Published in final edited form as:
Proc IEEE Workshop Appl Comput Vis. 2009 December 1; 2009(7-8): 1–6. doi:10.1109/WACV.2009.5403090.

Reading Challenging Barcodes with Cameras

Orazio Gallo and **Roberto Manduchi**
University of California, Santa Cruz

Orazio Gallo: orazio@soe.ucsc.edu; Roberto Manduchi: manduchi@soe.ucsc.edu

Abstract

Current camera-based barcode readers do not work well when the image has low resolution, is out of focus, or is motion-blurred. One main reason is that virtually all existing algorithms perform some sort of binarization, either by gray scale thresholding or by finding the bar edges. We propose a new approach to barcode reading that never needs to binarize the image. Instead, we use deformable barcode digit models in a maximum likelihood setting. We show that the particular nature of these models enables efficient integration over the space of deformations. Global optimization over all digits is then performed using dynamic programming. Experiments with challenging UPC-A barcode images show substantial improvement over other state-of-the-art algorithms.

1. Introduction

Virtually every packaged good is labeled with at least one form of barcode, generally a flavor of either the EAN or the UPC standards. The success of barcode technology for identification, tracking, and inventory derives from its ability to encode information in a compact fashion with very low associated cost.

Commercial laser-based, hand-held barcode scanners achieve robust reading with a reasonable price tag. Recently, however, there has been growing interest in accessing barcodes with regular cellphones rather than with dedicated devices. Since cellphones are of ubiquitous use, this would enable a multitude of mobile applications. For example, a number of cellphone apps have appeared recently that provide access to the full characteristics and user reviews for a product found at a store, or to on-line price comparisons via barcode reading. Tekin and Coughlan describe a system that detects barcodes in clutter and at a distance, to direct people with visual impairments towards barcodes on packages and decode them [11].

Unfortunately, the poor quality of the images taken by current cellphone cameras makes it surprisingly difficult to correctly decode barcodes, as shown by the limited performance that many commercial applications exhibit (see our tests in Sec. 4).

This paper presents a new algorithm for barcode reading that produces excellent results even for images that are blurred, noisy, and with low resolution. Quantitative comparisons on existing and new barcode image databases show that our technique outperforms other state-of-the-art softwares and compares favorably with other reported results.

A unique characteristic of our algorithm is that it never performs binarization of the graylevel brightness profile before processing. We argue that this early-commitment operation, executed by virtually all existing algorithms, translates into unrecoverable information loss, thus complicating all further processing. This is especially the case for low-resolution images, where

binarization errors may have catastrophic effects. For example, Fig. 1 shows a challenging barcode, which would be hardly decoded using binarization.

2. Previous work

Barcode decoding has been studied and optimized for decades and it now represents a consolidated industrial standard. (Pavlidis *et al*. provide an interesting analysis of this technology from an information theory standpoint [8].) Until recently, however, barcode reading was performed almost exclusively with dedicated, generally expensive hardware. Despite the rapid growth of interest in camera-based readers, most of the challenges posed by this new approach are yet to be solved.

Commercial scanners, such as those used in supermarkets, shine a light on the code and measure the intensity of its reflection, thus being virtually insensitive to ambient illumination. Mirrors then allow for the acquisition of multiple images of the code, which is also assumed to be fairly close to the scanner. As a consequence, the extracted scanlines are of very high quality.

Camera-based methods generally produce much lower quality scanlines. Moreover, the barcode generally does not completely fill the image; the first step to barcode interpretation is therefore its localization. We argue that this stage is intrinsically more robust to noise than the actual decoding, for it requires a lower accuracy. Existing methods for this step apply to the binarized image methods based on Hough transforms [5], edge orientation histograms [15], morphological operators [2], or wavelet transforms [14]. Other approaches simplify the problem assuming that the center of the image falls within the barcode area [6,13].

Although decoding is a far more delicate process than localization, virtually all current techniques use some kind of binarization for this stage as well; even with the many successful thresholding algorithms that have been proposed [7,9], a moderate amount of blur in the original image can dramatically affect the results (see Fig. 1).

Decoding can be performed by estimating the width of all the bars in the barcode from the binarized image [1,2,5]. Wachenfeld *et al*., instead, use an adaptive thresholding [13]. They then model the different digits and find the combination that best explains the scanline. Although they report a very high accuracy, they do not present any comparative test.

Krešić-Jurić *et al*. find potential edges differentiating the scanline and then use hidden Markov models to remove false positives [4]. Their method compares favorably with previous approaches although it was implemented on commercial barcode scanners. Tekin and Coughlan propose an elegant Bayesian framework for barcode decoding [10]. Their approach aims to find the edges of the bars in a fashion similar to Krešić-Jurić *et al*. [4] but they allow for both false positive and false negative. Tekin and Coughlan made their database available and in Sec. 4 we compare our results with theirs.

Our main contribution is an algorithm capable of decoding 1-D barcodes from noisy pictures, whether the source of noise be motion blur or lack focus. Our method also proves effective on highly compressed pictures. Additionally, we propose a simple method to segment the barcode out of the picture. Finally, we created a dataset of barcode images [1] and evaluate our algorithm on it by means of comparison with publicly available barcode readers. The comparisons show that the proposed method outperforms existing methods, in particular as the quality of the image worsens.

[1]The dataset is available at http://users.soe.ucsc.edu/~orazio/Data/Barcodes200909.zip

3. The algorithm

Given an image containing a barcode, two distinct operations are needed for accessing the information contained in the barcode: *localization* and *reading*. Localization typically relies on the strong textural content of the barcode. Reading can be performed on one or more scanlines extracted by the localization step.

Although our work focuses on barcode reading, we implemented a simple and fast localization algorithm that finds the highest energy region in the map generated by subtracting the vertical from the horizontal gradient. This algorithm is by no means optimal but it works reasonably well in our studies, as it only serves as a necessary pre-processing stage to enable the experiments of Sec. 4. We also implemented other algorithms from the literature [15,12] and none outperformed our simple method even with a higher computational load.

Our reading algorithm analyzes a single scanline contained in the detected barcode area. The only requirement is that the beginning and the end of the barcode pattern in the scanline are detected with a certain accuracy. For example, in our implementation we assume a localization tolerance in either end point equal to twice the width of the narrowest bar. Note that this task is much simpler than localizing all bars in the code, an operation that we avoid altogether. The UPC-A standard, in fact, requires that the initial and final bars be separated from the edge of the barcode by a *quiet area* of a width equal to at least 9 times the width of the narrowest bar, usually referred to as *base width*.

Here is the algorithm outline. First, based on the detected endpoints of the scanline, we compute the spatial location of each digit segment in the barcode. For each of the 12 digits in the barcode, we compare the intensity profile of the corresponding segment of the scanline with binary templates, each representing a specific digit value as shown in Fig. 2. In order to account for inaccuracy in the localization of the spatial extent of each digit, we allow these templates to shift and scale in the horizontal direction. After defining a likelihood function to measure how well a deformed (shifted and scaled) template explains the observed intensity, one may be tempted to search for the deformation parameters that maximize it (that is, the shifted and scaled template that best explains the data), hoping not to end up in one of the many existing local maxima. We take a better, and theoretically more sound, route: we integrate the likelihood over the space of deformations, having defined a prior distribution of the deformation parameters. One important contribution of this work is to derive an algorithm that computes this marginalization integral *exactly* and in affordable computing time.

The scanline could finally be decoded choosing the digit values that maximize the likelihood of the data within each digit segment; however, this may lead to incorrect results due to noise, blur, or other causes. The risk of such errors can be reduced by exploiting global constraints on the overall sequence of digit values. The idea is that the “optimal” deformed templates should sit side by side on a scanline, without overlaps or gaps. We define a global cost function that, for each possible sequence of digit values, penalizes overlaps or gaps in the sequence of deformed templates, with the deformation parameters obtained by least squares regression. The minimum cost sequence can then be found via dynamic programming.

Algorithmic details are provided in the next subsections; for a more in-depth description of the algorithm the reader is referred to our technical report [3].

3.1. Deformable models

We define a *model* (or *template*) $\mathcal{M}^k$ for digit k as a continuous piecewise constant function that alternates between −1 and 1, where a value of −1 (1) represents a black (white) bar (see Fig. 2). A model $\mathcal{M}^k$ for a digit in the left half of a UPC-A barcode begins with a ‘−1’ segment and

ends with a '1' segment, where both such segments have length of 1. The lengths of the constant segments between these two end segments depends on the specific digit k, as shown in Fig. 2. A model is therefore an archetypical representation of one digit of a standardized scanline, which also includes one bar from each one of the nearby digits. These two additional bars have base width and known polarity; adding such bars to the template increases robustness of the matching process.

A parameterized model $\mathcal{M}_{o,w}^{k}$ is a version of the original model that is based on the starting point of the pattern o and scaled with w, the base width. (Note that models are functions of the continuous line, while the observation $I(n)$ is defined over the discrete space of pixels.) An example of deformed model is shown in Fig. 3.

3.2. Digit segment – conditional likelihood

Once the barcode has been localized in the image, and the endpoints (o_S, o_E) of the selected scanline have been estimated, the approximate starting point of the j-th digit segment in the left side of the barcode is

$$o = o_s + 3w + 7w\,(j-1), \tag{1}$$

where:

$$w = \frac{o_E - o_S}{95} \tag{2}$$

is the estimated base width. These expressions derive from the fact that the overall length of the barcode is (ideally) equal to 95 times the base width, that each digit uses a segment (support) equal to 7 times the base width, and that the first 3 bars are guard bars.

The estimates computed in Eq. 1 can be noisy for localization imprecision or image distortion. However, suppose that the estimated location o and minimum bar width w are indeed correct. Then, in order to read the value of the digit, we could simply compare the intensity $I(n)$ within the segment with the models $\mathcal{M}_{o,w}^{k}$ for $0 \le k \le 9$, and pick the model that best fits the data. More precisely, we define the likelihood of the intensity within a generic digit segment when the digit takes a value of k (conditioned on o and w) as

$$p_k(I|o,w) \propto e^{-D(I,\mathcal{M}_{o,w}^{k})}, \tag{3}$$

where $I(n)$ represents the intensity profile of the considered scanline. The log-likelihood term D can be expressed as

$$D(I,\mathcal{M}_{o,w}^{k}) = \sum_{n=\lceil o-w \rceil}^{\lfloor o+8w \rfloor} D\,(I(n),\mathcal{M}_{o,w}^{k}(n)), \tag{4}$$

where the variable n takes on only integer values (see Fig. 3). Note that this sum is computed over all pixels that fall within the segment $[o-w, o+8w]$, which is the support of $\mathcal{M}_{o,w}^{k}(x)$.

After computing the means of the largest 50% and smallest 50% values of $I(n)$, μ_w and μ_b, and the cumulative variance σ^2, we model the log-likelihood D as:

$$D(I(n), -1) = \frac{[\max(I(n) - \mu_b, 0)]^2}{2\sigma^2} \tag{5}$$

and

$$D(I(n), 1) = \frac{[\min(I(n) - \mu_w, 0)]^2}{2\sigma^2}. \tag{6}$$

This function penalizes values of $I(n)$ that are small when $\mathcal{M}^k_{o,w}(n) = 1$ or large when $\mathcal{M}^k_{o,w}(n) = -1$.

3.3. Digit segment – total likelihood

A maximum likelihood approach based on the likelihood described above can be extremely sensitive to the even small errors of o and w. Such errors derive from the expected error in the estimation of the endpoints o_S and o_E. Assume for example that both o_S and o_E are computed with a tolerance of $\pm\Delta o$. Then, barring deformations or perspective effects, o has a tolerance of $\pm\Delta o$ as well, whereas w has a tolerance of $\pm 2\Delta o/95$. Thus, a method for taking uncertainty of the deformation parameters into account is necessary.

We approach this problem by first defining a probability density function $p(o, w)$ over the space of deformations. We then compute the total likelihood $p_k(I)$ by averaging $p_k(I|o, w)$ over such density:

$$p_k(I) = \int\int p_k(I|o, w) p(o, w)\, do\, dw. \tag{7}$$

Computing this integral may seem like a daunting task, especially if this needs to be performed in real time over an embedded computer such as a cell phone. On the contrary, we show that due to the particular nature of the model $\mathcal{M}^k$, and assuming a simple form for the prior $p(o, w)$, the integral in (7) can be computed *exactly* via numerical means with reasonably small complexity.

Our derivation exploits the fact that $D(I, \mathcal{M}^k_{o,w})$ is piecewise constant in the (o, w) space. This can be seen by breaking up the sum in (4) into six pieces, corresponding the segments in which $\mathcal{M}^k_{o,w}(x)$ takes on constant values of 1 or −1. More formally, within each of the constant tracts $[d_i, d_{i+1}]$, the function $\mathcal{M}^k_{o,w}(x)$ is identically equal to $(-1)^i$.

$$D(I, \mathcal{M}^k_{o,w}) = \sum_{i=1}^{5} A_i = \sum_{i=1}^{5} \sum_{n=\lceil d_{i-1} \rceil}^{\lfloor d_i \rfloor} D\,(I(n), (-1)^i). \tag{8}$$

Hence, a variation of o or w determines a change of A_i (and therefore of $p_k(I|o, w)$) only when it causes d_{i-1} or d_i to cross over an integer value. Consequently, $p_k(I|o, w)$ is piecewise constant, and the integral in (7) becomes a sum. Next, we show how to compute the terms in this sum.

Let $\{\mathcal{V}_k^t\}$ be the minimum partition of the (o, w) plane such that $p_k(I|o, w)$ is constant within each cell $\mathcal{V}_k^t$ (with t representing the index of cells in the partition). Then

$$p_k(I) \propto \sum_t e^{-D_t} \int\int_{\mathcal{V}_k^t} p(o,w)\, do\, dw, \tag{9}$$

where $D_t = D(I, \mathcal{M}_{o,w}^k)$ for any (o, w) in $\mathcal{V}_k^t$. Note that the cells $\mathcal{V}_k^t$ are polygonal. The list of cells $\{\mathcal{V}_k^t\}$, as well as the integral of $p(o, w)$ within each cell, can be computed offline and stored for online use. In fact, the cells form a periodic pattern (with period equal to 1 both in o and w), hence only the cells within such a period need to be stored.

As will be shown shortly, it is also useful to estimate, for each possible digit value k, the deformation parameters (o, w) given the intensity profile $I(n)$ within a digit segment. We choose the least squares estimator $(\bar{o}_k, \bar{w}_k)$ of these quantities (under the density $p(o, w)$), which is given by the conditional expectation.

3.4. Imposing spatial coherence

Our model makes the simplifying initial assumption that the digit segments are equally spaced (see (1)–(2)). This also implies that the base width w is constant across the bar-code. In practice, we should expect that the digit segment length may vary from segment to segment, hopefully within the confidence intervals Δo and Δw. Ideally, however, the segment representing a given digit in the scanline (as computed from the estimates $\bar{o}_k$ and $\bar{w}_k$) should be adjacent to (but non overlapping with) the neighboring segments. The choice of an incorrect value of k due to single-digit analysis is likely to result in a supported segment that does not fit well together with the other segments. This observation can be exploited by imposing a global constraint as follows.

Suppose that the j-th digit takes value $k(j)$. (Note that we need to make the dependency on j explicit in our notation from now on.) The estimated deformation parameters $(\bar{o}_{j,k(j)}, \bar{w}_{j,k(j)})$ define the supported segment $[\bar{o}_{j,k(j)}, \bar{o}_{j,k(j)} + 7\,\bar{w}_{j,k(j)}]$. We define the overlap/gap extent between the j-th and $(j + 1)$-th estimated digit segments as

$$O_{j,k(j),k_{j+1}} = |\bar{o}_{j,k(j)} + 7\bar{w}_{j,k(j)} - \bar{o}_{j+1,k(j+1)}|. \tag{10}$$

Now define a global cost function as follows:

$$C(\{k\}) = \sum_j \alpha O_{j,k(j),k(j+1)}^2 - \log p_{j,k(j)}, \tag{11}$$

where α is a balancing parameter, and the sum extends to all digits in the left and right half of the barcode. (α was set to be equal to 0.1 in our experiments). The cost function in (11) penalizes sequences of digit values that create large overlaps or gaps between two consecutive digit segments or that produce low values of likelihood. Dynamic programming can be used to minimize the cost function C over the space of sequences $\{k\}$ of digit values.

4. Implementation and tests

We implemented and tested our algorithm in Matlab. A simple localization algorithm was used to provide a scan-line segment input to our reader. The maximum tolerance Δo at the endpoints

o_S and o_E was set to $\pm 2w$, where w, the initial estimate of the base width, was defined in (2). In practice, this means that we expect the localization algorithm to possibly miss one the first bar in the start and end pattern. The minimum scanline width $o_E - o_S$ that our algorithm was able to decode was of 100 pixels. Note that this corresponds to a base width of only 1.05 pixels.

The computational speed of the algorithm depends heavily on the scanline width. This is mostly due to the fact that the number of cells V_k^t depends on the tolerances Δo and Δw, which are proportional to the scanline width. For example, when the scanline width is equal to 100 pixels, then 25 cells are generated. However, in the case of a high resolution image of a barcode at short distance, producing a scanline width of 1128 pixels, 2185 cells are generated. In order to reduce the number of cells to consider, we implemented a simple variation of the algorithm, by fixing the width of the first and last bars in the model to the minimum width considered. Remember that these are "overlap" bars with nearby digits, and that they always have unitary width in the model $\mathcal{M}^k$. With this modification, the number of cells is reduced to 17 in the 100 pixel scanline width case, and to 641 in the 1128 pixel case. The computational speed of the overall reader (excluding the original segmentation) ranges between 0.076 seconds to 0.65 seconds.

In order to assess the performance of the system, we tested it on a variety of images. Unfortunately, we could find only one barcode image database for comparative assessment[2]. This database, which was created by Tekin and Coughlan, is accessible at www.ski.org/Rehab/Coughlan_lab/Barcode. The images are all of high resolution, and each image was manually cropped around the barcode. The authors divided it into "Hard" and "Clean" subsets, and showed results of their algorithm on both sets [10].

In order to assess our system in other realistic situations, we gathered a number of images taken from two different cellphones, and created three new data sets. Data set 1 contains images at high resolution (1024 × 768) from a Nokia N95 cell phone. This device has autofocus capability, although not all images collected were properly in-focus, and some had some amount of motion blur. Data set 2 contains images taken by the same cell phone, but at 640 × 480 resolution, and highly compressed in JPEG (with apparent coding artifact). Data set 3 contains images at 1152 × 864 resolution taken with an older Nokia 7610 phone, with fixed focus. All three data sets have multiple pictures of several barcodes, taken from different distances and in different conditions.

Our algorithm was tested against the algorithm of [10], for the "Hard" and "Clean" data set, as well as against two barcode reading softwares that are available online. The first one, from DataSymbol[3], was also considerd in [10]. The second one, from DTK[4], was shown to produce impressive results. A third online software, from QualitySoft, was considered in [10], but we neglected comparison with it since it gives very poor results.

Numerical results from our tests are presented in Fig. 4. It is shown that in all the data sets, our algorithm outperforms the other techniques. In particular, our algorithm performs comparatively well for the most challenging sets (Data set "Hard" and Data set 3). A sample of images correctly decoded by our algorithm is shown in Fig. 5, while examples of failures are shown in Fig. 6. Note that in many cases, failure was due to incorrect initial localization due to poor segmentation. The reader is invited to zoom-in to notice how poor the image quality is for many of the pictures that our algorithm correctly decodes.

[2]The authors of [13] claim to have assembled a barcode image database for public use, but we have not been able to access it.
[3]http://www.datasymbol.com
[4]http://www.dtksoft.com

5. Conclusions

We have presented a new algorithm for barcode reading that can deal with images that are blurred, noisy, and with low resolution. Unlike previous approaches, this algorithm does not binarize the image, thus sidestepping a critical and often error-prone early-commitment procedure. We use deformable templates for representing each digit of the bar-code, integrating over the set of all expected deformations. A final procedure for global spatial coherence helps reducing the risk of errors at the individual digit level. The experimental results show improved performance with respect to other state-of-the-art software and algorithms, especially for the most challenging images. Our Matlab implementation of the overall system runs reasonably fast; porting of the algorithm to a Symbian cellphone is in progress.

The most obvious improvement to the algorithm would be to consider non-horizontal lines (for rotated barcodes) and multiple scanline analysis. The spatial coherence constraint could be easily extended to the multiple scanlines case. Another issue worth further consideration is the model for the greylevel values. Our current model is bimodal—each pixel is assumed to be black or white. In fact, the model could be improved to account for the point spread function of the optical system, which smears the edge of a bar across a few pixels. This enhanced model would be particularly useful for low resolution imagery, where even a small amount of blur may completely wash out the narrower bars.

References

1. Adelmann R, Langheinrich M, Flörkemeier C. A Toolkit for Bar-Code Recognition and Resolving on Camera Phones–Jump Starting the Internet of Things. Workshop Mobile and Embedded Interactive Systems at Informatik. 2006
2. Chai D, Hock F. Locating and decoding EAN-13 barcodes from images captured by digital cameras. Int'l Conf on Information, Communications and Signal Processing. 2005
3. Gallo O, Manduchi R. Reading 1d barcodes from noisy, blurred, and highly compressed pictures. Tech Report UCSC-SOE-09-26, UCSC. 2009
4. Krešić-Jurić S, Madej D, Santosa F. Applications of hidden Markov models in bar code decoding. Pattern Recognition Letters. 2006
5. Muniz R, Junco L, Otero A. A robust software barcode reader using the hough transform. Int'l Conf on Information Intelligence and Systems. 1999
6. Ohbuchi E, Hanaizumi H, Hock L. Barcode readers using the camera device in mobile phones. Int'l Conf on Cyberworlds. 2004
7. Otsu N. A threshold selection method from gray-level histograms. Trans on Systems, Man, and Cybernetics. 1979
8. Pavlidis T, Swartz J, Wang Y. Fundamentals of bar code information theory. Computer. 1990
9. Solihin Y, Leedham C. Integral ratio: a new class of global thresholding techniques for handwriting images. PAMI. 1999
10. Tekin E, Coughlan J. A bayesian algorithm for reading 1d barcodes. Canadian Conf on Computer and Robot Vision. 2009
11. Tekin E, Coughlan JM. An algorithm enabling blind users to find and read barcodes. Workshop on Applications of Computer Vision. 2009
12. Tropf A, Chai D. Locating 1-D Bar Codes in DCT-Domain. Int'l Conf on Acoustics, Speech and Signal Processing. 2006
13. Wachenfeld S, Terlunen S, Jiang X. Robust recognition of 1-d barcodes using camera phones. Int'l Conf of Pattern Recognition. 2008
14. Wang K, Zou Y, Wang H. 1d bar code reading on camera phones. Int'l Journal of Image and Graphics. 2007
15. Zhang C, Wang J, Han S, Yi M, Zhang Z. Automatic real-time barcode localization in complex scenes. Int'l Conf of Image Processing. 2006

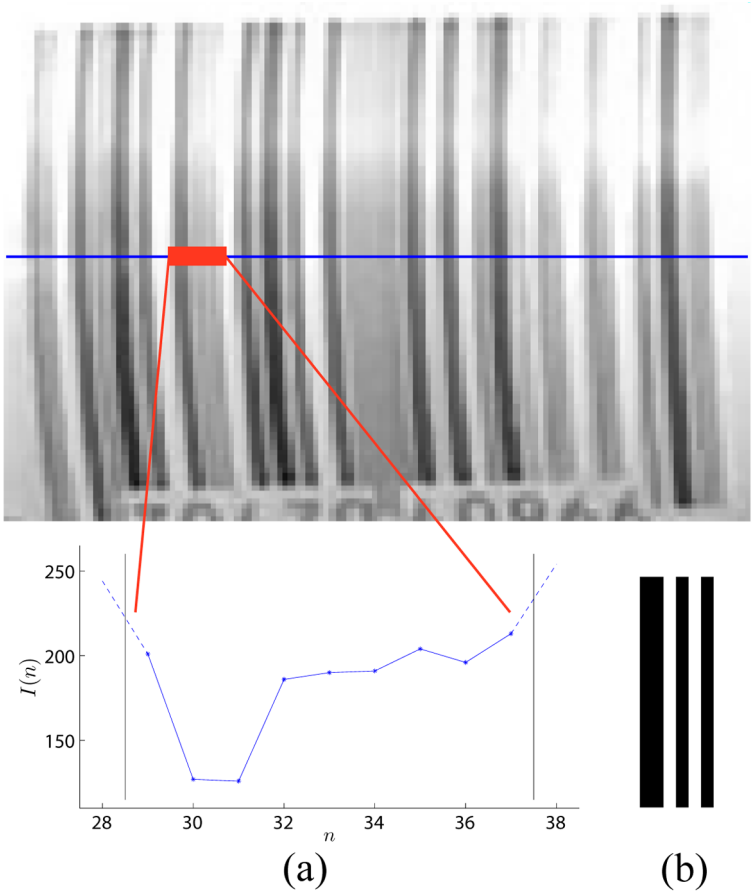

Figure 1.
A challenging barcode image that is correctly decoded by our algorithm. The intensity profile from the segment highlighted in red on the blue scanline is plotted in (a). The underlying sequence of spaces and bars is shown in (b). Note how blur and low resolution affect the intensity profile. A system that binarizes the intensity would be hard-pressed to detect the correct pattern.

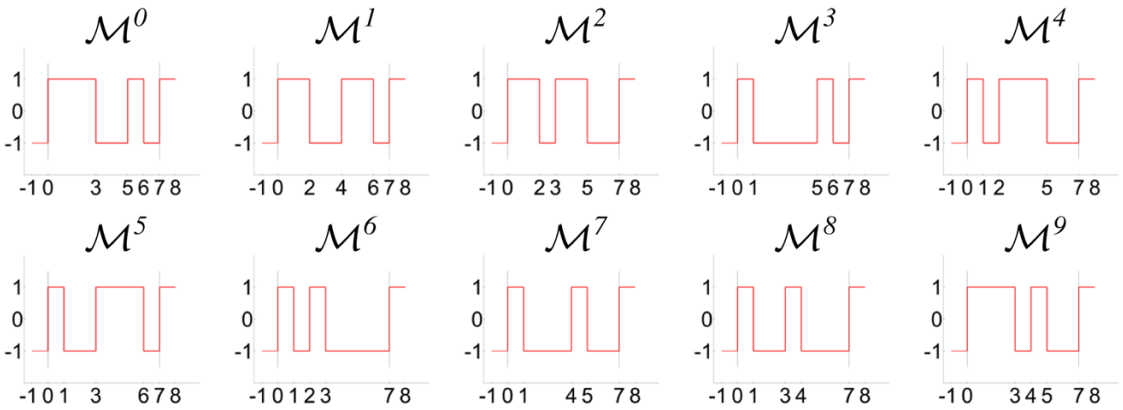

Figure 2.
Each digit in a UPC-A code is encoded with a sequence of two bars and two spaces, represented in these graphs by values of 1 and −1.

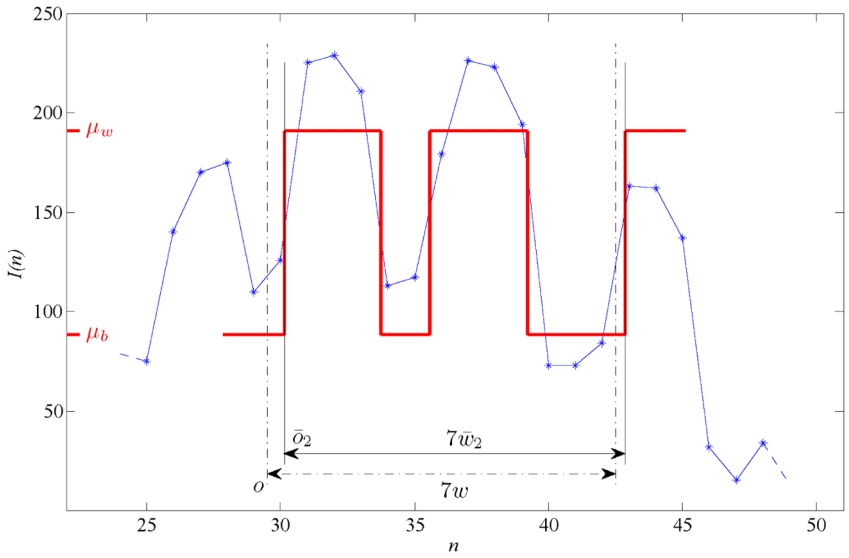

Figure 3.

A sample of intensity profile in a scanline (blue line). The segment $[o, o + 7w]$ represents the location of the initial digit segment obtained from (1)–(2), whereas the segment $[\bar{o}_2, \bar{o}_2+7\bar{w}_2]$ is the estimated support segment for $k = 2$. The red line represents the deformed model $\mathcal{M}^2_{\bar{o}_2,\bar{w}_2}$. For the sake of graphical clarity, the model was scaled in amplitude so that it alternates between μ_b and μ_w (as defined in Sec. 3.2).

Data set “Clean” from [10]				
Our algorithm	[10]	DataSymbol	DTK	Total
43	42	39	**43**	44

Data set “Hard” from [10]				
Our algorithm	[10]	DataSymbol	DTK	Total
19	2	0	1	35

Data set 1			
Our algorithm	DataSymbol	DTK	Total
43	26	42	62

Data set 2			
Our algorithm	DataSymbol	DTK	Total
10	4	6	10

Data set 3			
Our algorithm	DataSymbol	DTK	Total
9	0	0	20

Figure 4.
Comparative results showing the number of barcodes correctly detected in the different data sets considered. The last column reports the total number of images in each data set.

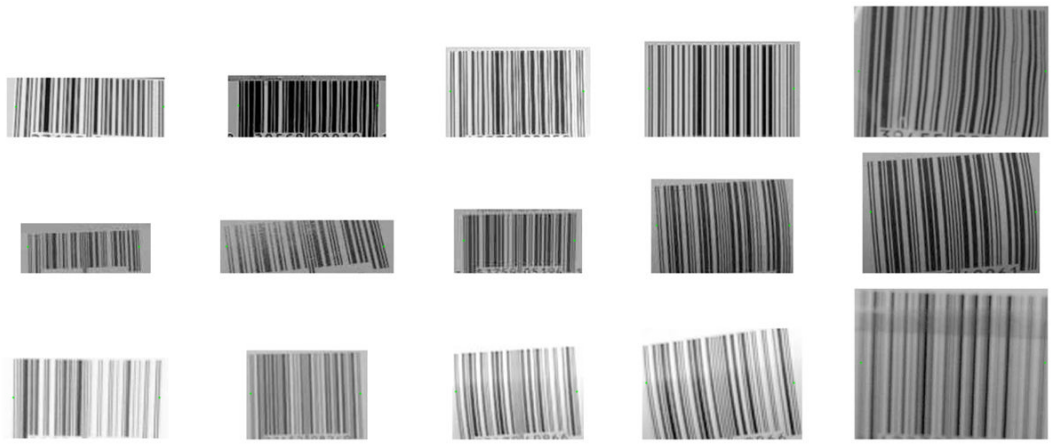

Figure 5.
Example of barcodes correctly decoded by our algorithm from our three data sets. Each image shows the segment extracted by the localization algorithm. The green stars indicate (o_S, o_E), the original estimates for the endpoints of the scanline.

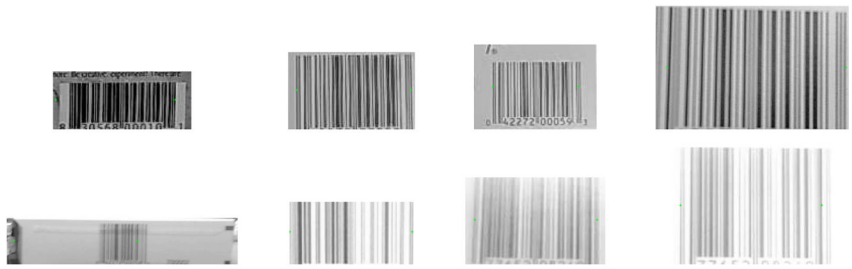

Figure 6.
Example of images in which our algorithm fails from Data Set 1 and 3. The green stars indicate (o_S, o_E), the original estimates for the endpoints of the scanline.