Locating and Decoding EAN-13 Barcodes from Images Captured by Digital Cameras

Douglas Chai and Florian Hock

Visual Information Processing Research Group
School of Engineering and Mathematics
Edith Cowan University
100 Joondalup Drive, Joondalup WA 6027
Perth, Australia

Abstract – **In this paper, we propose a vision-based technique to locate and decode EAN-13 barcodes from images captured by digital cameras. The ultimate aim of our approach is to enable electronic devices with cameras such as mobile phones and Personal Digital Assistants (PDAs) to act as a barcode reader.**

I. Introduction

In today's modern society, almost every consumer product has a barcode label. With the use of, say, a laser barcode scanner, information about a product such as description and price can be quickly obtained.

Another common item in many parts of the world is mobile phone. No doubt the use of mobile phone is becoming increasingly widespread, and its features also are rapidly growing. We can use mobile phone to access the Internet, and take pictures and videos. Perhaps, we can also use it as a barcode reader.

Consumers can capture an image of a barcode label using their camera phone (i.e. mobile phone with built-in camera). The phone, programmed to interpret barcode images, will generate the barcode value which is used to identify the product. By connecting online, consumers can then get access to a wealth of information about the product. This includes not only product description and price but also product review, price comparison, location of retailers, etc.

Some studies on reading barcode images can be found in [1] and [2].

In this paper, we present an image processing algorithm capable of interpreting a particular type of barcode, called EAN-13, from a digital image.

This paper is organized as follows: a brief explanation of the EAN-13 barcode (from the decoding point of view instead of encoding) is given in Section II; our proposed algorithm for locating and decoding barcode images is presented in Section III; and the concluding remarks can be found in Section IV.

II. The EAN-13 Barcode

EAN is the acronym for European Article Numbering. The EAN-13 barcode is used worldwide for marking retail goods. This barcode has 13 numeric characters, also known as digits. The first 2 or 3 characters are the country code. This is followed by either 9 or 10 characters (depending on the length of the country code) for manufacturer code and product code. The 13th character of the EAN-13 barcode is the checksum digit.

The EAN-13 barcode is a graphical rendering of a 13-digit code in the form of line pattern. An example of the EAN-13 barcode is shown in Figure 1. One can consider such line pattern as being constructed using dark bars and light spaces of various thickness, but for better clarity we simply treat the pattern as alternating black (1) and white (0) bars of various widths (width-1, 2, 3 or 4).

9 319636 431074

Figure 1: Example of an EAN-13 barcode.

The EAN-13 barcode always begins and ends with a black-white-black (101) pattern, which is known as the left hand guard pattern and the right hand guard pattern. There is also a white-black-white-black-white (01010) center guard pattern in the middle of the barcode. These 3 guard patterns are usually rendered with slightly longer vertical bars as can be seen in Figure 1. All bars in these guard patterns are of width-1.

Each digit of the barcode with the exception of the first is represented by a fixed-width bar pattern and is rendered separately from the other digits. Each pattern is consisted of 4 bars, alternating between black and white. Therefore, each pattern is constructed from two black bars and two white bars. Although each bar varies in width, the total width of the 4 bars is fixed and must equal to 7.

The bar patterns for the 2nd, 3rd, 4th, 5th, 6th and 7th digits of the code are grouped together to form the left hand side of the barcode, which is delimited by the left hand guard pattern and the center guard pattern.

Similarly, the bar patterns for the 8th, 9th, 10th, 11th, 12th and 13th digits of the code are grouped together to form the right hand side of the barcode, which is delimited by the center guard pattern and the right hand guard pattern.

When reading from left to right, the bar patterns for digits belonging to the left hand side always start with a white bar and end with a black bar. On the right hand side, however, the patterns always start with a black bar and end with a white bar.

The 1st digit of the barcode value is determined from the parity pattern of the left hand side.

Once all 13 digits are decoded, a checksum can be calculated to ensure that it matches the 13th digit of the barcode value. A mismatch would imply that the barcode was not correctly decoded.

More information on EAN-13 barcode can be found in [3].

III. ALGORITHM DESIGN

The proposed algorithm, as depicted in Figure 2, consists of two main components: the first is to identify the location of the barcode in the given image, while the second is to perform the decoding after the barcode has been located. However, prior to all these, we first pre-process the given input image. An example of input image is shown in Figure 3.

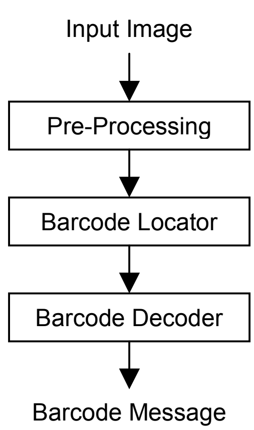


Figure 2: Block diagram of the proposed algorithm.

Figure 3: A 640×480 image captured by a digital camera.

A. Pre-Processing Input Image

Depending on the input image, the following two pre-processing steps may be necessary.

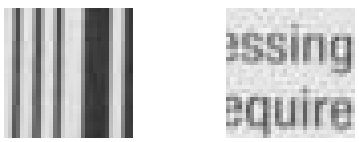
- The algorithm requires only the luminance information as input. Therefore, if the input is not in a grayscale format, then the luminance component will need to be extracted from the input image.
- The algorithm relies on both pixel level and block level processing. The block size of 32×32 pixels is chosen. If required, all four edges of the input image will be trimmed so that the image size is divisible by the block size.

B. Locating Barcode

We have proposed a block-based technique for finding the barcode location. This technique involves the following steps:

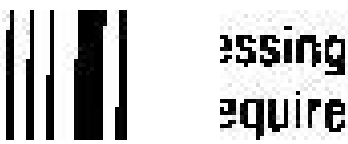
- The input image is divided into non-overlapping blocks of equal size. Figure 4 shows two blocks of size 32×32 pixels of the input image in Figure 3. The block denoted by X1 captures a part of the barcode, while the block denoted by X2 captures a part of the texts surrounding the barcode.
- For each block, the grayscale data is converted into binary using the Otsu thresholding technique [4]. The result is illustrated in Figure 5 using the above X1 and X2 blocks of this input image.
- A morphological operation known as "skeletonizing" [5] is then performed on each block of binary data. This is illustrated in Figure 6.
- Each connected components within each block are extracted and labeled. For instance, 5 connected regions were identified in block X1, and 6 in block X2. These are shown in Figure 7.

- The orientation of each labeled region within each block is calculated and compared in order to test for parallel property. The orientation is measured by the angle between the x-axis and the major axis of the ellipse that has the same 2nd moments as the region. The angles for each labeled regions for blocks X1 and X2 are given in Table 1. If all angles in each block are of similar values, then that block is considered as having a parallel line pattern, and the orientation of the line pattern is defined by the mean angle.
- A group of blocks with similar parallel line patterns would indicate the presence of a barcode. Table 2 shows the location of these line patterns and their orientations.
- Based on this information, the barcode region is then extracted, as illustrated in Figure 8.



(a) (b)

Figure 4: Two 32×32 samples of the input image as shown in Figure 3: (a) Block X1 is located within the barcode image region, and (b) block X2 is located at the surrounding texts.



(a) (b)

Figure 5: The binary representation of blocks X1 and X2.

(a) (b)

Figure 6: Blocks X1 and X2 after skeletonizing.

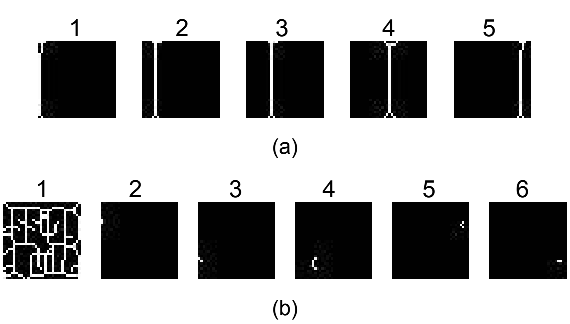


Figure 7: (a) 5 connected components in block X1, and (b) 6 connected components in block X2.

Table 1: Orientation of each labeled region within blocks X1 and X2.

Region	Angles in Block X1 (degrees)	Angles in Block X2 (degrees)
1	88.6336	34.1162
2	89.5410	90.0000
3	89.5410	45.0000
4	89.3953	90.0000
5	88.5934	90.0000
6	-	0.0000
Mean	89.1409	58.1860

Figure 8: The segmented barcode image region.

C. Decoding Barcode

The proposed decoding process is as follows:

- First, it obtains a sample row across the barcode image. This signal is shown in Figure 9.
- The signal is then converted into the binary form by simple thresholding based on the mean value of the signal. The signal in binary form, which is essentially a binary sequence of 1's and 0's, is graphically depicted in Figure 10.
- The run-length encoding is performed on the binary sequence so that the individual width of the alternating black and white bars can be determined.
- The next step is to determine where the barcode begins by searching for the left hand guard pattern.
- Once the left hand guard pattern is found, the decoder will read the next 6 sets of 4 bars to determine the values of the 2nd, 3rd, 4th, 5th, 6th and 7th barcode digits.
- The center guard pattern should be detected before reading the next 6 sets of 4 bars, which specify the values of the 8th, 9th, 10th, 11th, 12th and 13th digits.

Table 2: The location of blocks with parallel line pattern for the input image shown in Figure 3. Entries with zero value imply blocks with no parallel line pattern, while non-zero values refer to the orientation of the line pattern measured in degrees with reference to the horizontal axis.

1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	28
0	0	0	0	0	0	0	0	0	0	0	0	0	45	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	89	89	89	90	90	90	90	90	0	0	0
0	0	0	0	0	0	0	0	0	89	89	90	90	89	90	90	90	0	0	0
0	0	0	0	0	0	0	0	0	89	88	88	89	90	90	90	90	0	0	0
0	0	0	0	0	0	0	0	0	0	0	89	89	90	89	89	90	0	0	0
0	0	0	0	0	0	0	0	0	88	0	0	0	0	0	0	90	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

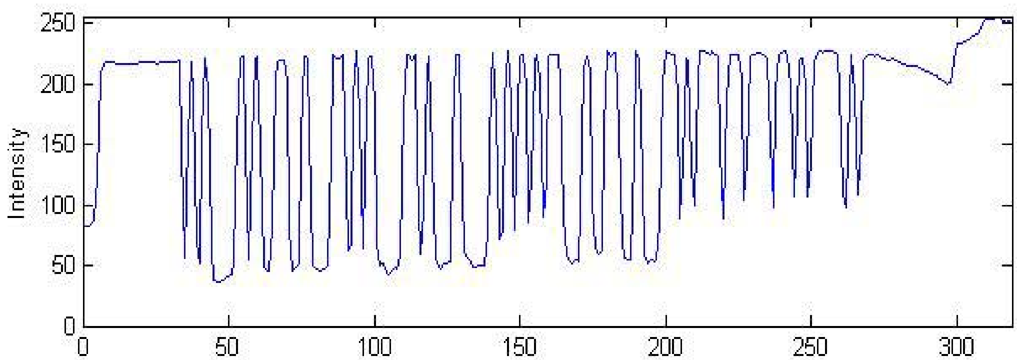


Figure 9: A sample row across the barcode image.

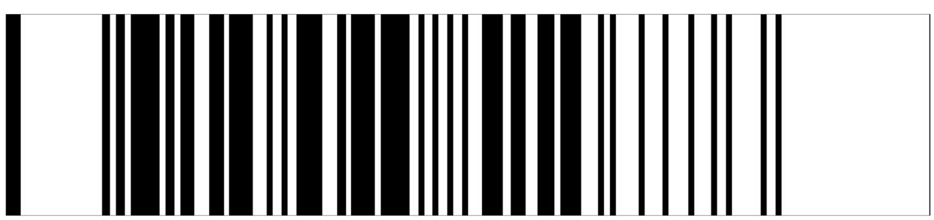


Figure 10: The thresholded sample row.

- The next set of bar pattern which belongs to the right hand guard pattern will verify the end of the barcode.
- Table 3 presents the bar patterns found from the thresholded sample row shown in Figure 10. Note that the bar patterns for the left hand side digits start with a white bar, while the bar patterns for the right hand side digits start with a black bar.
- The 1st digit is determined from the parity pattern of the left hand side. The parity pattern OEEOEO where O and E denote odd and even, respectively, specifies that the 1st digit is 9.
- Therefore, the barcode is 9 315693 510776 for this particular example.

Table 3: The bar patterns found from the thresholded sample row.

Position	Bar Pattern	Digit	Parity
Left hand guard pattern	1 0 1		
2nd digit	0 1 1 1 1 0 1	3	Odd
3th digit	0 1 1 0 0 1 1	1	Even
4th digit	0 1 1 1 0 0 1	5	Even
5th digit	0 1 0 1 1 1 1	6	Odd
6th digit	0 0 1 0 1 1 1	9	Even
7th digit	0 1 1 1 1 0 1	3	Odd
Center guard pattern	0 1 0 1 0		
8th digit	1 0 0 1 1 1 0	5	
9th digit	1 1 0 0 1 1 0	1	
10th digit	1 1 1 0 0 1 0	0	
11th digit	1 0 0 0 1 0 0	7	
12th digit	1 0 0 0 1 0 0	7	
13th digit	1 0 1 0 0 0 0	6	
Right hand guard pattern	1 0 1		

- Lastly as an error checking process, the algorithm calculates and verifies the checksum. This involves multiplying each of the first 12 digits of the barcode with a weight and then summing the results together, i.e.

$$weighted\_sum = \sum_{i=1}^{12} weight(i) \times digit(i)$$

where *weight* = [1 3 1 3 1 3 1 3 1 3 1 3 1 3]. The checksum is the smallest number which, when added to the *weighted_sum*, produces a multiple of 10. In our example, the *weighted_sum* is (9×1) + (3×3) + (1×1) + (5×3) + (6×1) + (9×3) + (3×1) + (5×3) + (1×1) + (0×3) + (7×1) + (7×3) = 114. Therefore the checksum digit is 6, since when added to the *weighted_sum*, results in 120, which is a multiple of 10. The checksum matches the 13th digit of the barcode, and hence validates the barcode value produced by the decoder.

IV. Concluding Remarks

An image processing based approach to locating and decoding EAN-13 barcodes from images captured from digital cameras was presented in this paper. The proposed algorithm identifies parallel line patterns at block level to determine the location of the barcode within the given input image. Once the barcode image region is identified, the decoding process is carried out according to the EAN-13 specification. The techniques proposed in this paper can easily be adapted to other one-dimensional barcodes.

References

[1] R. Muniz, L. Junco and A. Otero, "A robust software barcode reader using the Hough transform," *International Conference on Information Intelligence and Systems*, Bethesda, MD, USA, pp. 313-319, Nov.1999.

[2] E. Ohbuchi, H. Hanaizumi and A.-H. Lim, "Barcode readers using the camera device in mobile phones," *International Conference on Cyberworlds*, Tokyo, Japan, pp. 260-265, Nov. 2004.

[3] http://www.barcodeisland.com

[4] N. Otsu, "A threshold selection method from gray-level histograms," IEEE Transactions on Systems, Man and Cybernetics, vol. 9, no. 1 pp.62-66, 1979.

[5] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Second Edition, Prentice-Hall, Upper Saddle River, New Jersey, 2002.