars of the barcode respectively. Due to the presence of the noise, there are some infrequent narrow peaks and valleys appeared in the waveform. To eliminate the zero-crossings due to the noise, a window has been selected from both sides of the zero axes as shown in figure 5. The sophisticated algorithm (10), employed for detecting the zero crossings will effectively eliminate the noise. The introduction of the smoothing effect initially is for further narrowing of the noise.



**Figure 5. Barcode Waveform with the Zero-Crossing Window**

For the implementation, we suggest representing the waveform with *splines*. It should be noted that there is an inherent problem with splines when representing straight lines. For the straight lines, the spline tends to introduce a ripple effect at the corners. The effect of the ripples was insignificant in our implementation.

### Waveform Decoding

We introduce a novel method for waveform decoding in this section, considering the EAN13 barcode type as the sample barcode type. After locating the peaks and valleys, the minimum bar width can be determined (10) from the distances between consecutive peaks and valleys. The normalized distances can be calculated by dividing the distances between consecutive peaks and valleys by the minimum bar width. Table 1 shows the possible stripe to stripe combinations with the normalized distances.

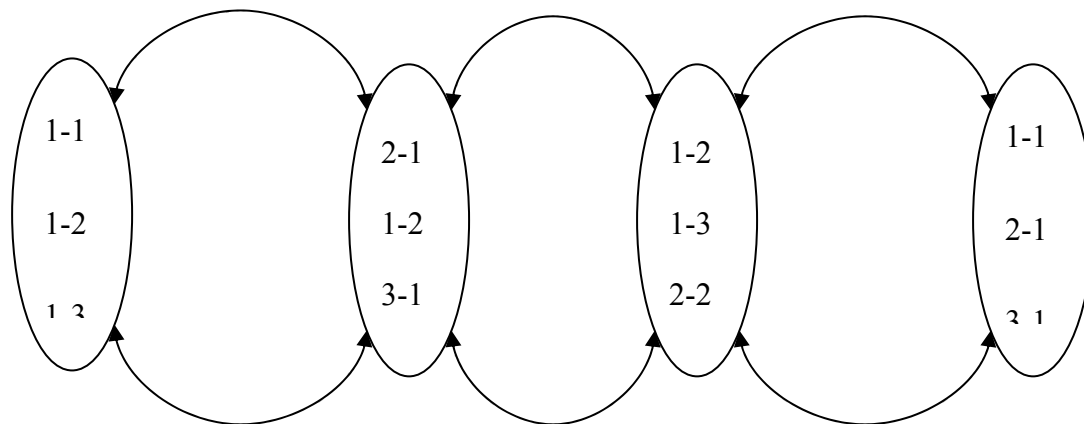
**Table 1. Possible Stripe to Stripe Combinations with the Normalized Distances**

Normalized Distance	Stripe to Stripe Combinations
1.0	1-1
1.5	1-2, 2-1
2.0	1-3, 2-2, 3-1
2.5	1-4, 2-3, 3-2, 4-1
3.0	2-4, 3-3, 4-4
3.5	3-4, 4-3
4.0	4-4

Due to the noise, the calculated normalized distances are shifted from their original values. Considering the shift of the normalized distances due to the noise, a predefined probability can be assigned to the stripe-to-stripe combination. Table 2 shows a part of the probabilities with the different normalized distances ranges.

**Table 2. The Probabilities of Stripe-to-Stripe Combinations.**

Normalized Distance Range		Probabilities of Stripe-to-Stripe Combinations										
		1-1	1-2	1-3	1-4	2-1	2-2	2-3	2-4	3-1	3-2	3-3
0.000	0.625	0	0	0	0	0	0	0	0	0	0	0
0.625	0.750	2	0	0	0	0	0	0	0	0	0	0
0.750	0.875	3	0	0	0	0	0	0	0	0	0	0
0.875	1.000	5	0	0	0	0	0	0	0	0	0	0
1.000	1.125	5	0	0	0	0	0	0	0	0	0	0
1.125	1.250	3	2	0	0	2	0	0	0	0	0	0
1.250	1.375	2	3	0	0	3	0	0	0	0	0	0
1.375	1.500	0	5	0	0	5	0	0	0	0	0	0
1.500	1.625	0	5	0	0	5	0	0	0	0	0	0
1.625	1.750	0	3	2	0	3	2	0	0	2	0	0



**Figure 6. The Constraint Network**

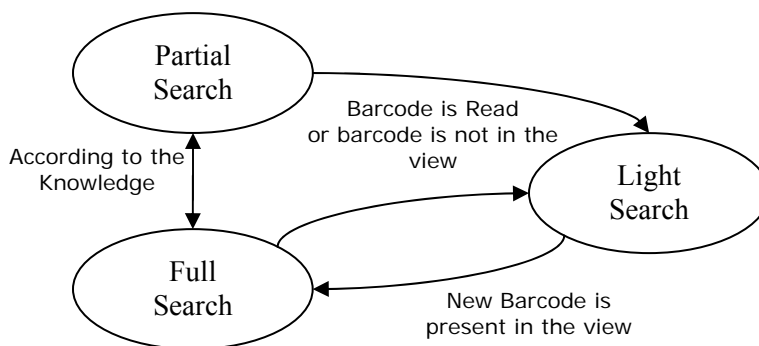
The constraint network can be generated with normalized distances and the probabilities shown in Table 2. Figure 6 illustrates the structure of the constraint network. The domains of the constraint network contain the stripe-to-stripe combinations. The matching of the stripe-to-stripe combinations in the consecutive domains and the value of the probability can be considered as the rules for checking the consistency.

Evaluating the constraint network until it yields a unique solution will give the most probable stripe-to-stripe combination sequence. If there are many solutions, then the rule; keeping the combinations with highest probabilities and discarding the other, can be applied. The result of the constraint network can be easily translated to the barcode value with the lookup table.

### Real-time Implementation

To use the computational power efficiently the algorithm is slightly modified to fit in to a multi-state model. The model consists of 3 states namely; *Full Search*, *Partial Search* and *Light Search*.

Full search state uses the maximum computational power. The implementation is done in such a way that most of the time it will operate on light search and partial search respectively. Initially the system operates on light search state, searching for a barcode. If present it will move on to the full search and try to decode it. As soon as it completes its reading it will move on to the light search mode, though the barcode is still in the scene. System reacts for the barcode if only it is removed from the scene and introduced again. When the system operates on full search it will gather some knowledge about the scene; lighting condition and orientation of the current barcode. This knowledge will be used in the partial search state where some steps in the algorithm could be eliminated since the system 'knows' about the scene enhancing the real-time functionality of the system, improving the frame rate. Figure 7 illustrates the states of the real-time implementation.



**Figure 7: Multi-State Approach of the Real-Time System**

## CONCLUSION

In this paper, we have discussed about a novel approach for barcode localization and barcode recognition. For barcode localization we have used a simple mathematical morphological operation method. For barcode recognition we have used a peak/valley detection method together with a consistency checking method.

Eventually, the test results proved that our algorithm is robust and reliable. Also the realtime algorithm proved it's viability in commercial applications. The algorithm uses an autonomous localization technique making the product more effective in autonomous systems. For an example in a product line, the barcodes on moving objects can be localized and recognized without human intervention.

We are currently investigating the use of our algorithm integrated in a hardware based platform to achieve higher performance.

## RESULTS

### Step by Step Demonstration of the Algorithm

#### Barcode Localization



Figure 8: Original Image

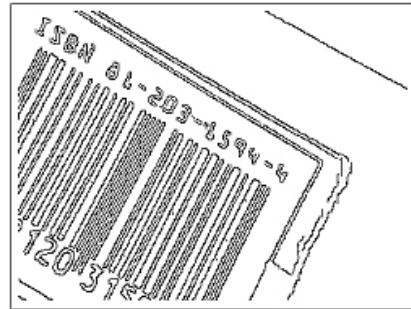


Figure 9: Edge Image

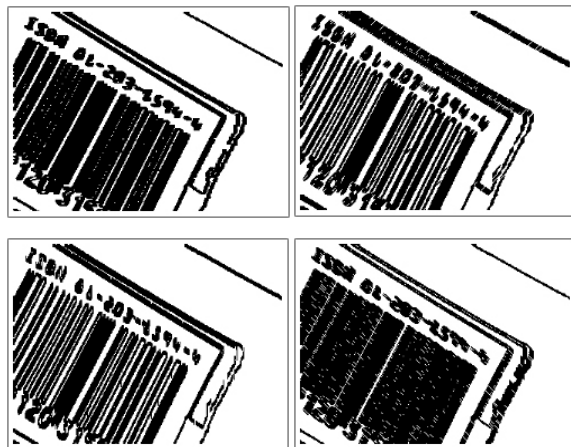


Figure 10: Dilation Using 'Line' Structuring Element in 0o,45o,90o and 135o

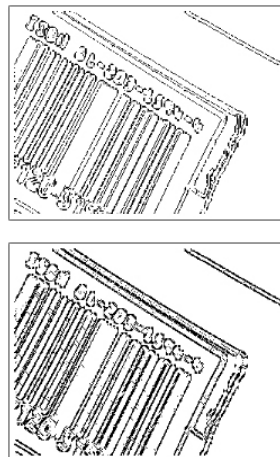
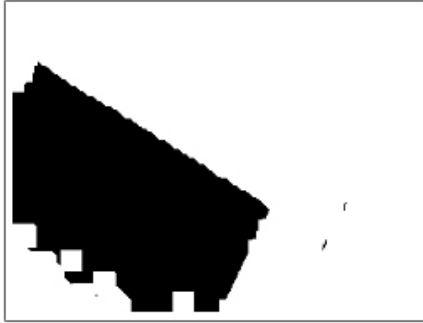


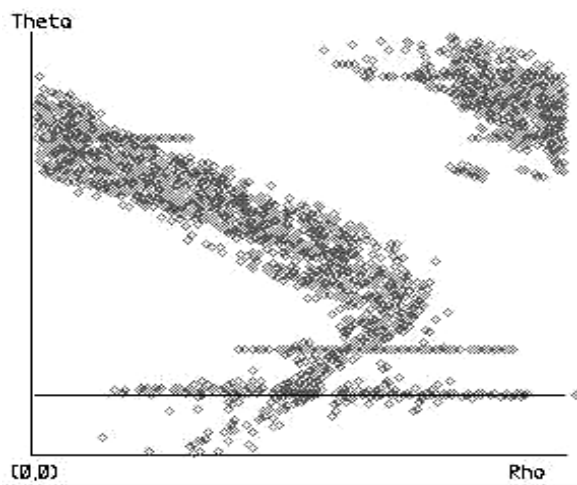
Figure 11: Differences of Dilated Images



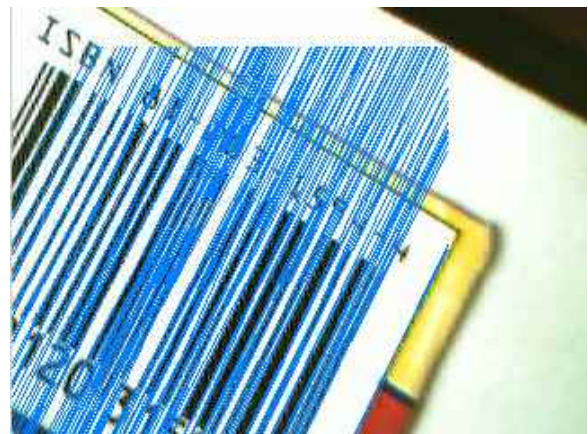


**Figure 12: Barcode Blob After Further Morphological Operations**

Barcode Transformation



**Figure 13: Hough Graph for the Edge Image**



**Figure 14: Detected Hough Lines.**

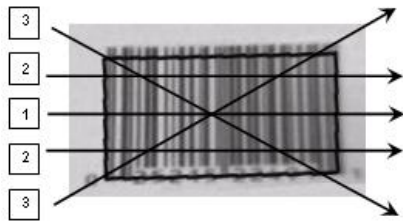


**Figure 15: Original Image**



**Figure 16: Transformed Image**

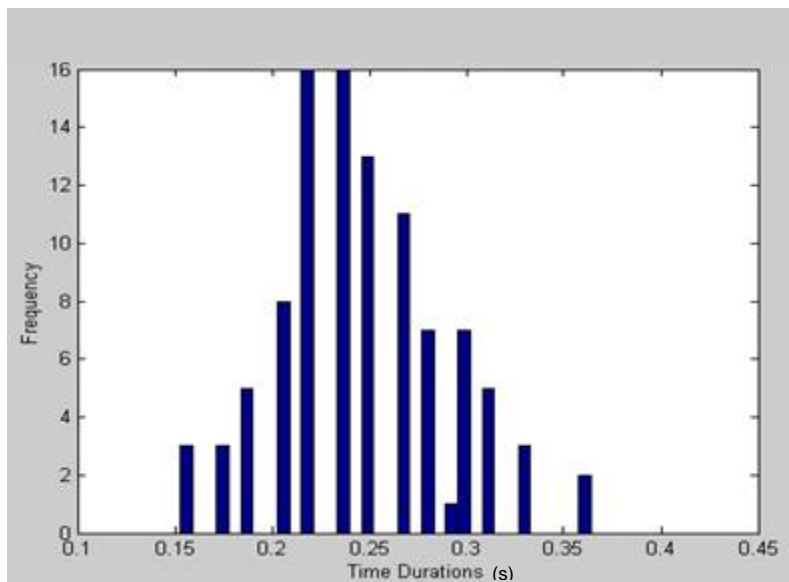
## Barcode Decoding



**Figure 17: Scanned Direction for Decoding**

## Performance Results

Our real-time implementation of *SmartCamBCR* was tested on a personal computer having Intel Celeron 1.7 GHz processor and 256MB RAM. Testing was carried out 100 times, in different backgrounds as well as different barcode orientations. Test data set was consisted of 45 different barcodes.



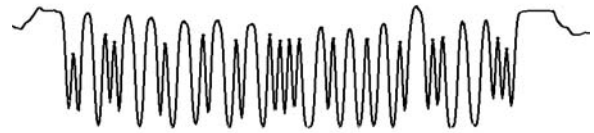
**Figure 19: Frequency Vs Elapsed Time To Read A Barcode.**

Testing gave the results of,

$$\mu = 0.245 \text{ seconds}$$

$$\sigma = 0.0434 \text{ seconds}$$

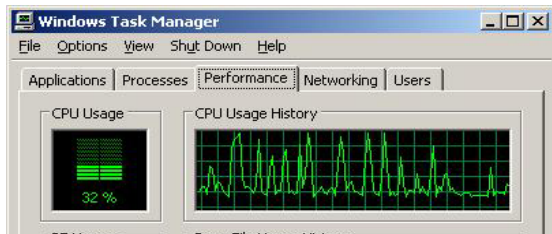
At the peak level it consumes a mean time of 0.245 seconds to decode a barcode. As mentioned in the real-time implementation, this does **NOT** suggest that the frame rate is constant at 4 fps. System reaches the frame rate of 4 fps only at the peak level, but will be staying in the range 15 to 30 fps most of the time.



**Figure 18: Generated Barcode Waveform for Decoding Using Cubic Splines.**

## Processor Utilization

Processor Utilization is monitored on the same platform having Windows XP operating system.



**Figure 20: Processor Utilization for the Real-Time Implementation.**

Each peak in the 'CPU Usage History' corresponds to a reading of a barcode. A higher peak corresponds to 'Full Search' and a lower peak corresponds to 'Partial Search' states.

## References:

- 1 ANDO, S.; HONTANI, H. Automatic visual searching and reading of barcodes in 3-D scene Vehicle Electronics Conference, 2001. IVEC 2001. Proceedings of the IEEE International , 2001 P. 49.
- 2 MUNIZ, R.; JUNCO, L.; OTERO, A. A robust software barcode reader using the Hough transform Information Intelligence and Systems, 1999. Proceedings. 1999 International Conference on , 1999 P. 313.
- 3 SHU-JEN LIU; HONG-YUAN LIAO; LIANG-HUA CHEN; HSIAO-RONG TYAN; JUN-WEI HSIEH Camera-based Bar Code Recognition System Using Neural Net Neural Networks. IJCNN '93-Nagoya. Proceedings of 1993 International Joint Conference on 1993: 2 P. 1301.
- 4 HOWLETT, R.J.; BERTHIER, S.; AWCOCK, G.J. Determining the location of industrial bar-codes using neural networks Image Processing and Its Applications, Sixth International Conference 1997 : 2 P. 511.
- 5 JAIN, A.K.; CHEN, Y. Bar code localization using texture analysis Document Analysis and Recognition, Proceedings of the Second International Conference 1993 P. 41.
- 6 ARNOULD, S.; AWCOCK, G.J.; THOMAS, R. Remote bar-code localization using mathematical morphology Image Processing and Its Applications. Seventh International Conference 1999 : 2 P. 642.
- 7 SHELLHAMMER, S.J.; GOREN, D.P.; PAVLIDIS, T. Novel signal-processing techniques in barcode scanning IEEE Robotics & Automation Magazine 1999 : 6 P. 57.

- 8 TURIN, W.; BOIE, R.A. Bar code recovery via the EM algorithm Signal Processing, IEEE Transactions 1998 : 46 P. 354.
- 9 BOIE, R.A.; TURIN, W. Noise-limited reading of bar codes Industrial Electronics, IEEE Transactions 1997 : 44 P. 816.
- 10 JOSEPH, E.; PAVLIDIS, T. Bar code waveform recognition using peak locations Pattern Analysis and Machine Intelligence, IEEE Transactions, 1994 : 16 P. 630.
- 11 SOMMERVILLE, I. Software Engineering ISBN 81-7808-497-X 2000 : 6.
- 12 PARMER, N. Drowsy Driver Detection System Machine Vision System Design Project Report, 2002.