Computer Vision Applied to Recognition Barcodes

Marcelo de Araújo Almeida and Álvaro Manoel de Souza Soares

Department of Mechanical Engineering, UNITAU—Universidade de Taubaté, Taubaté 12060-440, Brasil

Received: August 29, 2013 / Accepted: September 29, 2013 / Published: November 25, 2013.

Abstract: The bar code is one of the identification technologies used more successfully in the world. This technology was first developed as the second half of the twentieth century and quickly achieved acceptance in both industry and commerce. In this paper studies are discussed on the design, development and printing of one-dimensional barcodes of the symbologies Code 39 and Interleaved 2 of 5. There are also discussed techniques for recovering the information encoded in the bar code, which can be extended to other symbologies. The purpose of the paper is the development of an open solution with the objective to make the decoding of barcodes using computer vision techniques possible, which will be divided into four basic stages: acquisition, pre-processing, processing and presentation of results. Practical tests conducted with samples show satisfactory results, validating the operating principles of the elements involved. The proposed objective was achieved and practical tests validated the operating principles of the elements addressed by providing contribution to that segment decoding barcodes.

Key words: Computer vision, image processing, bar codes, Java language.

1. Introduction

The bar code is one of the identification technologies used more successfully in the world. This technology was first developed as the second half of the twentieth century and quickly achieved acceptance in both industry and commerce. The use of barcodes in the industry reached the point where a piece is marked in the early stages of manufacture and can be traced to its delivery to the customer. Bar codes also made it possible to optimize the processes of separation and transportation of materials, using automated means. This success comes in large part from the easy implementation of the technology and the low cost of the commercial equipment for decoding the information [1]. Although there are newer technologies for electronic identification, such as through radio frequency tags (RFID—radio frequency identification), they still can not be applied to any product and the equipment for both production and decoding of the tags still has a relatively high cost.

Conventional bar code readers can be found in various forms with format-specific applications, such as pistols, pens or query terminals. As a way of illumination, the conventional scanners use laser or LED (light emitting diodes). The main advantage of these readers is almost no interference from ambient light in the process due to the fact the reader emit a beam of light more intense than the ambient light, usually red, which sweeps across the area where the code is printed. The equipment interprets the reflection intensity, which is smaller on the darkest areas of the code. The readers based on this technology, however, have a limitation in that the distance measurements can be made when the ambient light begins to interfere with the reflection of the light emitted by the reader. Another problem inherent to traditional readers is that the code must be aligned with the direction of the scanning beam. Some devices minimize this problem by using multiple beams at different angles to try to decode the information.

The wide spread of barcodes also created some problems such as the necessity for the use of unsuitable materials such as polished metal. For example, to solve
this problem, industry has developed readers who used image recognition sensors via CCD (charge-coupled device), using image processing rather than the recognition of light reflection. Readers based on image recognition are able to acquire the barcode from a larger area, depending on the resolution with which the image was generated, and varying angles. For that they make use of image manipulating for location, magnification and rotation before performing the decoding. Unfortunately, some acquired images suffer from quality problems: noise, spots and low resolution in the area of interest. These quality problems may result from both the limitations of the equipment used, such as a camera with low resolution or improper adjustment, as the environment in which the images were acquired. This last factor has, for example, a movement during the acquisition: vibrations arising from a treadmill or inadequate illumination with multiple light sources are not controlled over the area of interest [2].

The barcodes must be printed with the highest contrast between the bars and spaces, avoiding colors that when exposed to the scanner, difficult or result in reading errors [1]. In this way it is safe to convert the images into monochrome images obtained in order to facilitate the extraction and data interpretation. The thresholding technique, which is the converting an image with multiple gray levels into monochrome, is one of the most used to prepare the image for a feature extraction, since it reduces the number of variables that could cause harmful interference to information [3, 4].

In this paper, it studies the characteristics and structures of the bar code system, followed by a computer algorithm, using image processing for decoding and data recovery.

The paper is organized as follows: Section 2 presents the structure and elements of the barcodes; Section 3 introduces the technologies employed and details the proposed algorithm; Section 4 presents the results of decoding process; Section 5 gives conclusions and presents future work.

2. Barcode System

The barcode system is constituted by a series of lines and spaces of various widths, which store information with different orders, called symbols. This format is also known as linear code or one-dimensional. There are encodings known as two-dimensional codes, however, despite being called bar code, they use other elements for representing information such as squares, dots, hexagons and others.

The Association for AIM (automatic identification and mobility) is the trade association representing manufacturers and equipment vendors, systems and accessories for automatic identification, including barcode, radio frequency identification, magnetic tapes, optical character recognition, voice recognition and vision systems.

In 1983, AIM Technical Symbology created the Committee (TSC) to provide technical assistance to the development of generic standards to the barcode. The TSC created five Uniform Symbology Specifications or simply USS to describe the five most common symbologies used in non-commercial applications: USS-39, USS-12/5, USS-Codabar, USS and USS-128-93 [1].

The elements of a bar code are the bars, spaces and mute zones. Each of these elements consists of a multiple of pre-defined width called module. The width of the module directly influences code density: the lower the module, the more symbols may be encoded in the same space. In general, codes with module less than 0.254 mm are known as high density; codes with module greater than 0.508 mm are known as low density.

The bars are lines taller than wide, usually black or very dark, with width varying between one and various modules. The spaces, usually in white or very light, can also vary from one to several modules. Typically, each module of a bar represents a digit “1” and each module of a space represents a digit “0” of the symbology. The specific ordering of bars and spaces within the code
forms the symbol, which stores information in a manner different from their own and others.

The mute zones are reserved margins, prior to the first symbol and after the last code symbol, free of bars, composed only of spaces. Its size usually measures at least 10 times the width of the module code.

Bar codes are composed of an initial mute zone, the start symbol, the coded data, the end symbol and the final mute zone. Some symbologies may have other information like separators and check digits along with coded data.

The characters can be represented separated from each other by a neutral zone called silent interval, which is not part of the code, or continuously, without there being silent intervals, i.e., all spaces are part of the coding.

The symbology Code 39 was created in 1974 and adopted in approximately 1982 by the department of Defense (LOGMARS), Services Administration (GSA) and the U.S. auto industry. It was originally designed to be printed on corrugated cardboard.

The USS-39 or Code 39, in Fig. 1, is a bar code symbology with a full set of alphanumeric characters, the same start and end character and seven special characters. The name “39” comes from the code structure, composed of three wide elements, a total of nine elements. The nine elements are made up of five bars and four spaces each. The ratio between the narrow and wider elements can be represented in the ratio of 1:2 or 1:3 modules. The bars and spaces are intercalated between the elements, and every symbol begins and ends with a slash. In the assembly of complete code the symbols are separated from each other by a silent interval.

The code 2 of 5 was developed in 1968 and subsequently the name “Interleaved 2 of 5” was proposed. It is generally used in industry for identification of products and containers to be stored and distributed.

The specification of the code, USS-12/5, establishes that the symbology is capable to represent only the set of numeric characters and uses a symbol to represent the start character and another symbol for the end character.

The name “Interleaved 2 of 5” derives from the method used to encode pairs of characters. In one symbol, two characters are represented together, using bars to represent the first character and spaces to represent the second. Each character consists of two wide and three narrow elements, with a total of five elements, which may be bars or spaces. The ratio between the narrow and wider elements, as in the code 39, can be represented in the ratio of 1:2 or 1:3 modules.

The characteristics of the code “Interleaved 2 of 5” make that it is even able to represent only groups of information, and commonly a 0 (zero) is added to the left of the information when it has an odd number of characters. The positions of the bars and spaces are not fixed for each character, resulting in that each pair of information has a unique representation. There are no silent intervals between the pairs in the symbol, i.e., all elements are part of the encoded information. If a binary encoding extracted, using a ratio of 1:2 for the elements modules wide, the initial and final symbol will always have the same representation: 1010 and 1101, respectively. The pairs have 14 of information bits wide, and the first number represented by “1” and the second number by “0”.

3. Software Development

A typical computer vision software is able to acquire, process and interpret images, presenting results which assist in decision making, whether by individuals or by automated systems. Using the concepts of an general use image processing system [6, 7], the proposed system uses a video camera and a analog to digital

Fig. 1 Text encoded in Code 39 [5].
converter and a software developed specifically for to recognizing patterns of barcodes decoding the information contained therein.

For application development, the Java language was chosen and used on IDE (integrated development environment) based on the Eclipse platform. The Eclipse is a platform for open source development which, among other advantages, features a huge collection of tools made available by countless developers around the world and has been improved every day. The objective of the platform is to provide means for compiling, distribution and management systems throughout their life cycle [8].

In addition to the native support for multitasking, the Java language also has a feature that made it ideal for use in distributed applications: its platform independence. This independence is due to the fact that the Java compiler does not generate instructions specific to a platform, but a program on intermediate code called bytecode, which can be described as a language machine to a virtual processor that does not physically exist. The compiled Java code can then be executed by a bytecode interpreter: the JVM (Java virtual machine) which is a processor emulator. The JVM will be responsible for the translation of bytecode to native machine commands. The JVM instruction set is optimized to be small and compact, and language performance is sometimes sacrificed to ensure this reduced set of instructions [9].

To make the acquisition of images from a video camera possible, the DSJ (DirectShow Java) wrapper is used. The DSJ is a low level library that allows applications developed in Java access to the image components from Microsoft Windows. It is a set of tools and resources available for any Java developer to be able to add multimedia capabilities to their applications [10].

The software developed uses the class DSFiltergraph to acquire an image containing a bar code for the subsequent decoding. The next step is to convert the image into a binary image, containing only black or white pixels, using a threshold algorithm. A binarized image is in fact a two dimensional matrix containing only the elements 1 and 0, respectively representing the tones black and white of the image. With the image in this format, it is far simpler to recognize the patterns of barcodes [3, 4, 11].

In Fig. 2, it is possible to observe the early stages of the decoding process. Since there are great differences between the codes to be analyzed, the character encoding patterns must be previously identified and will be one of the inputs to the software, along with the binarized image.

The first step of processing is to analyze the image and set the strategy for scan the lines in order to be able to decode the information in a reliable, faster and with less processing time possible. The process starts from the horizontal center line of the image and, in case of not finding the code, it moves to the next line to be analyzed below, then to the line at the same distance above the centerline. Thus, the image is scanned until it finds the code or no more lines to be analyzed.

Each selected line image is transformed into one-dimensional array which is passed to the process of identifying and decoding. If the process is not successful,
a new attempt is performed, in case the code is being displayed upside down.

The decoding process starts with the loading of data from the bar code symbology into the memory. With a loaded symbology, a scan is performed in line seeking the pattern of the start character. If it is not found, the next symbology is loaded and the search is performed for the new start character.

From the point where the symbology start character was found, the process starts a scan of the encoded information until it finds the pattern of the end character.

The identification of patterns is done through a function (see List 1) that processes the bit array, accumulating the number of equal bits (0 or 1) which are in sequence, and stores the values in another array, the size of which equals the amount of bars and spaces, where each character is encoded according to the symbology pattern. It is possible to distinguish the narrow elements and the wide ones in this way. With the information retrieved from the image, the process compares the data obtained with the symbol table of the bar code symbology to decode the character. The flow of the decoding process may be seen in details in Fig. 3.

![Fig. 3 Decoding process.](image-url)

**List 1 Function to find the patterns of character.**

```java
/**
 * @param row = vector of pixels
 * @param start = Number of columns to ignore
 * @param counters = vector to store the sequence bits (bars or spaces)
 * @throws NotFoundException if the code is not found
 */
protected static void recordPattern(BitArray row, int start, int[] counters) throws NotFoundException {
    int numCounters = counters.length;
    for (int i = 0; i < numCounters; i++) {
        counters[i] = 0;
    }
    int end = row.getSize();
    if (start >= end) {
        throw NotFoundException.getNotFoundInstance();
    }
    boolean isWhite = !row.get(start);
    int counterPosition = 0;
    int i = start;
    while (i < end) {
        boolean pixel = row.get(i);
        if (pixel ^ isWhite) { // that is, exactly one is true
            counters[counterPosition]++;
        } else {
            counterPosition++;
            if (counterPosition == numCounters) {
                break;
            } else {
                counters[counterPosition] = 1;
                isWhite = !isWhite;
            }
            i++;
        }
    }
    if (!(counterPosition == numCounters ||
          (counterPosition == numCounters - 1 && i == end))) { // code not found
        throw NotFoundException.getNotFoundInstance();
    }
}
```
4. Results

To obtain preliminary results, we selected two samples of different codes each having a particular characteristic.

To validate the decoding process, we used a sample image of the code 39. The image displays a bar code along with its numeric representation printed in black with a yellow background.

The first step of the process is to convert the color image into a grayscale image. Then convert the image to a binary image. The final result can be seen in Fig. 4. The software identified and decoded the code in the binarized image.

Having been validated the process for decode the image, the tests will be performed now with images captured directly from a digital camera equipped with a set of lenses for focus adjustment without adjustment option approach.

The image to be worked is a monochromatic barcode, USS-I2/5, printed in black with white background. According to the concepts passed by Erdei [1], this setting of barcode is among the ideal ones to perform the decoding. By applying once again the decoding process, the application could decode the image and retrieve the information contained in the barcode.

5. Conclusions

The results of the tests with conventional codes demonstrate the viability of software development and the possibility to apply it to logistic systems where the use of the barcode is required.

A proposal to continue the research leads to the development of a software capable of decoding barcodes Code 39 and USS-I2/5 written in other types of material. The codes are written in metal tubes with a low contrast between the bars and the surface of the tube. Since this type of printing is highly discouraged, it presents great difficulties for decoding which mainly due to reflective characteristic of the material. This characteristic makes the lighting and the objects around hinder the acquisition of an image with a good definition of the printed codes. Since the modification of the recording or the material is not an option in this particular case, the study should follow the development of a controlled environment which provides the lighting and reflection conditions for obtaining ideal imaging of the tubes.

References