A UNIFIED PROBABILISTIC FRAMEWORK FOR ROBUST DECODING OF LINEAR BARCODES

Umut Şimşekli

Boğaziçi University
Dept. of Computer Engineering
34342, Bebek, İstanbul, Turkey

Tolga Birdal

Technische Universitat München (TUM)
Dept. of Computer Science, CAMP
Boltzmannstr. 3, D-85748 Munich, Germany

ABSTRACT

Both consumer market and manufacturing industry makes heavy use of 1D (linear) barcodes. From helping the visually impaired to identifying the products to industrial automated industry management, barcodes are the prevalent source of item tracing technology. Because of this ubiquitous use, in recent years, many algorithms have been proposed targeting barcode decoding from high-accessibility devices such as cameras. However, the current methods have at least one of the two major problems: 1) they are sensitive to blur, perspective/lens distortions, and non-linear deformations, which often occur in practice, 2) they are specifically designed for a specific barcode symbology (such as UPC-A) and cannot be applied to other symbologies. In this paper, we aim to address these problems and present a dynamic Bayesian network in order to robustly model all kinds of linear progressive barcodes. We apply our method on various barcode datasets and compare the performance with the state-of-the-art. Our experiments show that, as well as being applicable to all progressive barcode types, our method provides competitive results in clean UPC-A datasets and outperforms the state-of-the-art in difficult scenarios.

Index Terms— barcode decoding, hidden Markov models

1. INTRODUCTION

One of the key technologies in closing the producer-consumer gap is product identification and traceability, which is currently achieved by the most prevalent technology: linear (1D) barcode scanning. The mass manufacturing, as well as the consumer goods market, regardless of the sector rely on barcodes, especially linear codes.

In manufacturing systems, high accuracy and speed lead to high throughput and more profit and thus they require robustness, durability, and noise tolerance in barcode scanning systems. At this point conventional laser scanners loose their charm, as they mostly fail to read multiple codes sequentially and necessitate customization of the conveyor belts. Instead, machine vision solutions are desired, where the cameras could capture multiple barcodes in arbitrary orientations and decode them. However, the performance requirements of those systems crave for a broad solution, which is able to decode various types of barcodes in difficult scenes, located on different surfaces, and posed under varying constraints.

From the consumers perspective, it is noticeable that in recent years, there has been a huge attempt in empowering end-users with cheap software tools to benefit from barcodes. A customer can now

Fig. 1. Different types of challenging barcodes (UPC-A and Code128) that are successfully decoded with our algorithm.

get more information about a product or can experience the entire shopping process without the burden of a shopping cart. It is also worth mentioning that as barcodes implicitly describe products, they are an important source of information for visually impaired people as well. Considering the uncontrolled setting of mobile barcode decoding through low resolution and low SNR cameras, the performances of available software are far from what is desired [1]. While the demand and reliance on barcodes prevail if not increase, we have noticed that there is no de facto standard for retrieving barcode information from cameras. This is due to the fact that a typical image obtained from a CCD/CMOS sensor is subject to various degradations, such as camera noise, motion blur, perspective transform, and lens distortion.

In this paper, we present a novel method for decoding barcode scanlines robustly from challenging scenes. Our aim is to achieve high accuracy, high speed, and generalizability by harnessing a hierarchical hidden Markov model (H-HMM) to model the intrinsic nature of the barcode signals, without any pre- or post-processing. While reaching the state of the art in terms of accuracy, our method, to the best of our knowledge, is the first to institute such an intuitive and flexible probabilistic framework, which eases the implementation of new barcode types. Our main contributions can be summarized as follows:

- We show how to model linear barcodes rigorously by using an H-HMM and establish a single pass algorithm for barcode reading. We cast the decoding problem to finding the optimal path in an HMM and enjoy the efficiency of the Viterbi algorithm.
- Our model is flexible. We propose a common decoding method for all types of progressive (i.e., non-interleaved) symbologies, such as UPC-A, EAN-13, CODE-128, CODE-39, where the state of the art is symbology-dependent.
- Our model is robust against blur, noise, non-linear deformations and captures the intrinsic nature of the waveform, without making use of inaccurate discretizations.
- We demonstrate that, with our method it is possible to customize the computation time - accuracy trade-off by utilizing a pruning strategy, leading to the state of the art accuracy in difficult scenar-

ios, and achieving real-time performance in typical settings. Some challenging barcodes that are correctly decoded by our method are shown in Figure 1.

2. RELATED WORK

At the development stage of barcode technology, information theoretic laser scanning [2, 3] was the only path. This perspective continued until today and cultivated the main developments in hardware oriented barcode scanning [4]. Later on, with the advancements of machine vision systems, many solutions are proposed to decode barcodes from images.

The first approaches suggested derivative analysis for peak localization on the waveforms [5], which are known to be noise sensitive. These techniques are further enhanced but despite the workarounds, they are far from completing the picture [6, 7]. Later on two step procedures (binarization then decoding) became the standard. The problematic binarization stage was approached by various methods utilizing edge operators, gradient information, blind deconvolution [8, 9, 10].

In the recent years, probabilistic models for decoding barcodes have become popular. Yet, these approaches did not go far from the other methodologies as they were mostly based on conventional binarization-decoding procedure. Krešić-Jurić et. al. have used HMMs to better detect the edges of barcodes [11], which again suffered from the drawbacks of binary labeling. Tekin and Coughlan proposed an elegant Bayesian framework to model the shape and appearance of the barcodes [12]. Their model is implicitly built upon the derivatives and subject to similar noise-sensitivity as the peak finding approaches.

Of all the approaches, the state of the art performance is successfully achieved by Gallo and Manduchi [13, 14]. The authors use deformable templates and perform maximum likelihood estimation on code digits. While conceptually being similar to our work, this algorithm coercively engineers the decoding process, requiring careful implementation and parameter selection. Furthermore, the authors target UPC-A (Universal Product Code A) codes explicitly making their method symbology-dependent. To the best of our knowledge there is no generic framework for decoding different barcode types.

3. PROBABILISTIC MODELING OF LINEAR BARCODES

In this section, we present a novel probabilistic model for modeling linear barcodes. Our aim is to infer the subsequent symbols given a barcode scanline that is obtained from a gray-scale image. This is a challenging task since the observed scanline can be degraded due to noise, blur, perspective/lens distortions, and non-linear deformations, which often occur in practice.

The main idea in our model is to incorporate the sequential and hierarchical information of the barcodes into a single dynamic Bayesian network. We explicitly model different layers of the hierarchy by using an H-HMM. The proposed model is very flexible and can be applied to a wide range of barcode symbologies.

We start by defining the observed scanline x_n, $n = 1, \dots, N$, where n is the pixel index, $x_n \in [0, 255]$ is the pixel intensity, and N is the length of the scanline. We assume that the scanline consists of successive non-overlapping symbols that can be of variable length, i.e., a symbol s will cover a subset of the pixels, say $x_{i:j} \equiv \{x_i, \dots, x_j\}$, and the collection of the symbols will cover the entire scanline $x_{1:N}$. Every pixel x_n belongs to a part of a symbol. An example UPC-A barcode, its corresponding scanline, and a symbol in the scanline are illustrated in Fig 2.
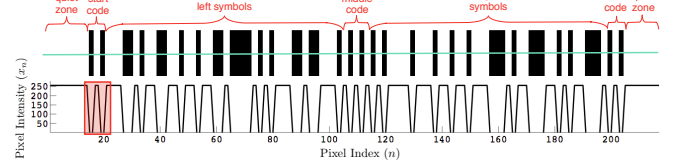
Fig. 2. A sample UPC-A barcode (top), its corresponding scanline (bottom), and a symbol in the scanline (shaded).

Our aim is to find the most-likely sequence of the symbols that are present in an observed scanline. In the sequel, we will precisely model the sequential evolution and the hierarchical structure of the symbols. Once this model is formed, the barcode string can be decoded by computing the most likely symbol sequence given the observed scanline.

In order to increase clarity, we will illustrate our model on the UPC-A[1] symbology where we will denote the UPC-A equivalents of the latent variables after defining them. Note that, our model is not specifically designed for UPC-A and is able to handle various types of symbologies.

3.1. Prior Model

We model each pixel x_n in a scanline as a noisy observation whose underlying probability distribution is governed by some latent variables. We will now proceed to the definition of these variables, that are illustrated in Fig 3(a). We define the 'symbol' variable $s_n \in D_s = \{1, \dots, S\}$, which denotes the symbol that the pixel x_n belongs to. S represents the number of symbols. For UPC-A, the set of the symbols is given as $D_s = \{1, \dots, S\} \equiv \{$*starting quiet zone, ending quiet zone, start code, end code, middle code, left digits (0, ..., 9), right digits (0, ..., 9)*$\}$ and $S = 25$. In this notation, $s_n = 4$ is equivalent to $s_n = $ 'end code'. Our ultimate aim is to find the most likely sequence of these variables.

In some barcode symbologies, such as UPC-A or EAN-13, the number of symbols that can be present in a single barcode is predetermined. For instance, in the UPC-A symbology, there must be 6 digits in between the start code and the middle code, and there must be 6 other digits in between the middle code and the end code. In order to be able to inform our model about the number of symbols that can occur in a scanline, we define the 'index' variable $m_n \in \{1, \dots, M\}$ that denotes the index of the symbol s_n. For UPC-A, it is sufficient to set $M = 6$ since the left and right digits are separated from each other by the middle code. For the variable-length barcode symbologies where the number of symbols in a barcode can vary (such as Code 128), we do not have such information, therefore we set $M = 1$.

The lengths of the symbols may differ in some symbologies. For instance, in UPC-A, the length of the start code is three base-widths (consists of three bars: black-white-black, each bar covering one base-width) whereas the length of a digit symbol is seven base-widths. On the other hand, due to warping and camera distortions,

---

[1]UPC-A is a pervasive progressive barcode symbology with 11 digits (symbols), with an added 12th digit for check-sum. It consists of alternating black-white stripe sequence with 4 possible bar widths. Each digit is encoded by a unique combination of such bars. The scannable area begins with a special start code and ends with the stop code. There are guard bars located in the middle, which optically inverses the encoding scheme to the right of the barcode. The overall length of UPC-A is 95 *base widths*, where a base width is the width of the thinnest bar.
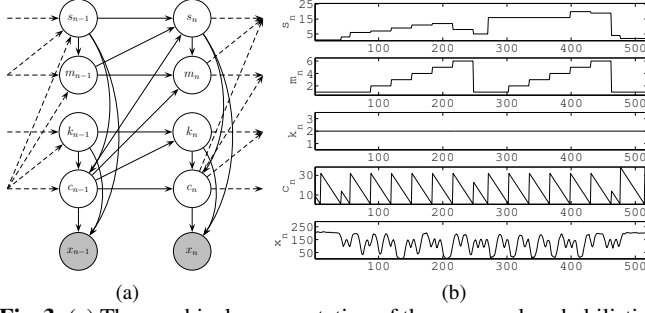
Fig. 3. (a) The graphical representation of the proposed probabilistic model. The nodes represent the random variables, the shaded nodes represent the observed variables, and the arrows determine the conditional independence structure. (b) A real barcode scanline that is correctly decoded by our method. The barcode encodes the string '013562000043'. The bottommost plot depicts the observed scanline and the first four plots depict the estimated latent variables.

the pixel-wise lengths of two symbols may differ within the same scanline even if they are supposed to have the same length. Here, we introduce the mapping $\kappa(s_n, k_n)$ that determines the length of s_n in pixels. Here, $k_n \in \{1, \ldots, K\}$ is the 'warping' variable that enables the symbols s_n to have different lengths within a scanline. For instance, suppose that the estimated base-width for a particular scanline is 5 pixels. Then, ideally, the start code would cover 15 pixels. However, we may want to allow this length to vary in the range $\{14, 15, 16\}$ in order to provide robustness to distortions. For this particular example, we need to set $K = 3$ and $\kappa(s_n = \text{'start code'}, k_n = 1 : 3) \equiv \{14 : 16\}$. We will call the expected width of a symbol as l_{avg} and define the pixel tolerance d_{avg}. We allow the expected length l_{avg} to vary in the set $\{l_{avg} - d_{avg}, \ldots, l_{avg} + d_{avg}\}$. In our experiments, we will set $K = 2d_{avg} + 1$.

In order to determine which part of s_n that x_n belongs, we define a 'count down' variable $c_n \in \{1, \ldots, \kappa(s_n, k_n)\}$, where $c_n = \kappa(s_n, k_n)$ implies that the symbol s_n starts at the pixel n and $c_n = 1$ implies that s_n is ending at pixel x_n; i.e., there will be a new symbol starting at pixel $n + 1$.

We assume that within the region of a symbol s_n, the count down variable c_n starts from the length of the current symbol $\kappa(s_n, k_n)$ and decrements by one at each pixel until it hits one. This deterministic evolution is incorporated in the model by placing a degenerate prior distribution over the count down variables as follows:

$$p(c_n|s_n, k_n, c_{n-1}) = \begin{cases} \delta(c_n - c_{n-1} + 1), & c_{n-1} \neq 1 \\ \delta(c_n - \kappa(s_n, k_n)), & c_{n-1} = 1 \end{cases} \quad (1)$$

Here, $\delta(\cdot)$ is the Kronecker-delta function where $\delta(s) = 1$ if $s = 0$ and $\delta(s) = 0$ otherwise.

Next, we assume the following prior distribution over the symbol variables:

$$p(s_n|s_{n-1}, m_{n-1}, c_{n-1}) = \begin{cases} \delta(s_n - s_{n-1}), & c_{n-1} \neq 1 \\ \tau_s(s_n|s_{n-1}, m_{n-1}), & c_{n-1} = 1 \end{cases} \quad (2)$$

where $\tau_s(s_n|\cdot)$ is the transition distribution for the symbol variables that incorporates the rules of a particular barcode symbology to the model. The prior distributions $p(c_n|\cdot)$ and $p(s_n|\cdot)$ together ensure

that the symbol variable s_n stays the same within the symbol's region (i.e., for $\kappa(s_n, k_n)$ pixels) since s_n must be equal to s_{n-1} until c_{n-1} becomes 1. In other words, during the presence of a particular symbol, the count down variable decreasingly counts down to 1. When $c_{n-1} = 1$, the symbol will transition to another symbol where this transition is governed by $\tau_s(s_n|\cdot)$. The transition distributions $\tau_s(s_n|\cdot)$ for UPC-A and CODE-128 symbologies are described in detail in the supplementary document [15].

We assume the following prior distribution on the warping variables k_n:

$$p(k_n|k_{n-1}, c_{n-1}) = \begin{cases} \delta(k_n - k_{n-1}), & c_{n-1} \neq 1 \\ \mathcal{U}(\{1, \ldots, K\}), & c_{n-1} = 1 \end{cases} \quad (3)$$

where $\mathcal{U}(\Sigma)$ denotes the discrete uniform distribution with support Σ. This distribution suggests that the length of a symbol is determined when the symbol begins. The length stays the same within that symbol.

We finally define the prior distribution of the index variables m_n as follows:

$$p(m_n|s_n, m_{n-1}, c_{n-1}) = \begin{cases} \delta(m_n - m_{n-1}), & c_{n-1} \neq 1 \\ \tau_m(m_n|s_n, m_{n-1}), & c_{n-1} = 1 \end{cases} \quad (4)$$

where $\tau_m(m_n|\cdot)$ is the transition distribution for the index variables, for UPC-A it encapsulates a straightforward collection of deterministic relations. Fig 3(b) illustrates the estimated latent variables from a real barcode scanline.

3.2. Observation Model

After defining the latent variables and their relations by using the prior distributions, we also need to relate them to the observed scanline x_n through an observation model. We follow a similar approach to [14] and define the following observation model:

$$p(x_n|s_n, k_n, c_n) = \prod_{i=0}^{1} \mathcal{FN}(x_n; \mu_i, \sigma_i^2)^{[f(s_n, k_n, c_n) = i]} \quad (5)$$

Here, $\mathcal{FN}(x; \mu, \sigma^2)$ denotes the folded normal distribution where $|x - \mu|$ is normally distributed with mean 0 and variance σ^2 and $[\cdot]$ is the indicator function where $[x] = 1$ if x is true and $[x] = 0$ otherwise. The mapping $f(\cdot)$ determines the color of the bar (black or white) corresponding to a part of a symbol, where the part is determined by the symbol s_n, the length of the symbol k_n, and the pixel index of that particular symbol c_n. Here $f(\cdot) = 0$ if the corresponding bar is black and $f(\cdot) = 1$ if the corresponding bar is white. For instance, consider a UPC-A start code of 3 pixels long (i.e., $s_n = 3$, $k_n = 1$, $\kappa(s_n, k_n) = 3$). The mapping for this symbol is defined as follows: $f(s_n, k_n, c_n = 1 : 3) \equiv \{0, 1, 0\}$, meaning that the first and the last bars are black and the middle bar is white. The parameters μ_i and σ_i^2 are the mean and the variance for different colors.

3.3. Inference

We are interested in the most likely state trajectory given all the observations. For practical purposes, we define the variable $\Psi_n \equiv [s_n, m_n, k_n, c_n]$, which encapsulates the state of the model at pixel n. The set of all possible states can be listed in a vector Ω and the state of the system at the pixel n can be represented as $\Psi_n = \Omega(j)$, where $j \in \{1, 2, \ldots, (S \times M \times K \times C)\}$ and
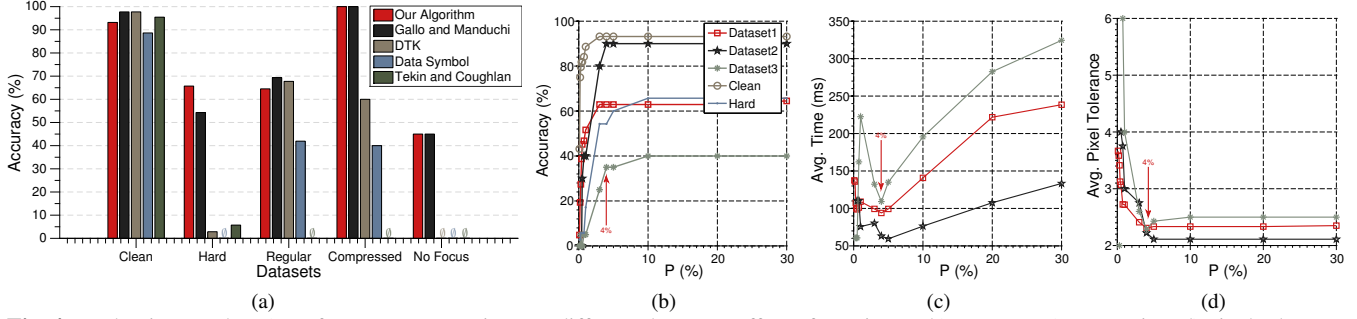
Fig. 4. Evaluation results. a) Performance comparison on different datasets. Effect of pruning on b) accuracy c) comp. time d) pixel tolerance.

$C = \max_{s,k} \kappa(s,k)$. The transition matrix of the HMM can be constructed by using Eqs 1-4. Since we reduce our model to a standard HMM, we can make use of the Viterbi algorithm in order to find the optimal state path.

For big values of S, M, K, and C the transition matrix becomes very large, but sufficiently sparse so that inference becomes tractable. However, in order to achieve real-time performance, we utilize an approximate inference schema where we keep the most likely P states at each step n and prune the rest of the states. This approach drastically reduces the computational complexity, enabling the method to work in real-time without compromising the accuracy.

4. EXPERIMENTS

4.1. Quantitative Evaluation of Accuracy

We have evaluated our algorithm on two standard datasets, discussed in [12, 14]. The widespread test bed for performance assessment of decoding algorithms is made available by Tekin and Coughlan at www.ski.org/Rehab/Coughlan_lab/Barcode. This dataset is composed of 'clean' and 'hard' to read codes, where each barcode is manually labeled. The second dataset was prepared by Gallo and Manduchi and contains images obtained in different settings: 'regular', 'compressed', and 'no focus'.

We compare the performance of our method with Gallo and Manduchi [14], Tekin and Coughlan [12], DataSymbol [16], DTK [17]. For the outputs of these algorithms we will refer to [14, 12].

In our experiments, we first estimate the basewidth of a barcode in pixels and then load the corresponding precomputed transition matrix to the memory. In the next step, we estimate the parameters of the observation model $\mu_{1:2}$ and $\sigma_{1:2}^2$ by following a similar strategy presented in [14]. We start by setting the pixel tolerance d_{avg} to 1 in the beginning and increase it by 1 until the checksum of the decoded string holds. We typically select the pruning parameter P as a percentage of the number of states, where this percentage is only around 1%, making P to be in range $[300, 2000]$ in the subject datasets. This strategy is capable of shrinking down the search space hundred folds, making pruning not just an optional nicety, but a requirement in order to achieve practicality.

Fig 4(a) comparatively demonstrates the performance of our method among the others. Note that the dataset developed in [14] requires the barcode regions to be located prior to the decoding stage. Even though we have also developed an industrial-strength barcode localization algorithm, which is beyond the scope of this study, in order to have a fair comparison, we use the same localization algorithm as in [14], where the details are presented.

All of the algorithms perform similarly on the clean dataset. The results show that Gallo and Manduchi achieve 97.72% accuracy on the clean dataset, while our method achieves 93.18% with default settings. On the other hand, in the hard dataset, we obtain 65.71% accuracy, while [14] remains at 54.28%. This result demonstrates the success of our algorithm in decoding the barcodes under challenging conditions. For a subjective evaluation, we invite the readers to check the video samples available on our webpage [15].

Note that, in certain data samples, the Gaussian assumption results in a typical model mismatch situation where the model tends to be too noise tolerant for the cases with little noise. In such cases, tolerating more noise allows the algorithm to find a more likely path, which may not exactly correspond to the correct readout. This is the reason why we report fewer correct reads in clean cases, and more in the hard. As a natural future direction, we will investigate different observation models.

4.2. Computational Aspects

Our algorithm has two stages: 1) preparation of the transition matrices and 2) the Viterbi decoding. The former is an offline stage, where the transition matrices are generated and stored. The latter is however, the main source of computational complexity, which can further be tuned. To achieve a reasonable trade-off between speed and accuracy, we resort to an approximate inference schema where we apply a simple pruning strategy. It is possible to find a reasonable trade-off as shown in Fig 4(b). In the default setting, our algorithm runs in the range $[60ms, 250ms]$, while pruning causes an exponential reduction in the computation time.

Figs 4(c) and 4(d) visualize the effects of pruning on computation time and average pixel tolerance (d_{avg}). Note that, decreasing the amount of pruning (P) increases d_{avg}, causing the algorithm to search a larger margin, consequently increasing the computational burden. For this reason, up to a certain threshold (4%), as P increases the computation time decreases. Yet, after this point, the opposite effect is observed. We refer to this noteworthy observation as the *critical valley* and use it in our experiments. As a result, pruning 4% of the entire state space still allows us to correctly decode most of the barcodes efficiently.

5. CONCLUSION

We presented an efficient H-HMM framework for decoding progressive barcode symbologies. Our accuracy was comparable to the state of the art in UPC-A datasets, while our method encapsulates the decoding process of different symbologies into a single framework. We have also shown how accuracy is traded off gracefully to gain speed as the state space is pruned. In future work, we plan on dynamically adjusting the prior parameters, exploit image gradient information and extend our work in order to cover interleaved codes.

6. REFERENCES

[1] F. von Reischach, S. Karpischek, F. Michahelles, and R. Adelmann, "Evaluation of 1d barcode scanning on mobile phones," in *Internet of Things (IOT), 2010*, Nov 2010, pp. 1–5.

[2] Jerome Swartz and Ynjiun P. Wang, "Fundamentals of bar code information theory," *Computer*, vol. 23, no. 4, pp. 74–86, Apr. 1990.

[3] William Turin and R. Boie, "Minimum discrimination information bar code decoding," in *Electrical and Electronics Engineers in Israel, 1996., Nineteenth Convention of*, Nov 1996, pp. 255–258.

[4] Karim Houni, Wadih Sawaya, and Yves Delignon, "One-dimensional barcode reading: an information theoretic approach," *Appl. Opt.*, vol. 47, no. 8, pp. 1025–1036, Mar 2008.

[5] E. Joseph and T. Pavlidis, "Bar code waveform recognition using peak locations," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 16, no. 6, pp. 630–640, Jun 1994.

[6] R. Janapriya, L. Kularatne, K. Pannipitiya, A. Gamakumara, C. Silva, and N. Wickramarachchi, "An intelligent algorithm for utilizing a low cost camera as an inexpensive barcode reader," in *Sri Lanka Association for Artificial Intelligence*, 2004.

[7] Peng-Hua Wang and Jia-Wei Ciou, "An image-based postal barcode decoder with missing bar correction," in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2013 Asia-Pacific*, Oct 2013, pp. 1–4.

[8] J Liyanage, "Efficient decoding of blurred, pitched, and scratched barcode images," in *Proceedings of the 2nd international conference on industrial and information systems*, 2007.

[9] L. Dumas, M. El Rhabi, and G. Rochefort, "An evolutionary approach for blind deconvolution of barcode images with nonuniform illumination," in *Evolutionary Computation (CEC), 2011 IEEE Congress on*, June 2011, pp. 2423–2428.

[10] Y. Lou, E. Esser, H. Zhao, and J. Xin, "Partially blind deblurring of barcode from out-of-focus blur," in *SIAM J. Imaging Sci.*, 2013.

[11] S. Krešić-Jurić, D. Madej, and Fadil Santosa, "Applications of hidden Markov models in bar code decoding," *Pattern Recognition Letters*, vol. 27, pp. 1665 – 1672, 2006.

[12] E. Tekin and J. Coughlan, "A bayesian algorithm for reading 1d barcodes," in *Computer and Robot Vision, 2009. CRV '09. Canadian Conference on*, 2009.

[13] O. Gallo and R. Manduchi, "Reading challenging barcodes with cameras," *IEEE Workshop on Applications of Computer Vision*, December 2009.

[14] Orazio Gallo and Roberto Manduchi, "Reading 1-d barcodes with mobile phones using deformable templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2011.

[15] U. Şimşekli and T. Birdal, "A unified probabilistic framework for robust decoding of linear barcodes: Supplementary document," `http://www.cmpe.boun.edu.tr/~umut/icassp15`, Accessed: 2014, [Online].

[16] "Data symbol," `http://www.datasymbol.com/`, Accessed: 2014-08-30 [Online].

[17] "Dtk," `http://www.dtksoft.com/`, Accessed: 2014-08-30, [Online].