Reading out 2D Barcode PDF417

Michael Trummer[1] and Joachim Denzler[2]

[1] Friedrich Schiller University of Jena, Chair for Computer Vision, Ernst-Abbe-Platz 2, 07743 Jena, Germany, trummer@informatik.uni-jena.de
[2] Friedrich Schiller University of Jena, Chair for Computer Vision, Ernst-Abbe-Platz 2, 07743 Jena, Germany, denzler@informatik.uni-jena.de

Abstract. Reading devices for 2D barcodes based on laser scanning hardware are widely spread. But for certain applications it is necessary to detect and read out such a barcode by means of an CCD camera. When reading a barcode from an image, special problems, caused by poor printing quality, document condition or reflections, must be handled. This paper shows possibilities to overcome these problems and proposes a clear hierarchical procedure to robustly read out a 2D barcode PDF417 from an adequate camera image.

1 Introduction

Two-dimensional barcodes as PDF417 have gained importance in many applications like identification and data management tasks. One such symbols offers a data capacity up to 2710 digits, 1850 text characters or 1108 bytes, respectively, and allows various error correction levels (cf. specification (AIM 2001)). Since the use of such code symbols is obligatory for certain identification documents (covered by specification (ICAO 2005)), a method has been developed to read out these kind of symbols in cooperation with the Cross Match Technologies GmbH. In the context of this paper, reading out means the image processing task to establish the binary matrix belonging to a barcode symbol, and not to decode the contained information. The source images are taken with a three megapixel CCD camera installed inside of a document reading device with controlled illumination.

Searching publications for relevant previous work points out lots of patents protecting reading devices. Also many commercial internet homepages offering solutions for printing and laser scanning barcode symbols can be found. But, regarding actual procedures for reading out barcode PDF417 from digital images, there is no literature, yet.

The remainder of this paper is organized as follows. Section 2 shows the architecture of a PDF417 symbol in a comprehensive manner. The actual reading procedure is being described in section 3. Practical results are demonstrated in section 4. Section 5 gives a summary of the proposed procedure and of the results.

2 Code Architecture

This sections shows the structure of a PDF417 symbol and introduces necessary terminology. In accordance to (AIM 2001), barcode *symbol* means the whole barcode as shown in Fig. 1.

A PDF417 symbol can be seen as a matrix of *codewords* (Fig. 1 (b)). Each codeword consists of 17 binary elements, the *modules*. Within a codeword, the 17 modules are arranged in four bars (black parts) and four spaces (white parts). Each bar and each space contains one to six modules. A PDF417 symbol is composed of defined *start* and *stop patterns* and the *data region* in between. For illustration of the components see Fig. 1. The last barcode column is always blinded out for privacy purposes.

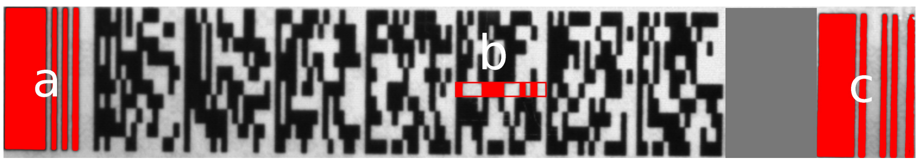

Fig. 1. Example for PDF417 symbol. Start and stop pattern (a,c) and one codeword (b).

The marked codeword (b) in Fig. 1 indicates, what are *rows* and *columns* in a PDF417 barcode symbol (since it is the crossing of a row and a column). The first module of each codeword is black, the last one is white, whereby column separation is guaranteed. To ensure separability of rows, codewords of consecutive rows belong to different subsets of all possible codewords. Thus, it is impossible to have two identical codewords among each other within a PDF417 symbol.

The next section shows how this knowledge about the symbol structure can be used to establish the symbol reader and improve its robustness.

3 Reading Procedure

The characteristics of a symbol's architecture are incorporated into the reading procedure in the following, hierarchical manner. The overall target is to transform the data region into a grid of appropriate dimensions in order to decide for each module by integrating within the respective grid field. Thus, the approach uses the matrix structure of the code symbol.

First, the barcode symbol has to be located within the source image. This is achieved by contour analysis aiming at the start and stop pattern. Within the cut image region (containing the whole barcode symbol), a rectification is performed to yield the rows horizontal and the columns vertical. These properties can then be utilized to analyze the numbers of rows and columns as well as the positions by x- and y-histograms. Finally, this information allows to establish the desired grid within the rectified symbol image and to decide for each grid field, if the underlying module is black or white.

3.1 Finding the Barcode Symbol

The first step performs a contour search and analysis in order to find the start and stop pattern of a PDF417 symbol that consist of two kinds of elements. These are wide (see Fig. 2, a) and thin (Fig. 2, b) rectangles. With respect to runtime efficiency, candidate contours are rated hierarchically. A contour is classified as an element of the start or stop pattern only if the contour has the following features: it is closed; contour length and area are within certain boundaries (depending on resolution parameters); the height-width-ratio is not wider than square; the corresponding region has the overall shape of a rectangle, considering noise and distortion. This last feature is being checked by a fitting method using region moments (Voss and Suesse 1997).

Fig. 2. Finding the barcode symbol by contour analysis. Elements of start and stop pattern marked, wide (a) and thin (b) rectangles.

With the known positions of the start and the stop pattern, the image region containing the barcode symbol can be extracted accurately. Thus, the cut barcode region endows a new image. An interesting characteristic of this approach is that only one element of the start and the stop pattern, respectively, needs to be found to ensure the correct extraction of the symbol region.

3.2 Rectification

The barcode frequently appears slightly rotated in the source image. Further distorting effects can be non-linear distortions caused by some bending of the document, a lamination sheet or poor printing quality. By performing a projective transformation of the actual barcode corners to the image corners of the cut image, rotation can be balanced and the other disturbing effects can be smoothed. This transformation (homography) is calculated from the mappings of four image points (actual corner points of the barcode symbol mapped to image corners) by singular value decomposition as described in (Hartley and Zisserman 2002). Here, the crucial task is to find the actual corner points of the symbol. To do so, at first the convex contour of the barcode symbol is computed. This contour shows a rectangular shape, where corners can be identified by a local corner measure. Given one contour point, the coordinates of the next i points in both directions are observed, and the x- and y-intervals covered by this part of the contour. The size of these intervals is denoted by dx_1, dy_1 for the next contour points in one direction of x, and dx_2, dy_2 for the next points in the other direction. The local corner measure m_x in point x is given by

$$m_x = |dx_1 - dx_2| + |dy_1 - dy_2|. \tag{1}$$

This measure reaches its maximum, if the two contour tails at point x are perpendicular to each other and parallel to the image axes. Evaluating the corner measure from Equation 1 for each contour point yields four clear local maxima along the contour indicating the barcode corners. The number i of contour points to be considered on each side surely depends on the image resolution. But this parameter has proven to be non critical, since a wide range of values yields identical results. For the present application, $i = 25$ has been chosen.

Once the barcode symbol corners are located, it is possible to calculate and to apply a homography mapping the barcode corners to the image corners of the cut barcode image. This yields a new image of the barcode symbol without margins, ideally, and with axially parallel rows and columns of the barcode matrix.

3.3 Detecting Barcode Rows

The rectified image of the PDF417 symbol shows the barcode rows in horizontal orientation. This fact is utilized for detecting the barcode rows.

The main idea is not to detect barcode rows, but the transitions between the rows. This approach is induced by the fact that codewords of consecutive rows belong to different subsets of all possible codewords. Thus, codewords of one column and consecutive rows differ in at least one module. If we consider a vertical scan line leading through these differing modules, the transition between the two rows is being detected as an alternation between black and white (object and background). In order to achieve robust detection, the vertical scan line is moved through the image, and detected alternations are cumulated in an y-histogram (see Fig. 3).

Fig. 3. Y-histogram of detected vertical alternations (left). Detected row transitions (right).

The histogram in Fig. 3 shows clear peaks indicating row transitions. The non zero values between the peaks are due to image noise and interpolation effects associated with the homography. The peak identification is solved by using the method of (Otsu 1979), that is applied to the y-histogram of the histogram in Fig. 3.

After this step, the number of barcode rows is known as well as the positions of the upper and lower row borders.

3.4 Detecting Barcode Columns

The information about column number and positions defines the second dimension of the desired grid for reading out the barcode symbol. The column detection regards the fact that each codeword starts with a black module and ends with white one. For barcode columns follows that each has a stack of black modules on the left and of white modules on the right side. Consequently, the beginning of a barcode column can be identified uniquely. This yields an easy decision rule based on the x-histogram of object pixels (see Fig. 4).

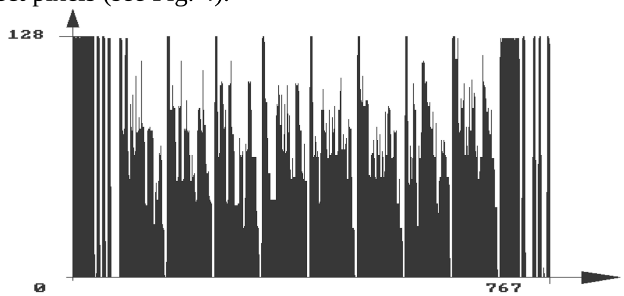

Fig. 4. X-histogram of object pixels.

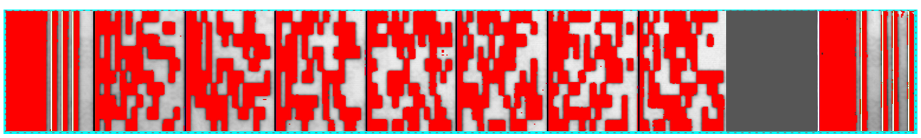

Fig. 5. Candidate positions for starting barcode columns (light) and detected columns (dark).

Exploring the histogram from Fig. 4, each peak preceded by a zero value is a candidate position for a starting column. In practice, the criteria (peak height, “zero” value, distance between “zero” and peak position) have to be softened in order to handle noisy and distorted images. From the list of candidate positions, the actual column positions need to be chosen. First, the maximum distance between two consecutive candidate positions is computed. In accordance to the code architecture, this maximum distance is equal to the column width of the barcode symbol. Starting in the list at the position of this first column found, the further column positions to the left and to the right are computed. The result is shown in Fig.5. The start and stop patterns can be identified as non valid columns (not inside the data region) by the large rectangles.

At this point, column number and positions are available.

3.5 Reading out the Barcode Symbol

At this stage of processing, all the necessary information for aligning a grid with the barcode data region is given. In fact, the borders of each single module can be calculated using the information yielded in former processing steps. Upper and lower module borders are given by the respective row borders. Left and right borders can be calculated from the respective column borders, whereby each column is divided into 17 equal sub-columns (the module stacks). For the example image, the center of each grid field is highlighted in Fig. 6. Each grid field is used to determine the state (back/white) of one module.

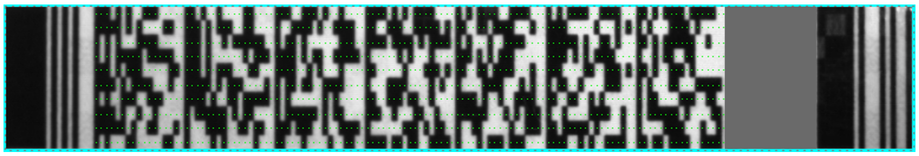

Fig. 6. Rectified barcode image. Centers of grid fields for each module highlighted.

Now, the image pixels inside a grid field are associated with the underlying barcode module. For this, the values of all the pixels within a grid field are taken into account. The intermediate result is a double-value matrix containing the mean values of the grid fields. Due to the mentioned distortions and to discretization, this matrix is in general not binary.

The conclusive step could be a simple binarization. But it is obvious that values close to the threshold are not reliable. Therefore, the interpolated value of the grid field center (cf. Fig. 6) is assigned to grid fields with a mean value close to the binarization threshold. Within the application, grid fields with mean values within 10% above or below interval mean are treated in this way. Afterwards, the final result is given as a binary matrix like in Fig. 7.

Fig. 7. Final result as binary matrix.

In comprehension, an orthogonal grid is established using information computed from the source image (rows, columns). The barcode image is transformed into this grid, and the state of each module is estimated by integrating within each grid field and by deciding with the help of extended binarization (see preceding paragraph).

The next section will give information about testing results.

4 Practical Results

A database of hundreds of real-world barcode images was available for testing and visual inspection. But since this database, lacks ground-truth data and since manual labeling is still under construction, this collection does not allow comprehensive tests, yet. Therefore, *module recognition rate r* (rate of correctly classified binary elements) against image noise has been tested systematically by means of a synthetic barcode image (20 rows, 10 columns, module resolution 3x12 pixels). The image noise on the 8-bit gray values has been modeled as *Gaussian noise with standard deviation* σ. For some images the procedure was unable to detect the correct dimensions (rows, columns) of the barcode symbol. The *percentage of rejected images e* and the recognition results of images not rejected are presented in Table 1 (30 images generated and tested for each value of σ).

σ	10	20	30	40	50	60	70	80	90	100
e (%)	0	0	0	0	0	0	6.67	3.34	13.34	16.67
r (%)	100.00	100.00	100.00	99.97	99.86	99.77	99.58	99.02	98.73	97.68

Table 1. Test Results. Percentage *e* of rejected barcodes. Module recognition rate *r* (within the barcode images not rejected) against Gaussian noise, standard deviation σ .

As a further hint at the procedure's performance, Fig. 8 gives an example of a low quality barcode image and the achieved recognition rate.

Fig. 8. Example with low contrast, geometric distortions and low resolution. *r* = 97.7%

In conclusion, not only the impressions from visual inspection and from the practical application, but the results of systematic tests and of worst case analysis underline the procedure's robustness against the mentioned kinds of distortions.

5 Summary

This paper proposed a procedure that solves the image processing task of reading out a 2D barcode symbol PDF417. It has been demonstrated, how the special knowledge about the code architecture can be used to yield a robust hierarchical solution for this problem. Practical experiments have been carried out to prove the procedure's capability to handle noticeable amounts of image noise, distortions and other inhibiting circumstances. The according software has already been installed on commercial document reading devices.

References

AIM Association for Automatic Identification and Mobility, Uniform Symbology Specification PDF417, ISO/IEC 15438:2001

ICAO International Civil Aviation Organization (2005), Machine Readable Travel Documents, Sixth Edition, ISO/IEC 7501:2005

Voss, K. and Suesse, H. (1997) Invariant Fitting of Planar Objects by Primitives, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 19, no. 1, pp. 80-84

Hartley, R. and Zisserman, A. (2002), Multiple View Geometry in Computer Vision, Second Edition, *Cambridge University Press*, Cambridge, p. 91

Otsu, N. (1979) A threshold selection method from grey level histograms, *IEEE Trans on Systems, Man, and Cybernetics*, vol. 9, pp. 62-66