2D Barcode Localization and Motion Deblurring Using a Flutter Shutter Camera

Wei Xu
University of Colorado at Boulder
Boulder, CO, USA
Wei.Xu@colorado.edu

Scott McCloskey
Honeywell Labs
Minneapolis, MN, USA
Scott.McCloskey@honeywell.com

Abstract

We describe a system for localizing and deblurring motion-blurred 2D barcodes. Previous work on barcode detection and deblurring has mainly focused on 1D barcodes, and has employed traditional image acquisition which is not robust to motion blur. Our solution is based on coded exposure imaging which, as we show, enables well-posed deconvolution and decoding over a wider range of velocities. To serve this solution, we developed a simple and effective approach for 2D barcode localization under motion blur, a metric for evaluating the quality of the deblurred 2D barcodes, and an approach for motion direction estimation in coded exposure images. We tested our system on real camera images of three popular 2D barcode symbologies: Data Matrix, PDF417 and Aztec Code.

1. Introduction

Barcodes are ubiquitous in product packaging and other commercial applications. Due to the limited capacity of 1D barcodes, 2D symbologies have been increasingly adopted in recent years. Whereas 1D barcodes are traditionally scanned with rotating laser illumination and linear sensor arrays, 2D barcode symbologies require imaging sensors for scanning. Image-based 2D barcode scanning presents additional challenges, notably that of projecting sufficient illumination over the area of the target. Lacking active illumination, passive 2D barcode scanners often acquire images with blur due to relative motion between the barcode and the camera.

We have developed a system for high quality capture of 2D barcode images based on coded exposure [12]. The objective of coded exposure imaging is to capture images where the removal of motion blur by de-convolution is well-posed. In order to recover the sharply-focused barcode from a coded exposure image, blur parameters must be estimated before applying de-convolution. We assume that motion blur arises from linear, constant-velocity motion, in which case we need only recover the direction and extent of an object's movement during capture. After barcode localization in a larger image, we estimate the direction of motion based on a local autocorrelation histogram. Once rectified to a reference motion direction, we estimate the extent of motion blur by applying several de-convolutions, and choosing the one which produces an image with a bi-modal intensity histogram that serves as a prior for sharply-focused barcodes. We present the details of our algorithm, and experiments which demonstrate its utility on real camera images. In addition, we demonstrate that a traditional imaging setup (i.e., without coded exposure) does not necessarily produce a useful barcode image, *even when de-blurred using an accurate blur estimate*.

2. Related Work

There have been a number of methods proposed in the literature for barcode imaging. Zhang *et al*. [14] improve the efficiency of 1D barcode localization from sharply-focused images by jointly estimating texture and shape. Lu *et al*. [9] proposed a robust 1D barcode localization approach against an uncertain 3D background using hierarchical feature classification and processing. Hu *et al*. [4] extracted the 2D barcode region from a sharply-focused image using texture direction analysis. Parikh and Jancke [11] develop an approach to localize Microsoft's High Capacity Color Barcode (HCCB), but don't address localization of black-and-white 2D barcode symbologies. None of these localization approaches consider significant motion blur, the effects of which mute high-frequency image features upon which they rely.

Motion de-blurring is a well-studied topic, and a blurred image B is generally modeled as the convolution of a blur kernel K with a latent (sharp) image L plus noise:

$$B = K \otimes L + N \tag{1}$$

where $\otimes$ denotes convolution, K is the blur kernel (also called the Point Spread Function (PSF) of the blur) caused by relative motion between the target and the camera and N denotes sensor noise at each pixel. When K is unknown,

estimation of L is known as *blind de-blurring*.

Yahyanejad and Strom [13] propose a blind de-blurring algorithm for 1D barcodes, assuming linear motion, by using an image intensity prior of clean, blur-free 1D barcodes. Unlike our method, they do not address barcode localization, and capture images using a traditional shutter. This second difference is crucial because, even when K is known, estimation of L by non-blind de-blurring is ill-posed if blur arises from linear, constant velocity motion captured through a traditional shutter. In such cases, K is a rectangle filter and de-convolution is ill-posed at a number of spatial frequencies at which the Fourier transform of K (also known as the Modulation Transfer Function (MTF)) goes to zero. Though there are various de-blurring techniques that produce visually pleasing results despite lost spatial frequencies (e.g. [5]), barcode information at these frequencies may be crucial to decoding and cannot be recovered post-hoc. To overcome this deficiency in general imaging, Raskar *et al*. [12] proposed a modification of the capture process using a fluttering shutter to make subsequent image de-convolution well-posed. The computation of de-convolution also becomes much faster due to flutter shutter since the problem is now well-posed and can be solved via direct matrix inverse.

Ding *et al.* [1] recently proposed a blind deconvolution method for flutter shutter images based on image statistics and knowledge of the shutter sequence used to capture the image. Though that algorithm produces reasonable de-blurring results in several cases, others fail due to the failure of a critical assumption: 2D barcode images do not follow the well-known statistical behaviors of natural images. In particular, whereas natural images are generally modeled as having isotropic power spectra, 2D barcode image statistics are generally skewed by the presence of high-contrast edges in orthogonal directions. We will use these features, along with spatial statistics of 2D barcodes, in our barcode localization and de-blurring algorithm.

3. System Design and Development

Our 2D barcode localization and de-blurring algorithm is intended for a checkout scenario, in which a stationary camera captures images of an object moving in the camera's field of view. Because the camera in this scenario is stationary, a reference background image can be used to perform a coarse segmentation of an image into foreground and background components. We assume that the 2D barcode has uniform blur, and that its blur arises from linear, constant-velocity motion. Note that our experiments were performed with objects scanned by hand through the camera's field of view, and that real-world deviations from this assumption were handled gracefully.

Figure 1 shows an overview of the key steps of our barcode localization and de-blurring algorithm. We first locate the barcode region using corner features, as described in Sec. 3.1. Then, blur direction is estimated from the localized barcode region using a local autocorrelation histogram, as described in Sec. 3.2. We then rotate the input image to produce a rectified one in which motion is horizontal. Next, after the barcode region is re-localized in the rectified image, we compute several de-blurred versions of the barcode region, using the code provided by Raskar *et al*. [12], with different estimates of the blur extent. We then choose the de-blurred image that best matches an intensity histogram prior, as described in Sec. 3.3. Optionally, the full image may be deblurred using the chosen blur size, and the image may be rotated back to its original orientation for display.

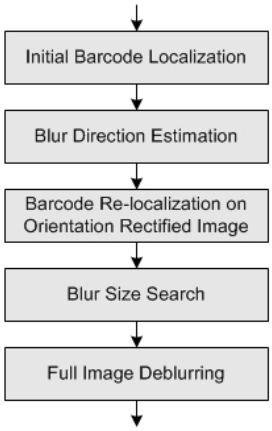


Figure 1. System overview.

3.1. 2D barcode localization

The steps of our 2D barcode localization procedure are shown in Figure 2. The first step is gamma correction, in which image intensities are linearized. We then apply a simple background subtraction, removing those pixels whose intensity is close to the background (with threshold τ_1), and those over-/under- saturated areas (with thresholds τ_2 and τ_3). Then, corner detection is applied to the gamma-corrected image, producing both a corner location map and a corresponding corner strength map. Based on these two maps, 2D tensor voting is performed in the local neighborhood of each corner location to generate a dense corner density map, which is then thresholded to find corner concentrations. This thresholded density map is then cleaned up by hole filling and image morphology operations, followed by connected components computation to generate candidate barcode regions. Finally, the barcode location is chosen from these candidate regions based on a measure considering both intensity and gradient characteristics of 2D barcodes. Among these steps, the corner detection step and the barcode region selection step have application-specific designs and are described in detail as follows.

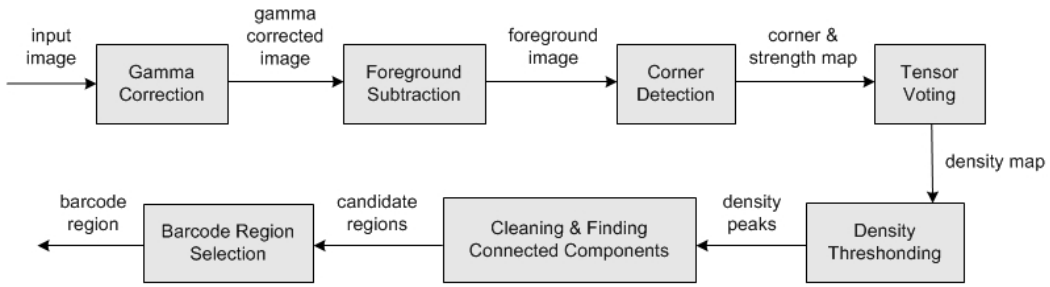


Figure 2. Procedure of 2D barcode localization. See text for details.

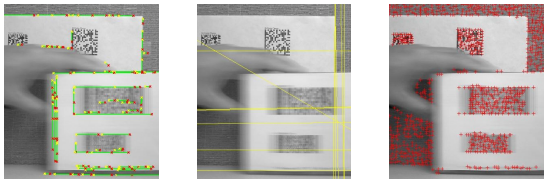
3.1.1 Corner features for 2D barcode localization

We employed corners as low-level features for localizing 2D barcode. A 2D barcode is a patten comprised of small, rectangular black patches on a white background. As a result, its gradient orientation histogram has two strong peaks at orthogonal orientations. Corner features, whose own localization is determined by orthogonality in local gradient orientation distribution, provides a natural tool for us to localize barcode area from this gradient prior.

Before deciding on the use of corner features for localization, and considering Zhang's use of Hough transform-based line detection for barcode boundaries [14], we tested and compared corners and line segments on their effectiveness for 2D barcode localization under motion blur. Figure 3 shows a typical example of how these features behave under motion blur. Figure 3(a) and 3(b) show that when moderate or severe motion blur acts upon a 2D barcode, line segment-based barcode localization becomes unreliable. 1D motion blur usually changes the edge structure on a barcode by generating false line segments along the motion direction and by breaking line segments orthogonal to the direction into smaller pieces, reducing the utility of edges for localization. One the other hand, as shown in Figure 3(c), the local density of corner features persists under motion blur. The exact locations of the corners shift, however, preventing us from directly locating a barcode's boundary corners as in [11]. Based on these observations, we use corner density to localize a 2D barcode region. We use Perter Kovesi's implementation [7] of the Harris corner detector for corner detection.

3.1.2 Choosing the barcode region

Given our computed corner density map, we estimate the barcode region from potentially several regions of dense corners. This cannot be done with a simple threshold, as there may be multiple regions in the image with comparable corner densities (e.g., regions of text). Thus we need a method to find the true barcode region from several candidate regions, and use three properties. First, since a barcode has strong black/white contrast, its appearance under motion blur will still have relatively high variance. We measure a score S_1 as:



(a) Hough transform (b) Kernel-based hough transform (c) Harris corners

Figure 3. Detection of line segments and corners on a moderately motion blurred image. (a) First 150 line segments located by MATLAB's hough transform routine, (b) First 20 lines located by the kernel-based hough transform (KHT) [3], (c) output of the Harris corner detector.

$$S_1 = Var(P) \tag{2}$$

where Var is the variance. Second, a barcode region must be a concentration of corners, thus we define S_2 as:

$$S_2 = \sum_{(x,y)\in C(P)} M(x,y) \tag{3}$$

where $C(P)$ is the set of all detected corners in the patch P, $M(x,y)$ is the corner strength (magnitude) map.

Third, the barcode region must reside in the foreground area. Considering that our initial background subtraction provides only a rough result, we treat it accordingly and define a score S_3 as:

$$S_3 = \begin{cases} R_F(P), & \text{if} \quad R_F(P) \geq \tau_F \\ 0, & \text{if} \quad R_F(P) < \tau_F \end{cases} \tag{4}$$

where $R_F(P)$ is ratio of foreground pixels in P and τ_F is a threshold. The final score is a combination of S_1, S_2 and S_3

$$S = S_1 S_2 S_3 \tag{5}$$

and the region with the maximum S score is designated as the barcode region.

3.2. Blur direction estimation

As stated above, we assume that the motion blur is uniform across the barcode region, and that it arises from 1D, constant-velocity motion, but the direction of that motion and its extent are unknown. Here we propose an approach to detecting direction of motion in a flutter shutter image *without assuming natural image statistics*. Existing frequency domain blur estimation techniques make assumptions about the image's statistics [1] or about the effects of blur [6] that do not hold on barcode images (which have non-natural image statistics) with flutter shutter motion blur (which lack telltale spatial frequencies with no contrast). Considering this, we pursue blur estimation in the spatial domain instead. We use local autocorrelation histogram to estimate the motion direction for flutter shutter images. Local autocorrelation measures the similarity of a signal to itself in a local spatial neighborhood. Given a shift $(\Delta x, \Delta y)$ from a point (x, y) on an image I, the auto-correlation function is defined as [8]:

$$f(x,y) = \sum_{(x_i,y_i)\in W} [I(x_i,y_i) - I(x_i+\Delta x, y_i+\Delta y)]^2 \quad (6)$$

where W is a local window centered at (x, y). Using the first-order Taylor expansion to approximate the shifted image $I(x_i + \Delta x, y_i + \Delta y)$,

$$I(x_i+\Delta_x, y_i+\Delta y) \approx I(x_i,y_i) + \begin{bmatrix} I_x(x_i,y_i) & I_y(x_i,y_i) \end{bmatrix} \begin{bmatrix} \Delta_x \\ \Delta y \end{bmatrix} \quad (7)$$

where $I_x(x_i, y_i)$ and $I_y(x_i, y_i)$ denote the partial derivatives of i in x and y directions respectively, $f(x, y)$ can be represented as:

$$\begin{aligned} f(x,y) &\approx \sum_W \left[\begin{bmatrix} I_x(x_i,y_i) & I_y(x_i,y_i) \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right]^2 \\ &= \begin{bmatrix} \Delta x & \Delta y \end{bmatrix} C(x,y) \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \end{aligned} \quad (8)$$

where

$$C(x,y) = \sum_W \begin{bmatrix} I_x^2(x_i,y_i) & I_x(x_i,y_i)I_y(x_i,y_i) \\ I_x(x_i,y_i)I_y(x_i,y_i) & I_y^2(x_i,y_i) \end{bmatrix} \quad (9)$$

Matrix $C(x, y)$ captures the gradient structure of the local neighborhood W. Suppose the eigendecomposition of $C(x, y)$ gives eigenvalues λ_1 and λ_2 ($\lambda_1 > \lambda_2$) and corresponding eigenvectors V_1 and V_2, then V_2 shows the major direction of local gradients. The peak of the orientation histogram of V_2 and gives an estimation of the local gradient:

$$\hat{\theta}_{grad} = max(hist(\theta(V_2)) \quad (10)$$

where $\theta(V_1)$ is the orientation of V_2.

In [8], Liu *et al*. used a weighted version of this histogram to identify motion blur over a region. However, our experiments show that it only works when the edges arising from motion blur are stronger than the local lateral edge structure. For strong patterns like 2D barcodes, this approach usually locates the direction of one side of the barcode. Here, we assume this side is orthogonal to the blur direction and the blur direction is:

$$\hat{\theta}_{blur} = \pi/2 - \hat{\theta}_{grad} \quad (11)$$

3.3. Blur size estimation

After we find the blur direction θ_{blur} and use it to rectify the blur direction to the horizontal direction, we must estimate both the extent of the blur and whether its travel in the rectified image was from right to left or from left to right. Because the shutter sequence used to capture the flutter shutter image is not symmetric, the left/right ambiguity needs to be resolved in order to apply the proper deblurring. We resolve this ambiguity and determine the extent of motion blur by producing several de-blurred images and choosing the one that most resembles a well-focused 2D barcode. We search over a large range of extents, and use a sign convention to distinguish left-to-right motion from right-to-left motion, e.g. a value of 10 represents a blur extent of 10 pixels of left-to-right motion, and -10 represents an extent of 10 pixels of right-to-left motion.

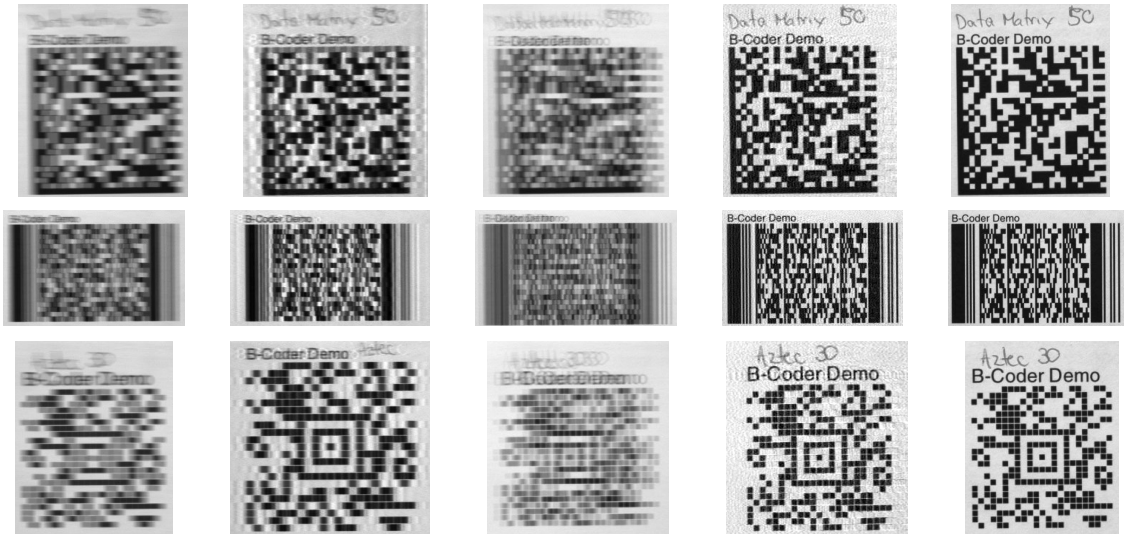
We adopt the barcode image quality metric proposed by Yahyanejad and Strom [13] to evaluate the deblurring result, and to choose between the various de-blurred image. This metric was initially designed for 1D barcode deblurring and was based on the observation that the intensity histogram of clear 1D barcode line is bi-modal. Here we extended it to operate on 2D barcode regions. Given an test image I, it is first segmented into two parts:

$$I = I_1 \cup I_2 \quad \text{where} \quad \begin{cases} I_1 = \{I(x) | I(x) > \mu_I\} \\ I_2 = \{I(x) | I(x) < \mu_I\} \end{cases} \quad (12)$$

where μ_I is the mean intensity of I. The quality metric is then computed as:

$$Q(I) = \frac{\sigma_{I_1} + \sigma_{I_2}}{(\mu_{I_1} - \mu_{I_2})^2} \quad (13)$$

where μ_{I_1}/μ_{I_2} and $\sigma_{I_1}/\sigma_{I_2}$ are mean intensity and variance of I1/I2 respectively. This metric favors an image intensity histogram with two peaks where intra-peak variance is small and inter-peak distance is large. Our experiments show this metric works when the input does not break the



(a) Traditional image (b) Deblurring of (a) (c) Flutter shutter (d) Deblurring of (c) (e) Still image

Figure 4. **Comparisons between flutter shutter and regular image deblurring**. Rows are, from top to bottom, barcodes of Data Matrix, PDF417 and Aztec. Column (a) is a blurred barcode captured using a regular camera and (b) is (a) de-blurred using the Lucy-Richardson algorithm; (c) is the barcode captured by our flutter shutter camera and (d) is its deblurred version; (e) is a still, blur-free image of the barcode for comparison.

the bi-model assumption. In practice, the barcode localization scheme described in the previous section guarantees this pre-condition.

Though other techniques (e.g. coordinate descent [2]) use energy minimization and a greedy search strategy to locate an optimal solution, we have found that they cannot be applied in our case due to the non-convexity of the quality metric S. In order to avoid getting stuck in a local optimum, we instead use an exhaustive search strategy which guarantees the global optimum solution. Specifically, given a rectified 2D barcode and a range of the parameters of the blur kernel, we test every value in this range by using it to deblur the barcode (using the code provided with [12]). For each such de-blurred barcode, we measure the quality metric S, and choose the one with the best (lowest) value.

4. Experiments

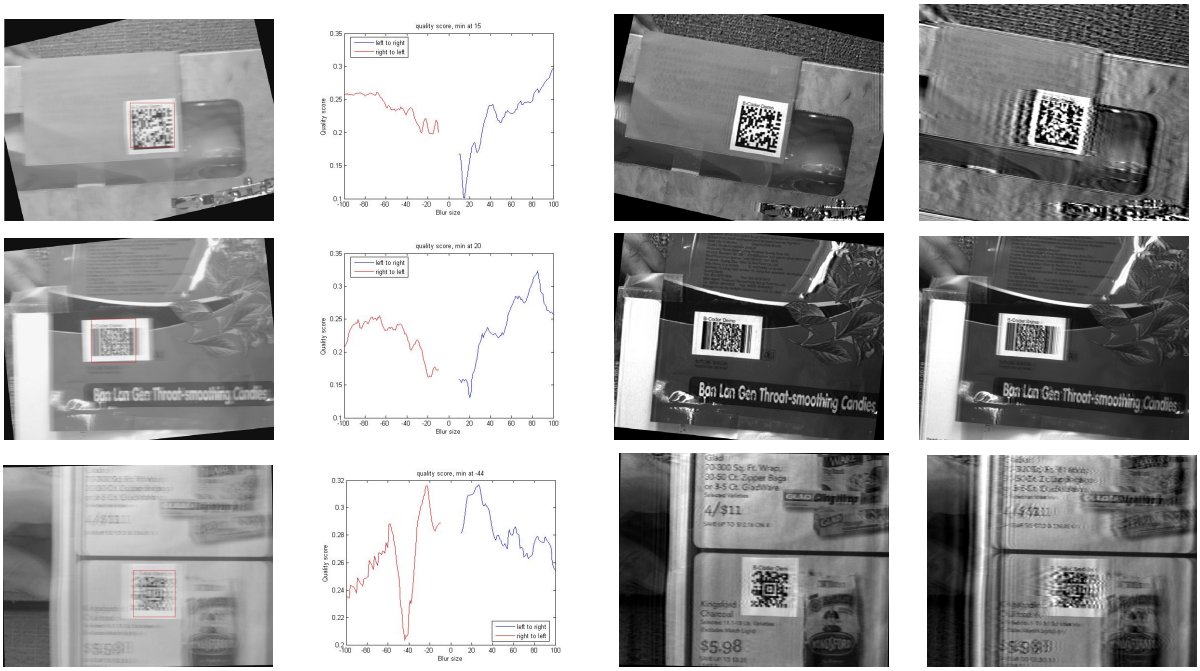
Our flutter shutter camera is built on a Point Grey Flea®2 camera, and uses custom software to capture a coded exposure image following a pre-designed binary timing sequence [10]. We used the reference code from [12] for deconvolving flutter shutter images once they are rotated into the rectified orientation.

Our first experiment was to determine whether coded exposure images are more useful for barcode recognition than traditional images. To that end, we collected motion-blurred images using both techniques, none of which could be directly decoded. We then de-blurred them using manually determined blur parameters. Figure 4 shows the comparison on three type of 2D barcodes: Data Matrix, PDF417 and Aztec. We found that, while all of the de-blurred flutter shutter images could be de-coded, only the Aztec could be de-coded from the de-blurred traditional shutter image. Note that all decoding was performed using proprietary decoding software.

Our subsequent experiments are to illustrate the barcode localization and blur estimation system. Figure 5 shows multiple examples of running our system on real 2D barcodes and a comparison to a recently proposed flutter shutter blind deblurring approach [1]. Because of using natural image statistics, the algorithm of [1] produces acceptable results in certain cases (e.g. second row of Figure 5(d)), but fails when sufficient non-barcode texture is lacking. By contrast, we have found that our system provides consistently high-quality results. More experimental results are provided in the supplemental material of this paper.

5. Conclusions and Future Work

In this paper we described a 2D barcode localization and motion deblurring system. The system adopts a flutter shutter camera for its high deblurring quality, and focuses on the barcode region to exploit the strong intensity patterns of a 2D barcode. We developed a sequence of approaches, from barcode localization to blur estimation, to realize our system. The high deblurring speed of a flutter shutter images and our processing only the small barcode region make it



(a) Localized barcode (b) Quality scores (c) Our deblurring result (d) Deblurring result of [1]

Figure 5. Deblurring examples of barcodes of Data Matrix, PDF417 and Aztec symbologies. Column (a) is the orientation rectified image with localized barcode marked in red. (b) Plots deblurring quality scores (eq. 13) during blur estimation. Note the curves are non-convex, which makes it hard for a greedy search strategy to successfully find the true blur size. (c) and (d) show the outputs of our system and of [1] respectively.

possible to thoroughly search over a large range of blur parameters for the true value. The algorithm is also highly parallelizable, in that each de-blurred image can be computed without regard to the others.

Experiments show that the system works well, including those where the motion deviates from our assumption of constant velocity, linear motion. Using a large search range of 180 different values, our unoptimized MATLAB implementation of the algorithm takes ≈ 30 seconds to deblur an 800×600 barcode image on a 3.47GHz desktop computer. We observe that most of the computing time is spend on testing large blur sizes. One direction of future work is to develop an approach to adaptively narrow this search range according to the input. Another direction is to make the system more robust by relaxing some of the assumptions on motion types.

Acknowledgements

Flea®2 is a trademark of Point Grey Research, Inc. All brand names and trademarks used herein are for descriptive purposes only and are property of their respective owners.

References

[1] Y. Ding, S. McCloskey, and J. Yu. Analysis of motion blur with a flutter shutter camera for non-linear motion. In *Proc. 11th European Conference on Computer Vision*, 2010.

[2] R. Fergus, B. Singh, A. Hertzmann, S. Roweis, and W. Freeman. Removing camera shake from a single photograph. *ACM Transaction on Graphics*, 25(3):787–794, 2006.

[3] L. A. F. Fernandes and M. M. Oliveira. Real-time line detection through an improved hough transform voting scheme. *Pattern Recognition*, 41(1):299–314, 2008.

[4] H. Hu, W. Xu, and Q. Huang. A 2D barcode extraction method based on texture direction analysis. In *Proc. Fifth Int'l Conf. on Image and Graphics*, pages 759–762, 2009.

[5] P. Jansson. *Deconvolution of Image and Spectra, 2nd ed.* Academic Press, 1997.

[6] H. Ji and C. Liu. Motion blur identification from image gradients. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[7] P. D. Kovesi. MATLAB and Octave functions for computer vision and image processing. `http://www.csse.uwa.edu.au/~pk/research/matlabfns`.

[8] R. Liu, Z. Li, and J. Jia. Image partial blur detection and classification. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–8, 2008.

[9] X. Lu, G. Fan, and Y. Wang. A robust barcode reading method based on image analysis of a hierarchical feature classification. In *Proc. 2006 IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, pages 3358–3362, 2006.

[10] S. McCloskey. Velocity-dependent shutter sequences for motion deblurring. In *Proc. 11th European Conference on Computer Vision*, 2010.

[11] D. Parikh and G. Jancke. Localization and segmentation of a 2D high capacity color barcode. In *Proc. 2008 IEEE Workshop on Applications of Computer Vision*, pages 1–6, 2008.

[12] R. Raskar, A. Agrawal, and J. Tumblin. Coded exposure photography: motion deblurring using fluttered shutter. *ACM Transaction on Graphics*, 25(3):795–804, 2006.

[13] S. Yahyanejad and J. Strom. Removing motion blur from barcode images. In *Proc. IEEE International Workshop on Mobile Vision (in Conjunction with CVPR'2010)*, 2010.

[14] C. Zhang, J. Wang, M. Y. Shi Han, and Z. Zhang. Automatic real-time barcode localization in complex scenes. In *Proc. 2006 IEEE Int'l Conf. on Image Processing*, pages 497–500, 2006.

A. Parameter Settings

There are a few application-specific parameters existing in our approach, including the density voting radius R_{vote}, the density threshold T_{den} and the blur size search range σ_K. In all our experiments, we set $R_{vote} = 50$ and $\sigma_K = [-100, -10] \cup [10, 100]$. $R_{vote} = 50$ is an experimental setting decided by the minimum image distance between barcode region and other candidates. $\sigma_K = [-100, -10] \cup [10, 100]$ is also a sufficiently large range in practice.

The only parameter that needs fine tuning is T_{den} which is a percentage threshold. In practice we find $T_{den} = [0.01, 2.0]$ (which means keeping the top 1%$\sim$20% most dense votes) is a reasonable range for accurate localization of the 2D barcode. However, this is still not enough for a fully automatic system and we have exploited other means to bypass this manual parameter selection. What we tried first is the bounding rectangle expanding scheme developed by Parikh and Jancke for HCCB barcode [11], however, this scheme does not always work for a regular 2D barcode. We finally took the following strategy to bypass this parameter turning process at the cost of some localization accuracy: We simply used a middle-range threshold by setting $T_{den} = 0.03$, which will produce a localization of the majority of the barcode, and may also arouse false alarms of non-barcode regions. We rely on the the score metric described in Sec. 3.1.2 of the paper to identify the true barcode region from competitors (with $\tau_F = 0.9$).

Other parameters are set as follows: we set $\gamma = 2.2$ for inverse gamma correction of the input image as in [2]. We set the magnitude threshold $\tau_{corner} = 200$ for the Harris corner detector. This is a pretty low threshold considering corner magnitude is attenuated by blur. For foreground subtraction, we set $\tau_1 = 30$, $\tau_2 = 254$ and $\tau_3 = \text{mean}(I) * 0.9$. For local autocorrelation histogram computation, we use a sampling window W of size 11×11 pixels. All the adopted image morphology operations (including erosion and dilation) use a square mask of size 10×10 pixels.