Decoding Bar Codes from Human-Readable Characters

Kjersti Aas Line Eikvil

Norwegian Computing Center, P.O. Box 114 Blindern, N-0314 Oslo, Norway

e-mail: Kjersti.Aas@nr.no Tel: (+47) 22 85 25 00 Fax: (+47) 22 69 76 60

Abstract

In this study we have decoded bar codes by recognizing the human-readable characters of the interpretation line printed below the bar pattern. Using this approach, we were able to successfully decode bar codes with a resolution of 0.8 pixels per module.

Keywords: Bar codes, CCD images, character recognition.

Decoding Bar Codes from Human-Readable Characters

1 Introduction

Bar codes are an automatic identification technology which have been widely used for the last two decades. The typical input devices for bar codes are 1-dimensional CCD or laser scanners, and manual scanning of product data in supermarkets has been the most widespread application. Today bar codes are used in a number of applications like handling of returned goods, sorting of letters, processing of giros, and registration of forms from the public. In an increasing amount of these applications, image analysis systems are becoming an important part for tasks like optical character recognition, object classification, and quality control. Hence, it would be profitable if the bar code could be decoded directly from the image captured in the image analysis system.

The image analysis systems will typically require that the entire object is contained in the image, and for speed it is necessary that all operations can be performed on the same image. The bar code should therefore also be decoded from this image, and this restriction puts an upper bound on the resolution which can be obtained for the bar code. This creates a need for methods that can decode bar codes of modest to low resolution from 2D images.

Viard-Gaudin and Normand (1993, 1994) have described a system for a postal application of bar code reading from grey level images. Here, the bar code is first located by analysing the gradients within the image, and then the bar pattern is projected onto a line to create a 1-dimensional signal. This signal is decoded by examining the zero-crossings of its second derivative. A blur model is used to compensate for incorrect focusing and low resolution. Their approach is reported to give successful decoding of bar codes for resolutions down to 1.5 pixels per module.

Obviously, all bar code symbologies contain a bar pattern. However, almost all bar codes do also contain an interpretation line where the information encoded in the bar pattern is printed in human-readable characters beneath the bars. Bar patterns are specially designed for reading by 1-dimensional scanners, but when decoding bar codes from low resolution grey level images, it is no longer obvious that the bar pattern is the optimal information source. The human-readable characters may be less effected by blur and low resolution, and may be an alternative or an additional source of information. In this study we have therefore looked into the problem of segmentation and decoding of the human-readable characters beneath the bar pattern.

2 Bar codes

Different bar code schemes, or symbologies, exist. Two common symbologies are UPC (Universal Product Code) which is used in the USA and Canada, and EAN (European Article Number) which is used in the rest of the world. These are the symbologies used on all retail products, newspapers, magazines and books. Other symbologies are used in other markets like health care, postal service and automotive industry. In our study we were interested in bar codes for retail products. We have looked at the EAN symbology, but EAN and UPC are closely related, as EAN is a superset of UPC.

The EAN bar codes appear in two versions; EAN-8 which is a short form encoding 8 digits and EAN-13 which encodes 13 digits. Bar codes of the EAN-13 standard (see Figure 1, left) contain a two digit country code, a four digit manufacturer number, a five digit product number, and one check digit. The bar pattern explicitly encodes 12 digits, while the 13'th digit is implicitly given through the choice of code patterns. The bars can have several widths which are multiples of a unit width, and each digit in the code is represented by 2 bars and 2 spaces covering a total width of seven unit modules. In addition, the code contains 3 guard patterns covering a total of 11 modules (3+5+3). This gives a total length of 95 modules for the bar pattern. For human interpretation the digits encoded by the bar pattern are, as part of the standard, also printed in the OCR-B font beneath the bars.

3 Data

In our study an important issue was to investigate how well we could succeed in decoding bar codes from images from an inexpensive camera. A monochrome analog CCD camera (VICON VC 2130) was used in combination with a frame grabber (Matrix Comet) and a white diffuse light source.

Resolution, blur and contrast are three important factors in the decoding of bar codes. In our experiments the resolution for the bar patterns varied from 0.7 to 1.9 pixels per module. The distance to the objects was fixed, while the size of the bar codes varied. Hence, we experienced no blur due to out-of-focus bar codes. Still, we got a blurring effect (additional to that caused by the low resolution) as the frame grabber seemed to smooth the signal from the analog CCD. These effects do not appear when using more expensive digital cameras. As the bar codes in our test set were all printed on plane surfaces, variation in contrast within the bar codes was not a problem. However, the overall contrast within the bar codes was quite low as a result of the blurring.

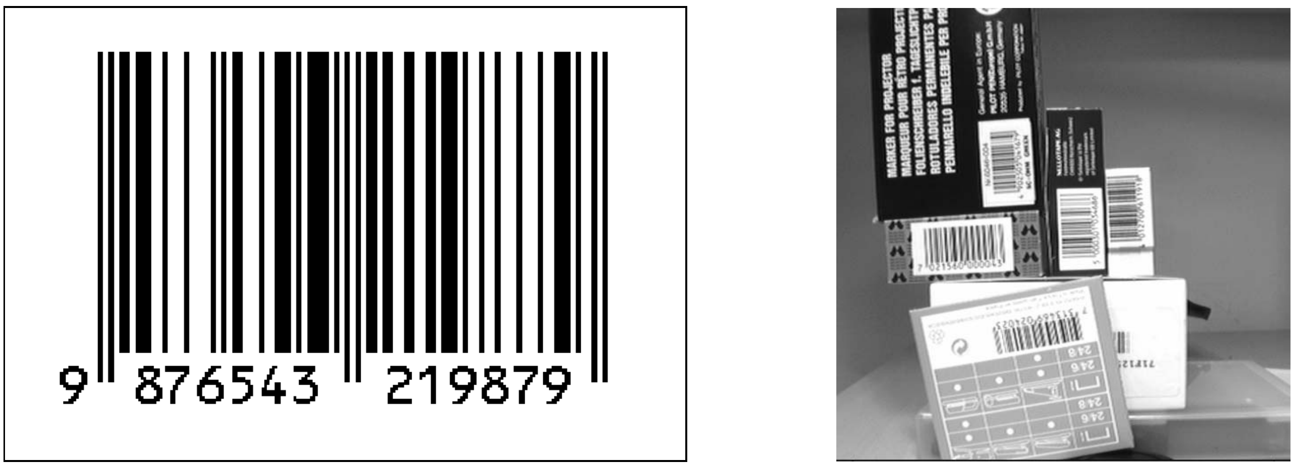

Figure 1: ***Left:*** *Bar code of EAN-13 standard.* ***Right:*** *Original grey level image.*

Our data set consisted of 18 images with 512×512 pixels and 256 grey levels. Each image contained several bar codes, giving a total of 71 codes. The orientation of each bar code was close to either the vertical axis (19 codes) or the horizontal axis (52 codes). All bar codes followed the EAN-13 standard. One of the grey levels images, containing 5 bar codes, is shown in Figure 1, right. Although the data set in our experiments was rather small, it would serve the purpose of indicating whether our method for decoding of bar codes would be feasible.

4 Methods

4.1 Bar code segmentation

The method used for locating the bar codes is based on the work of Viard-Gaudin et. al., and we will therefore only give an outline of the main steps. A more detailed description of the method is available in their paper (1993).

The main idea is to extract areas having a high density of mono-oriented gradients by a texture analysis which is performed directly in the spatial domain. The gradients are found by using the Sobel operator which gives both the magnitude and the direction of the gradients. In our case we know that the code is either close to vertical or horizontal. Hence, we discard all gradients with directions closer to the diagonals than to the vertical or horizontal axes. Weak gradients (low magnitude) are also discarded. Two binary images are constructed, one for the vertical gradients and one for the horizontal gradients, by setting the pixels where the respective gradients appear.

To speed up the further processing, the two binary images are down sampled (Figure 2, left) before they are processed with a morphological filter (Figure 2, middle). The geometry of the resulting connected components is then analysed to select the bar code candidates (Figure 2, right). A rough segmentation is now obtained, and sub images corresponding to the segmented bar codes can be defined. To make sure that no parts of the bar code are lost, the sub images are chosen slightly larger than the circumscribing rectangle of the extracted connected components. Figure 3 (left) shows the sub image corresponding to the connected component marked with grey in Figure 2 (right).

4.2 Bar code positioning

To be able to read the bar code from the sub image resulting from the segmentation process, the exact position and rotation of the bar code within the sub image must be determined.

4.2.1 Estimation of rotation

An initial estimate, a_0, for the rotation angle of the bar code can be obtained from the histogram of the gradient directions (on the interval $[0°, 180°\rangle$) of the extracted sub image. Only gradients with a high magnitude are considered, and a_0 is found as the most frequent angle in the histogram.

A more exact rotation of the bar code is found by identifying the upper and lower margins of the code. Points belonging to these margins are determined as edge points found when moving from the centre of the code toward the margins along lines with slope a_0. By moving in approximately the same direction as the bars, we ensure that the encountered edge will not belong to bar edges within the code.

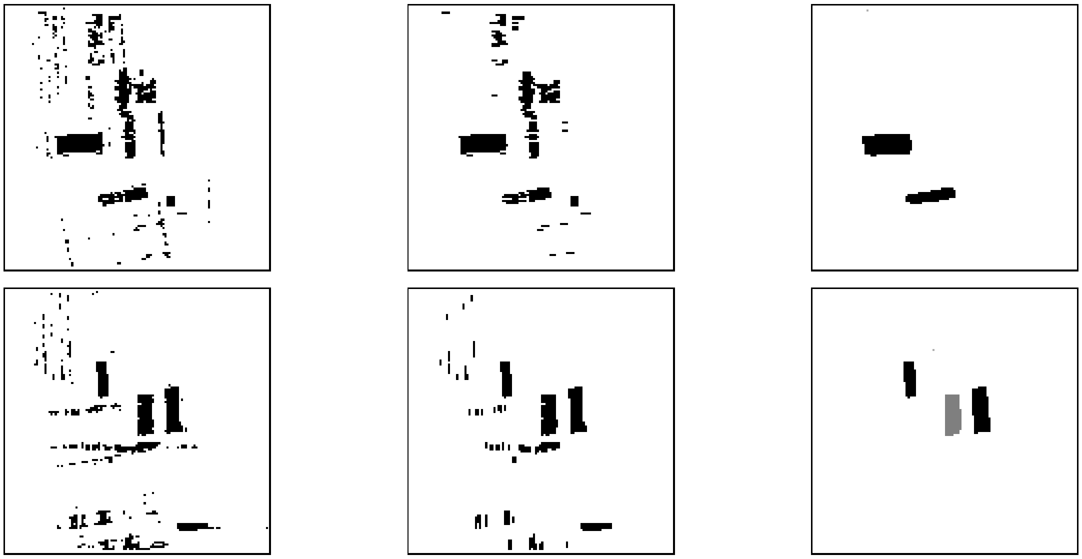

Figure 2: ***Left:*** *Result of gradient quantization and down sampling.*

Middle: *Result of morphological filtering.* ***Right:*** *Extracted connected components.*

A point belonging to the upper (lower) margin is assumed to be found if the difference between the grey levels of the preceding and succeeding points is greater than a threshold, T. In our experiments, T was selected manually, and the same threshold was used for all the images.

For each edge pixel a Gaussian reconstruction filter, described by Seitz (1988), is used to determine the edge to subpixel accuracy. Then, the rotation of the bar codes is estimated by fitting two parallel lines to edge points found along the the upper and lower margin, using the following regression model:

$$y = a_1\, x + b_{11}\, I + b_{12}\, J \tag{1}$$

Here (x, y) are the edge points, I is 1 if the edge point belongs to the upper margin and 0 otherwise, and J is 1 if the edge point belongs to the lower margin and 0 otherwise.

The procedure for extracting points along the margins, results in edge points both on the margin, and outside the margin. Hence, we needed a method which would not be sensitive to outliers. The least median of squares regression (LMS), described by Rosseeuw, has this advantage. It differs from the more common least squares regression in that it minimizes the median and not the sum of the squared residuals. Actually, almost half of the data may be corrupted in an arbitrary fashion and the LMS estimate will still follow the majority of the data. Figure 3 (middle) shows the edge points extracted from the bar code in Figure 3 (left) and the regression lines fitted to these points.

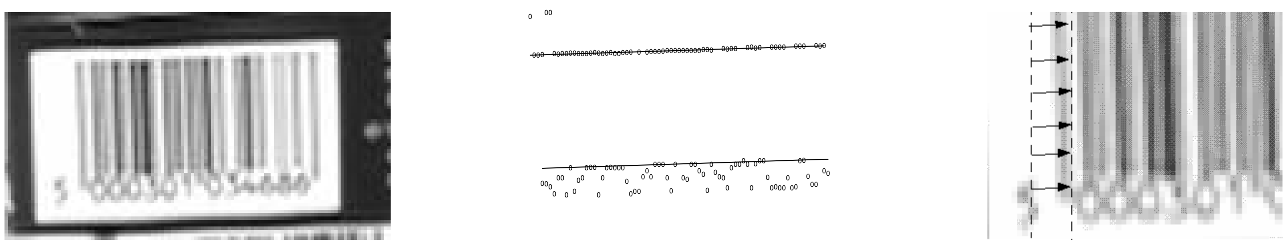

Figure 3: ***Left:*** *The sub image selected for further processing.* ***Middle:*** *Edge points found in the sub image. The fitted regression lines are superimposed.* ***Right:*** *Finding the edge of the left margin.*

4.2.2 Estimation of the bar code width

The width of the bar code is found by estimating the positions for the left and right margin of the code. The equations for the left and right margins of the bar code, are given by:

$$y = a_2\,x + b_{21} \quad \text{and} \quad y = a_2\,x + b_{22} \tag{2}$$

Since the left and right margins are perpendicular to the upper and lower margins, their common slope a_2 is equal to $1/a_1$. Moreover, initial estimates of the upper left, (x_{ul}, y_{ul}), upper right, (x_{ur}, y_{ur}), lower left, (x_{ll}, y_{ll}), and lower right, (x_{lr}, y_{lr}), cornerpoints of the bar code can be obtained from the non-outlier points in 4.2.1. The intercepts of the left and right margins could be derived directly from these cornerpoints. To get more precise estimates, however, we will follow a procedure which is similar to that described in Section 4.2.1.

Moving from left to right along line segments perpendicular to the left margin (as illustrated in Figure 3, right), the pixel values are investigated, and an edge point is said to be found if the difference between the preceding and the succeeding point exceeds a threshold T, where T is the same threshold as used in Section 4.2.1. Hence, we are looking for a sharp transition from light to dark. When such an edge is found, we also require that the preceding lighter field, which is assumed to be the quiet area outside the bar code, is wider than the widest bar in the bar code. Estimates of the bar code width,

w_0, and the widest bar, w_b, can be found from the initial estimates of the corner points of the bar code:

$$w_0 = 1/2 \left[\sqrt{(x_{ur}-x_{ul})^2+(y_{ur}-y_{ul})^2}+\sqrt{(x_{lr}-x_{ll})^2+(y_{lr}-y_{ll})^2}\right] \qquad \text{and} \qquad w_b = \frac{4}{95}\, w_0 \tag{3}$$

The procedure results in a number of edge positions (x_{1i}, y_{1i}) for the left edge of the bar code. Similarly we find points along the right edge of the bar code (x_{2i}, y_{2i}). From these edge points, the estimates $\hat{b}_{21}$ and $\hat{b}_{22}$ for the intercepts b_{21} and b_{22} in the equations for the two margins, can be found as:

$$\hat{b}_{21} = \text{median}(y_{1i} - a_2\, x_{1i}) \qquad \text{and} \qquad \hat{b}_{22} = \text{median}(y_{2i} - a_2\, x_{2i})$$

From the equations for the margins of the bar code, we can compute the width of the code, w_t:

$$w_t = \begin{cases} b_{22} - b_{21} & \text{if } a_1 = 0 \\ a_1\,(b_{22} - b_{21}) & \text{otherwise} \end{cases} \tag{4}$$

We can also compute the exact corner points of the code.

4.3 Bar code reading

As we use an approach where the human-readable characters beneath the bar pattern are the basis for the decoding, we need to locate and segment each of the digits. The following sections describe the methods used for segmentation and classification of these digits.

4.3.1 Determination of the digit box

When the bar code is located as described in the previous section, the exact position of the box circumscribing the human-readable digits must be determined. To find the position of this digit box, we will use information derived from the bar code standard.

For EAN-13 bar codes the distance from the start of the first digit to the left margin of the bar code is approximately 0.125 times the width of the bar code. We also know that for the OCR-B font, in which the digits are printed, the height of a digit, h_d, is approximately 1.25 times the width, w_d, $h_d = 1.25\, w_d$. From Section 2 we have that each digit covers a width of seven modules and that the total width, w_t, of the bar code is 95 modules. Hence, the width of one digit is $w_d = \frac{7}{95}\, w_t$, where w_t is given by Equation 4.

The orientation of the bar code has at this point only been determined on the interval $[0°, 180°\rangle$, and therefore we do not not yet know whether to look for the digits beneath or above the bar pattern. We solve this problem by extracting and analysing two candidate boxes of dimension $1.125w_t \times h_d$; one on each side of the bar pattern. Of the two candidate boxes, one will contain the digits, while the other one will contain background from outside the bar code. It is thereby reasonable to assume that the box containing the digits has larger grey level variance than the empty box. Hence, the candidate with the greatest mean local standard deviation is chosen as the correct digit box.

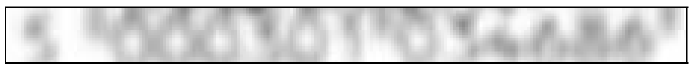

Figure 4: *The digit box which was extracted from the sub image in Figure 3 (left).*

Within the extracted digit box the exact start and end of the digit sequence are found by computing the variance for each column of the box and thresholding the resulting 1-dimensional signal by Otsu's method. The columns of the image that are classified to foreground (high variance) are assumed to correspond to the digits. The first transition from background to foreground is assumed to be the start of the digit box, while the transition from foreground to background close to the right margin of the bar code is taken to be the end of the digit box. Finally, the digit box is resampled by bilinear interpolation to a fixed height and width. Figure 4 shows the digit box extracted from the sub image in Figure 3 (left).

4.3.2 Segmentation of single digits

When the digit box has been determined, it should be split into smaller boxes containing only one digit. From the definition of an EAN-13 bar code, we know the relative position of each digit within the box. Ideally, this should be enough to perform the segmentation, but as the extraction of the digit box is usually not sufficiently accurate, we will only use these positions as initial estimates for the split points.

To improve the estimates, we use the assumption that the average grey level along the columns of the digit box will be higher between two digits than within a digit. This is reasonable because the background is light while the digits are dark. A 1-dimensional signal is therefore obtained from the sum of grey levels along each column of the digit box. This signal is smoothed by a linear filter, and the maximums of the smoothed signal are determined.

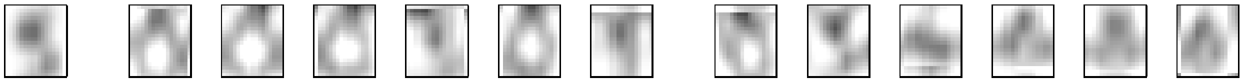

Figure 5: *The segmented digits.*

The final segmentation is determined by going back to the initial split points, and for each split point locating the closest maximum of the smoothed signal and choosing this as the revised split point. The single digits resulting from the segmentation of the box in Figure 4 are shown in Figure 5.

4.3.3 Normalization of digits

Due to blurring, images with low resolution tend to have a greater mean value and a smaller standard deviation than images with high resolution. Therefore the digit images are normalized to a fixed mean and standard deviation before they are classified. There may also be some displacement of the digits within the sub images. This problem is solved by computing the centroid of the grey levels in the sub image and shifting the image if the centroid is not equal to the centre of the image. The resulting normalized digit image consists of N pixels, which values can be labelled by y_i, $i = 1..N$, and which is input to the classification algorithm. In our study, the size of the normalized images was chosen as 17×20 pixels, giving $N = 340$.

4.3.4 Classification of digits

For the classification, a prototype was constructed for each digit. Each prototype is a statistical description of the image corresponding to the digit class. It consists of the pixel value estimates μ_i, $i = 1..N$, and their standard deviation σ_i, $i = 1..N$, for the corresponding class of digits. For an arbitrary digit image y each measurement (pixel value), y_i, is assumed to be Gaussian distributed with density:

$$f(y_i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left\{ -\frac{(y_i - \mu_i)^2}{2\sigma_i^2} \right\} \tag{5}$$

Maximum likelihood estimates for μ_i and σ_i are obtained from training data.

Assuming that the prototype consists of N independent measurements, the joint density is given by:

$$f(y_1, ..., y_N) = \prod_{i=1}^{N} f(y_i) \tag{6}$$

A Bayesian classification rule was used for classification, i.e. an unknown digit was assigned to the type which maximizes the joint probability in Equation 6. Due to the exponential form of the Gaussian density, working with the natural logarithm of Equation 6 is more convenient. Hence, an unknown digit image y is classified to the class c which minimizes:

$$\sum_{i=1}^{N} \left[\ln(\sigma_i^c) + \left(\frac{y_i - \mu_i^c}{\sigma_i^c} \right)^2 \right] \tag{7}$$

4.3.5 Verification of classification

The digit sequence encoded in an EAN-13 bar code contains a check digit. By computing a check sum it is thereby possible to verify whether a bar code has been correctly decoded. For a sequence of decoded digits $c_i, i = 1..N$, this sum can be computed as:

$$S = \sum_{i=1}^{13} f_i * c_i \qquad \text{where} \qquad f_i = \begin{cases} 1 & \text{if} \quad i \in \{1, 3, 5, 7, 9, 11, 13\} \\ 3 & \text{if} \quad i \in \{2, 4, 6, 8, 10, 12, \} \end{cases} \tag{8}$$

For the decoding to be correct, the sum S should be divisible by 10, otherwise one or more of the digits have been erroneously decoded.

Assuming that only one digit has been erroneously classified, one could try to identify this digit by considering the probabilities associated with the Bayesian classifier. One option, is to choose the digit for which the probability of the best match is the smallest. Another option, is to choose the digit for which the difference between the probabilities of the best and second best match is the smallest. Having identified the one erroneously classified digit, it is straightforward to determine the correct class by using the check sum. If more than one digit is incorrect, none of these approaches will work.

There is, however, another way of dealing with this problem. In an application using bar codes, there will almost always exist a database containing the different bar codes that may appear. Such a database is needed to interpret the numeric codes of the bar code, but it can also be used to find the best match for erroneously decoded bar codes. In the case of errors, the erroneously classified bar code can be compared to the legal codes of the database to determine the closest match. The matching can be performed quite efficiently by first comparing the initial two digits of the code to a set of legal country codes, then the next four digits should be matched against the set of manufacturer codes for the specific country, and finally the last five digits should be matched against the product numbers for the given manufacturer.

5 Experimental Results

The following experiments were carried out on the data set described in Section 3.

5.1 Segmentation of the bar code

The segmentation is the process of identifying the region in the image corresponding to the bar code, as described in Section 4.1. All the 71 bar codes were successfully located in the images, and no non-bar code objects were selected. However, two of the bar codes were not properly segmented, but were split into two separate connected components. This was caused by the severe smoothing of the bars, which resulted in a too large area without gradients within the bar code. However, this may be helped, by allowing close rectangular regions to be merged when the connected components are analysed.

5.2 Recognition of single digits

Our data set contained a total of 897 digits, where the number of digits of each class varied from 261 ("0") to 25 ("8"). When designing the prototypes for each digit, we used all the 71 bar codes of our data set. This means that each prototype was generated from digit images with a resolution varying from 0.7 to 1.9 pixels per module. Hence, although we used the same data set for constructing the prototypes as for the classification, there should be no danger of over-fitting the model to the data.

All digits were classified using the same set of prototypes, and the results of this classification are listed in Table 1 (left). We have arranged the digits into three groups, depending on their resolution R, given in pixels per module. For bar codes with a resolution less than 1 pixel per module we obtained a 94% recognition rate for the single digits, and we were able to decode bar codes with a resolution as

	Single digits			Bar codes		
Resolution	$R < 1.0$	$1.0 < R < 1.2$	$R > 1.2$	$R < 1.0$	$1.0 < R < 1.2$	$R > 1.2$
Nof. samples	312	286	299	24	22	23
Correct rec.	94%	98%	100%	54%	86%	100%

Table 1: *Result of classification of single digits and entire bar codes.*

low as 0.77 pixels per module. About 50% of the errors in the classification were caused by inaccurate segmentation of the digit box or the digits, while the remaining errors were simply due to low contrast and low resolution.

5.3 Recognition of bar codes

For a bar code to be successfully decoded, it is necessary to recognize all the 13 digits of the code. Table 1 (right) summarizes the results when looking at the entire code.

By using the checksum of the bar codes, we were able to identify all the bar codes where one or more digits were erroneously classified. By investigating the probabilities from the Bayesian classifier, we tried to identify the erroneously classified digits. However, none of the approaches were able to identify all the erroneously classified digits. This is probably due to the fact that our prototypes were designed from digits with a large variation in resolution and had a large standard deviation while the distance between the classes was small. A better result might have been obtained by constructing different prototypes for the different resolutions. However, our dataset was too small to be split up in this way.

As previously mentioned, for an application using bar codes, there will usually be a database containing the different bar codes that may appear. Such a database can be used to find the best match for erroneously decoded bar codes. Through experimental simulations we tested how a matching of the bar codes would perform, by using a set of 774 legal codes originating from a genuine product database. This database contained bar codes from 16 countries, 69 manufacturers and 733 products. Some of the bar codes differed only by one digit.

From the set of bar codes from the database, a new set of bar codes was simulated with a probability of $p(e_i) = 0.1$ for exchanging one of the digits with another one. This gave a dataset where the probability of all the 13 digits of the bar code being correct, was $P = [1-p(e_i)]^{13} = 0.25$. Each code of this simulated dataset was matched against the database, and for 93.3% of the bar codes a singular best match was found. For the remaining 6.7% of the bar codes there were more than one possible match. Hence, with a classification rate similar to what we obtained in our experiments, a correct decoding for about 95% of the bar codes should be possible.

6 Summary and Conclusion

In this study we have shown how bar codes can be decoded through recognition of the human-readable characters printed beneath the bar pattern. Viard-Gaudin et. al. have previously reported successful decoding of bar codes with a resolution of 1.5 pixels per module. In our experiments we were able to successfully decode bar codes with a resolution of 0.8 pixels per module. For single digits originating from bar codes with a resolution less than 1 pixel per module, a correct recognition rate of 94% was obtained. For the entire bar code consisting of 13 digits, this will only give a recognition rate of about 45%. However, the erroneously classified bar codes are easily identified by computing their check sum. By introducing a verification approach, where these bar codes are matched against a database of legal bar codes, we have shown that a total recognition rate of at least 95% should be obtainable.

We believe that a digital camera would give less blurring, and that this would make it possible to read bar codes of less resolution than 0.8 pixels per module. There also lies a potential improvement in combining the recognition of the human-readable characters with a decoding of the bar patterns as described by Viard-Gaudin et. al.

References

[1] N. Normand and C. Viard-Gaudin (1994), A Two-Dimensional Bar Code Reader. *Proceedings of 12th IAPR Int. Conf. Pattern Recognition*, 3, 201–203.

[2] C. Viard-Gaudin, N. Normand and D. Barba (1993), A Bar Code Location Algorithm Using a Two-Dimensional Approach. *Proceedings of 2nd ICDAR Int. Conf. Document Analysis and Recognition*, 45–48.

[3] P. Seitz (1988), Optical superresolution using solid-state cameras and digital signal processing. *Optical Engineering*, 27 (7), 535–540.

[4] P. J. Rousseeuw (1984), Least median of squares regression. *Journal of the American Statistical Association*, 79, 871–888.