A Robust Software Barcode Reader using the Hough Transform

Rubén Muñiz, Luis Junco, Adolfo Otero
Dept. of Computer Science. University of Oviedo (SPAIN)
E-mail: {rubenms, junco, otero}@lsi.uniovi.es

Abstract

Nowadays, barcodes are used in many different applications and environments. For most applications, such as access control, price calculation, etc., a handheld scanner is enough, but in other environments where the volume of information is very high and time is critical, hardware scanners aren 't the best choice. In such situations, a powerjul software barcode reuder can process the barcode readers present in a scanned document without human interaction.

The most commonly approach used to implement this kind of software scanners is to simulate the handheld scanner behavior by tracing one or more lines (the hardware laser beam) and measure the width of the barcode's lines and spaces. These methods present a serious handicap, since they are highly sensitive to eventual noise (human signatures, marks) that can be present in a code.

In this paper we present a method based on the Hough transform which solves the problem mentioned before, and that can be easily adapted to read any I-D barcode.

1. Introduction

Linear barcodes are one of the oldest technologies related to the Automatic Identification and Data Capture (AIDC). We are all familiar with the basic barcode we can find in any product in a supermarket. This barcode is called UPC/EAN/JAN and is one variation of over 250 barcodes that have been designed just over time. Barcodes like this are referred to as linear barcodes because they are made up of a collection of bars and spaces frequently known as *elements* or *modules.* Fortunately many of these barcodes have never gained broad acceptance and we usually only consider 1 O-12 linear barcodes. The most common examples in use today are: UPC/EAN/JAN, Code 128, Code 39, Code 93, and Interleaved 2 of 5. Typical data content capacity varies from 8 to 30 characters with some barcodes restricted to numerals only, and others using full alphanumeric information.

Linear barcodes are used in many applications where the use of a simple numeric or alphanumeric code can provide the key to a database of "products." The most obvious limitation is the amount of data that can be stored in a linear barcode. Other problems arise from recognizing low quality printing (low contrast or poor ink receptivity).

For most applications, a manually operated laser scanner is used, but in many others, the volume of information to process is so high that an automated approach has to be implemented. An example of such environment can be found in our drugstores. When a patient goes to a drugstore to buy a prescribed medicine, the pharmacist attach the medicine's barcode in the reserved space for this purpose, and places the drugstore's mark and his signature. Then, the prescriptions are sent to a local organization, which processes all the prescriptions coming from each drugstore in the province. This organization, depending on the country zone, has to process 1-5 millions of prescriptions each month. For each prescription, there are three different kinds of barcodes, a bidimensional one (PDF 417) which identify the patient, and one EAN13 and one Code39 codes which are used, respectively to compute the medicine price and to identify the prescription serial number.

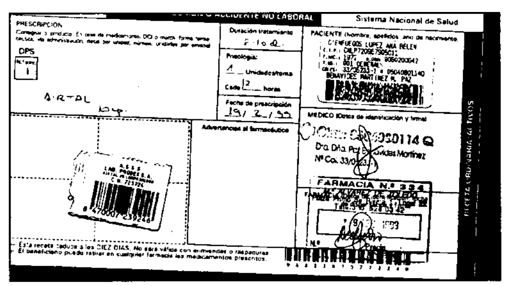

Figure 1. A scanned Spanish prescription

2. The automated approach

As we have mentioned before, in many applications it is necessary to process huge amounts of barcodes in a little time. We have also introduced the prescription's problem present in Spain. For example in the province of Asturias, it has to be processed about 1.500.000 prescriptions every month. Each prescription contains information (medicine's prize, patient and doctor identification numbers and prescription serial number) that will be stored in a database and sent to the administration. There are two ways for getting this information:

1) Using a handheld scanner operated by a dedicated worker.
2) Using a document scanner which scans each prescription and creates an entry in a database table. This record will be used by software barcode readers to access the image which contains the document and, once processed, to store the required information in the adequate table position.

Obviously, the second approach requires less human work, apart from the scanning process. The main handicap of this solution is that recognition software is not perfect and it will introduce a certain number of errors. These errors depend on a certain factors, such as: rotation of the prescription and / or codes, bad scan quality and human-introduced artifacts (signatures, marks, etc.). Under this conditions the software reader has to be robust enough to successfully read the codes. If we use the common approach of simulating the laser scanner by tracing lines (like the scanner's laser beam), at a certain number of angles we will find that a noisy or a rotated code (Fig. 2) can make the software fails.

In the following lines, we will present a robust method based on the Hough transform [1], as well as experimental results obtained using our method.

3. A simple but noise-sensitive method

When we want to face a barcode decoding problem, the first method we can think of is based in the work of a laser beam scanner. This device measures the width of the elements of the code by tracing a red line across it. This approach can be easily adapted to be used in a computer program. To do this it is only necessary to define an array and to "trace" normal lines to the code bars. When the line intersects with a black pixel, a "1" is put in the corresponding array position; when a white pixel is detected, we will put a "0" in that position (or nothing if the array was initialized with zero values). In the following figures we will show a sample barcode and the array plot obtained by using this simple method.

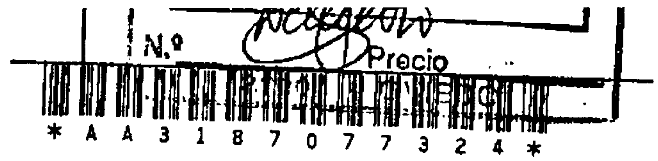

Figure 2. Noisy barcode

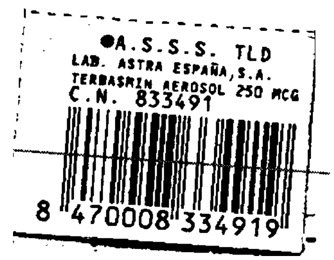

Figure 3. An EAN-13 barcode and the line used to measure the elements' width

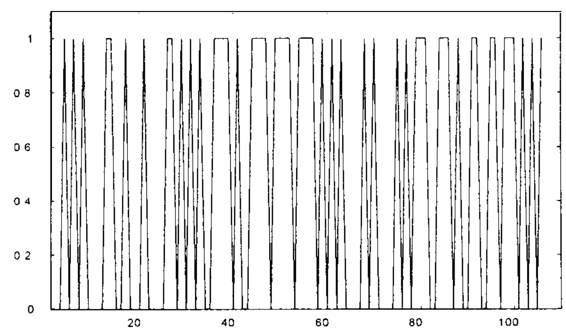

Figure 4. Array resulting of applying the simple method to the previous barcode

It can be easily seen, that the previous example was very easy to solve, since the code is completely clean, without non-desired artifacts. This situation would change if we wanted to decode the Code 39 barcode showed in figure 2. In that case, the code is affected by noise that leads the presented method to a wrong decoding.

In some cases, this handicap can be avoided by tracing more than one line across the code, so the resulting arrays can be combined to allow a successful reading. But if the amount of noise present in the code is too high, this approach will fail too.

4. A robust method based on the Hough transform

4.1 The hough transform

The Hough transform [I] [2][3] is a standard tool in image segmentation that allows recognition of global patterns in an image space by identification of local patterns in a transformed parameter space. It is particulary useful when the patterns we are looking for are sparsely digitized, have "holes" and/or the pictures are noisy.

The basic idea of this technique is to find curves that can be parameterized like straight lines, polynomials, circles, etc., in a suitable parameter space. Although the transform can be used in higher dimensions, the main use is in two dimensions to find, for example, straight lines, centers of circles with a fixed radius, parabolas, etc.

As an example, let's consider the detection of straight lines in an image. To be able to detect vertical lines, the line equation is usually given in its normal form: $\rho = x\cos\theta + y sin\theta$, where ρ is the distance from the origin and θ the angle with the normal. Using this formulation, A4 collinear points lying on a line $x\cos\theta_i + y\sin\theta_i = \rho_i$ yields M sinusoidal curves that intersect at (ρ_i, θ_j) in the parameter space.

4.2 Algorithm for finding straight lines using the Hough transform

One simple algorithm for computing the Hough transform [3][4] uses a 2-D matrix with θ and ρ as its rows and columns, called the accumulator matrix, or simply the A-matrix. For a binary image, we have to process all the pixels one by one. For each non-zero pixel (x_i, y_i), we obtain a finite number of (θ_i, ρ_i) values, determined by the dimensions of the A-matrix. For each point (θ_i, ρ_i) in the Hough plane, the corresponding A-matrix cell is incremented by one unit. When all pixels are processed, the accumulator matrix contains the number of collinear pixels for all the lines found in the image.

4.3 Barcode decoding

If we apply the Hough transform to a sub-image which only contains a barcode (figure 6a), and we plot the A-matrix using a 3D representation, the result will be the one showed in figure 6b.

Figure 6a. Barcode subimage

In that figure, it can be seen the result of applying the Hough transform to the image showed in figure 6a. The A-matrix dimensions were [92 - 94] degrees for the θ dimension and [-900 – 900] for the ρ dimension. The maximum line height found was -60 pixels.

If we fix a value θ_b as the best orientation for analyzing the barcode and we proceed to plot the corresponding A-matrix vector we will get the function showed in figure 6c.

Then, the next step to perform is to analyze the mentioned function in order to decode the represented barcode. The basic idea of the analysis is always the same for linear barcodes, but small modifications will be required for each case. In the following lines we will explain this process adapted to the decoding of a Code 39 barcode.

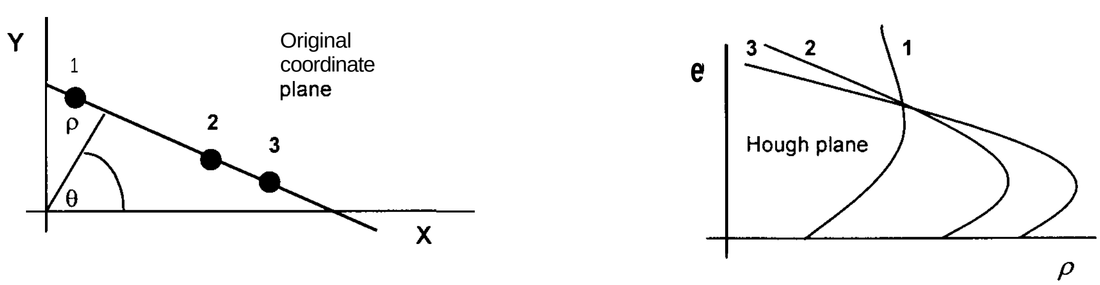

Figure 5. Original and Hough plane

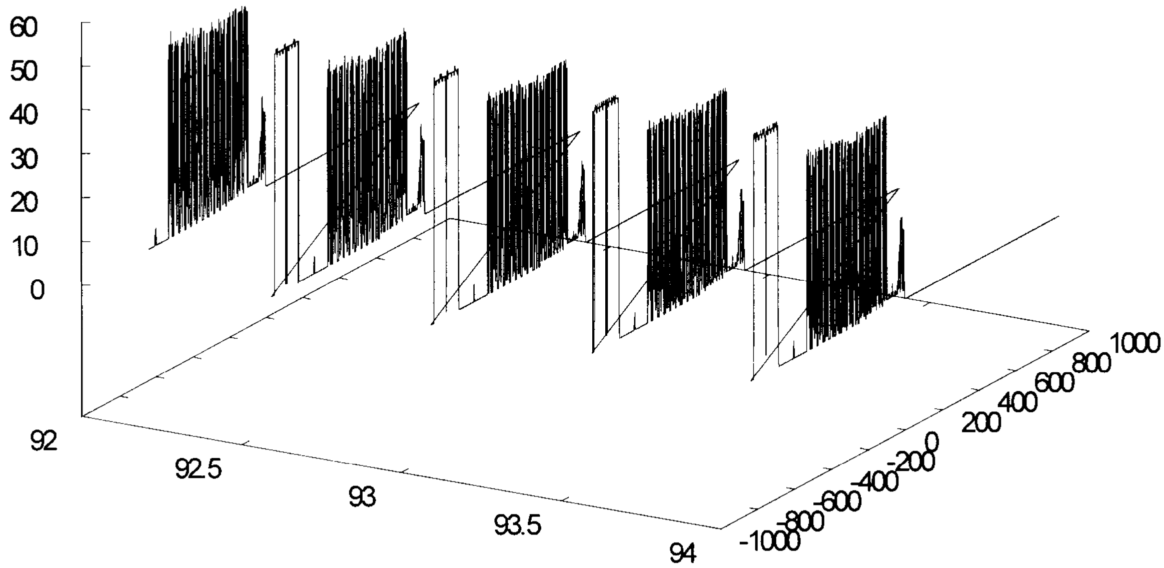

Figure 6b. 3D plot of the A-matrix for a Code 39 barcode

5. Code 39 barcode

5.1 Structure

Code **39** is an alphanumeric barcode [5] that can encode decimal numbers, the upper case alphabet, and the following special symbols:

- * $ / % +

Code 39 characters are constructed using nine elements, five bars and four spaces. Two of the bars and one of the spaces are wider that the rest. Wide elements represent binary ones (1), and narrow elements represent binary zeros (0).

To enable a decoder to distinguish between the wide and narrow elements, a minimum wide to narrow ratio is needed. Depending on which resolution has been used for the printing of barcode, the width of the wide element should be at least two times greater than the narrow element. A ratio of three to one is better for recognizing purposes. All elements of the same type should be printed the same size (the width of a narrow bar should be the same as a narrow space).

5.2 Decoding

If we zoom a region of the plot presented in figure 6c to see the shape of the three first characters (*AB), the barcode function will appear as in figure 7. In this picture it can be observed something obvious: the wider a character bar is, the more contiguous parallel lines it is composed of. This can be easily seen if we realize that this plot represents the lines detected at a given angle, that is, the X coordinate is the line's distance to the origin, and the Y coordinate is the line's length.

As we have said before, a Code 39 barcode is composed of five bars and four spaces. Two of the bars are wider than the other three, and one space is wider than the rest. This allow us to perform a very simple but effective analysis over the previous function. This process can be described as follows:

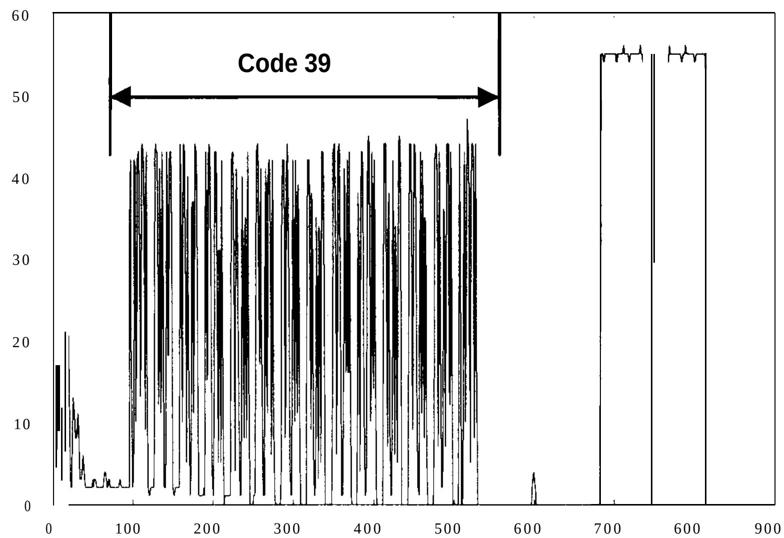

Figure 6c. Plot of a column of the A-matrix for a Code 39 barcode

1) Select a threshold value that will divide the function in two areas. This selection can be done, either fixing a value a *priori* or trying with a range of values until the following steps give us a right solution, i.e., the barcode is properly decoded.
2) For each value less than the threshold, they will be replaced by a zero value. With this step the spaces are lowered to zero and, consequently, the eventual noise present in the barcode is removed.
3) Once the previous step is done, the function will present only zeros and positive values, which are, respectively, the bars of the barcode. Now we have to determine, for each character, the wide bars and space.
4) In order to determine the wide space of a character, we only need to count the number of zeros among its bars. The space with the greatest number of zeros, if unique, will be selected as the narrow one.
5) For determining the wide bars of a character, a similar process to the previous step can be applied. If we sum the values of the function for each bar, we will get a five-element vector. If we take a closer look to this vector, we will find that there are two maximum values which correspond to the two wide bars. The justification for this approach is that the fact which makes the difference between a wide and a narrow bar is the bar area. This can be easily determined by summing the lengths of the lines which compose the bar. In figures 8a and 8b, a plot of a '*' character is shown, which is coded as a '010010100' sequence. In the point plot figure of the character, it can be observed that the wider bars are composed of three large lines (more that 40 pixels long) while the other bars only contains one or two lines.
6) Finally we only have to decode the character using the information about the width of the bars and spaces previously collected.

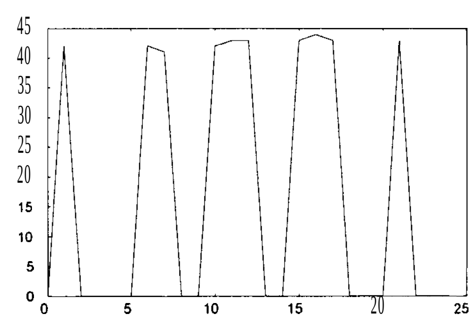

Figure 7. Plot of the function of the three first characters of a Code 39

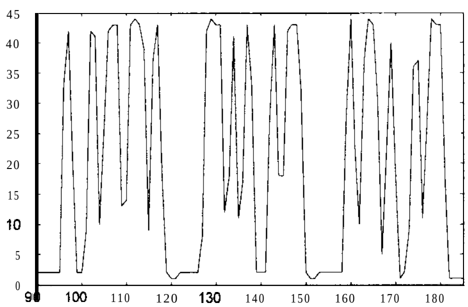

Figure 8a. Line plot of a '*' character

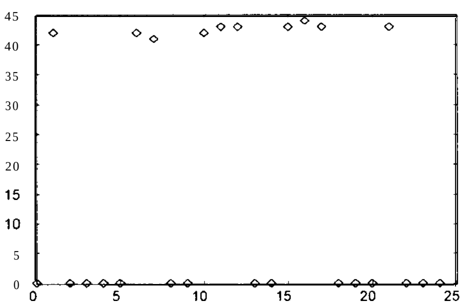

Figure 8b. Point plot of a '*' character

6. Results

Nowadays, there is only one system available in Spain which can be applied to the prescriptions' problem. This system has been developed by a local company, and it is nothing more than a SQL Server front-end which uses the SDK provided by Atxel Inc. [6] for the barcode decoding purposes.

Our system was implemented and tested in two different stages. First of all, we developed an interactive tool that allowed us to load an image (prescription) and see the output of our decoding routine (the readed barcode). Axtel, provides a demo tool, based on their SDK, similar, but much more complete than ours, which is showed in the following figure with an example of a noisy barcode. Note that this tool was not able to decode the entire barcode.

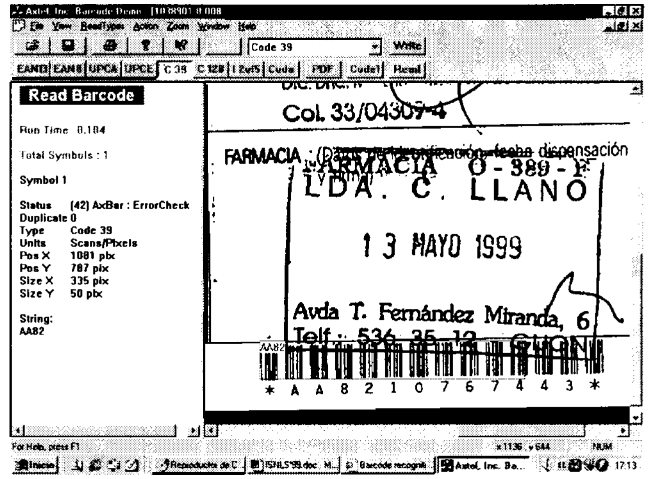

Figure 9. Axtel decoding software screenshot

On the other hand, our method is able to properly decode that kind of code, as it is showed in figure 10.

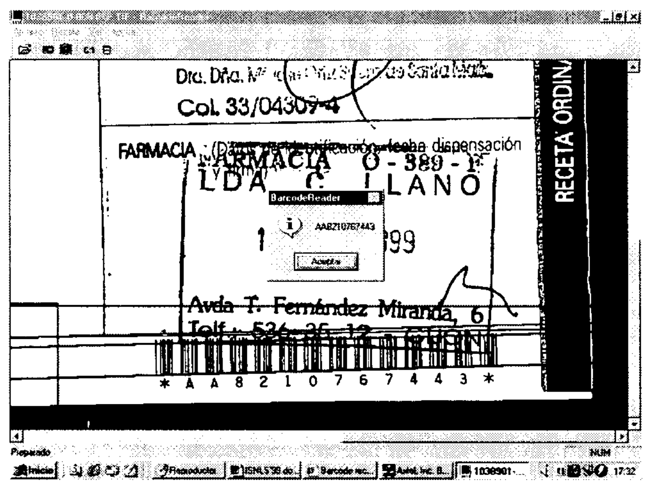

Figure IO. Our own interactive decoding tool

Once our decoding routine was successfully tested on a large set of images, we proceeded to implement a batch tool, which processes the prescriptions automatically. With this new application, we selected a sample set composed of 23038 prescriptions for decoding its Code 39: 2135 1 barcodes were successfully read, while 1687 were not. This gave us a hit rate of 92,6%, while the Axtel-based application could only decode half of the images.

Nowadays, our system is used to process 1,500,000 prescriptions each month. First, the Axtel application is applied, which successfully decodes a half of them. After that, our system is able to process about 70-80% of the remaining prescriptions.

In addition, our method has been recently adapted for EAN-13 decoding and it is currently being tested. We can not provide accurate results yet, but they are very close to the results obtained for Code 39.

7. Conclusions

In this paper we have showed an alternative method to decode linear barcodes using the Hough transform. It has been successfully applied to solve a concrete problem: the reading of Code 39 and EAN-13 barcodes present in Spanish prescriptions. It has been also compared with a comercial system based on Axtel's SDK, which is a standard in the software barcode readers domain.

8. References

[1] Hough, P. V. C., “A Method and Means for Recognizing Complex Patterns“, U.S. Patent No. 3,069,654, 1962

[2] Duda, R.O., and P.E. Hart, “Use of the Hough Transform to Detect Lines and Curves in Pictures”, Communications of the ACM, vol. 15, 1972, pp. 1 l-l 5

[3] O'Gorman, F., and M.B. Clowes, “Finding Picture Edges through Collinearity of Feature Points”, IEEE Transactions on Computers, vol. C-25, 1976, pp. 449-454

[4] Haralick R.M., Shapiro L.G., “Computer and Robot Vision”, Addison Wesley, 1993

[5] http://www.barcodeusa.com

[6] http://www.axtel.com