Coded Exposure Deblurring: Optimized Codes for PSF Estimation and Invertibility

Amit Agrawal Yi Xu*
Mitsubishi Electric Research Labs (MERL)
201 Broadway, Cambridge, MA, USA
[agrawal@merl.com,xu43@cs.purdue.edu]

Abstract

We consider the problem of single image object motion deblurring from a static camera. It is well-known that deblurring of moving objects using a traditional camera is ill-posed, due to the loss of high spatial frequencies in the captured blurred image. A coded exposure camera [17] modulates the integration pattern of light by opening and closing the shutter within the exposure time using a binary code. The code is chosen to make the resulting point spread function (PSF) invertible, for best deconvolution performance.

However, for a successful deconvolution algorithm, PSF estimation is as important as PSF invertibility. We show that PSF estimation is easier if the resulting motion blur is smooth and the optimal code for PSF invertibility could worsen PSF estimation, since it leads to non-smooth blur. We show that both criterions of PSF invertibility and PSF estimation can be simultaneously met, albeit with a slight increase in the deconvolution noise. We propose design rules for a code to have good PSF estimation capability and outline two search criteria for finding the optimal code for a given length. We present theoretical analysis comparing the performance of the proposed code with the code optimized solely for PSF invertibility. We also show how to easily implement coded exposure on a consumer grade machine vision camera with no additional hardware. Real experimental results demonstrate the effectiveness of the proposed codes for motion deblurring.

1. Introduction

Motion deblurring is an important problem for computer vision applications and consumer photography. Motion blur in photographs manifests due to camera motion (e.g., hand-shake), object motion or a combination of both. In this paper, we are concerned with deblurring images of fast moving objects captured from a static camera. There has been significant amount of research on estimating the point spread function (PSF) of an optical system and recovering a sharp image from captured noisy blurred image. Blind deconvolution techniques attempt to estimate the PSF from the given blurred image itself. It is well-known that if the Fourier spectrum of the PSF contains zeros, simple inverse filtering will amplify noise and produce ringing artifacts in the deblurred image. Several techniques using image priors and noise models such as Wiener filtering [16] and Richardson-Lucy algorithm [19, 13] have been proposed to handle such non-invertible PSF's. See [6] for details.

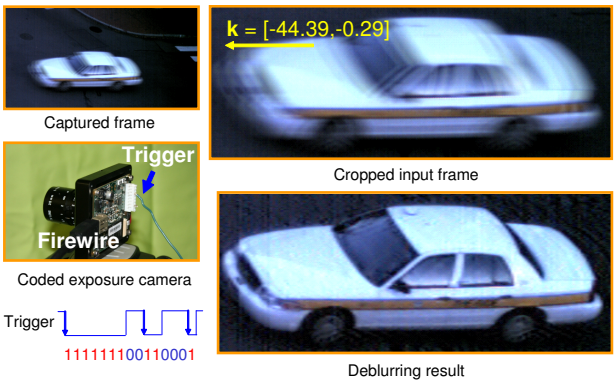

Figure 1. Using a carefully designed code, one can achieve *both* PSF estimation and invertibility for motion deblurring via coded exposure camera. Photo of fast moving car (top) was captured using the coded exposure camera with code 1111111100110001 (bottom left). Deblurring result (bottom right) using estimated motion PSF shows the effectiveness of the proposed codes.

The idea of engineering the motion PSF to make it invertible and simplify motion deblurring was first proposed in [17]. The key concept was to open and close the shutter within the exposure time to preserve high spatial frequencies in the captured image, using a carefully designed binary code. The code was chosen so that the resulting PSF does not have any zeros in its frequency transform and is as broadband as possible. In contrast, a traditional camera keeps the shutter open for the entire exposure duration leading to a low pass PSF, which is not invertible. This was further extended to handle out of focus blur using coded

*Yi Xu is currently a graduate student at Purdue University.

aperture in [23].

However, the problem of PSF estimation still remains even if the PSF is made invertible. One needs to estimate the motion of moving objects in order to deblur them. In [17], PSF is estimated by manually finding the motion direction. Then the blurred image is rectified and deblurring is performed for several blur sizes, and the best visual result is chosen as output. Our goal is to automate PSF estimation and we are inspired by the recent work in this area using transparency (alpha matting) [7, 2]. Specifically, we use the motion from blur (MFB) approach presented in [2].

Although these approaches were demonstrated for estimating PSF using a traditional camera, we show that motion from blur constraint also holds for coded exposure, for those parts of blur that correspond to ones in the code (Figure 3). In case of analyzing motion blur as alpha matting, the 'foreground' corresponds to blurred object. Since the estimation of transparency or alpha matting requires locally smooth foreground/background, the optimal code for invertibility does not work well for PSF estimation because it leads to non-smooth blur and foreground. Furthermore, since the MFB algorithm relies on locally smooth alpha values to compute alpha gradients, PSF estimation becomes even more difficult. Our key idea is to find an *invertible* code which also results in smooth blur for some *parts* of the blur profile to help in PSF estimation.

At first glance, it might appear that both good PSF estimation and PSF invertibility cannot be simultaneously achieved. In [9], a 2D code for coded aperture was designed so as to intentionally insert zeros in the frequency spectrum of the PSF. The locations of zeros were used to estimate the PSF scale. However, inserting zeros in the frequency spectrum of the PSF inherently leads to non-invertible PSF, making deconvolution ill-posed. In addition, in presence of noise, deciding which frequency magnitude is zero is extremely unstable. [9] uses several heuristics, image priors and a learning based approach for locating zeros in the frequency spectrum of the blurred image. In this paper, we show that one does not need to sacrifice invertibility for PSF estimation. Both can be simultaneously achieved by careful code selection, but with a slight increase in the deconvolution noise compared to the optimal invertible code. Our approach does not use any training data or learning methods.

Contributions: Our contributions are as follows.

- We show that the motion from blur constraint also holds for those parts of coded blur that correspond to ones in the code with a constant scale factor given by the ratio of the number of ones to the code length.
- We demonstrate that smooth blur is easier to estimate and the optimal invertible code could worsen PSF estimation as it leads to discontinuities in the blur profile.
- We outline criteria for both good PSF estimation capability and invertibility and propose two search methods to quickly find the code for large code lengths.
- We show how coded exposure can be implemented on available machine vision cameras with no additional hardware.

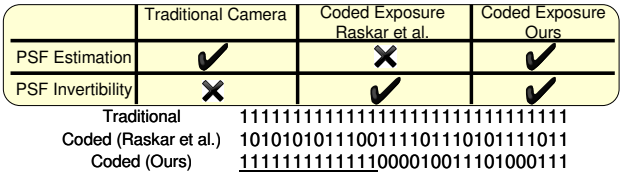

Figure 2. A traditional camera is good for PSF estimation since it results in smooth blur, but has poor deblurring performance. While coded exposure makes PSF invertible, it introduces discontinuities in blur that makes PSF estimation difficult. By carefully choosing the code, we achieve both PSF estimation and invertibility.

Limitations: Our method is limited to low-frequency background due to blur estimation using alpha matting, which requires smooth background. In addition, using a single image similar to [17] also limits the background, since high-frequency background results in noisy alpha matte and leads to deblurring artifacts at the layer boundaries. Using multiple images or better alpha matting techniques tailored to motion blur would allow handling such cases. Our method is also limited to linear motion model. Although restrictive, linear motion model can handle a broad class of spatially varying motions that can be rectified to linear motion. In addition, MFB [2] is capable of handling broader classes of object motions, such as rotational and non-parametric motion blur and our approach would benefit from it.

1.1. Related work

Coding and modulation: Multiplexing techniques are becoming popular for several computer vision and graphics applications. Schechner and Nayar [20] use illumination multiplexing using Hadamard codes to improve the signal to noise (SNR) ratio in image capture. This was extended in [18] to include the effect of sensor noise and saturation. Coded aperture techniques use MURA codes [4, 1] to improve capture SNR in non-visible imaging and invertible codes for out-of-focus deblurring for photography [23]. Zomet and Nayar [26] replace the conventional lens of the camera with parallel light attenuating layers whose transmittances are controllable in space and time for useful applications such as split field of view and instantaneous pan and tilt. Light field capture using frequency domain multiplexing was proposed in [23] and using multiplexed coded aperture in [12]. Nayar *et al*. [15] proposed programmable imaging by using a digital micro-mirror device (DMD).

PSF manipulation: Two important classes of techniques involve modifying the PSF and make it (a) invertible or (b) invariant. Wavefront coding methods [3] use a

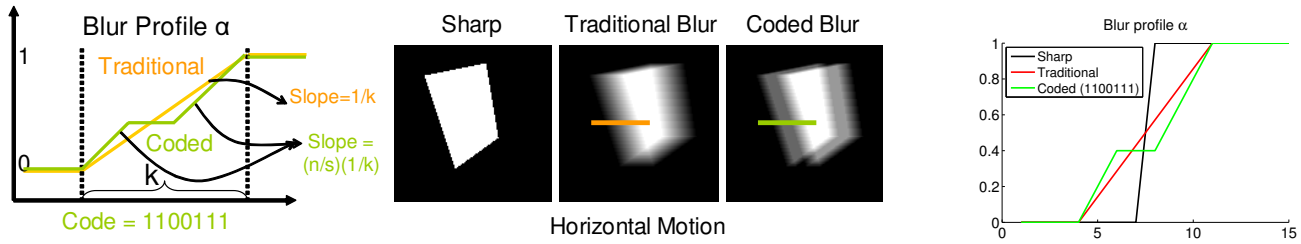

Figure 3. (Left) Motion from blur constraint holds for coded exposure for those parts of the blur which correspond to ones in the code. For traditional camera, slope of α equals $1/k$. For coded exposure, slope increases by a factor of n/s. (Middle) Synthetic example showing a polygon shaped object moving horizontally. (Right) Corresponding blur profiles obtained from blurred synthetic images.

cubic phase plate in front of the lens to make defocus PSF invariant to depth. This enables the use of a single deconvolution filter to recover the sharp image without knowing the depths in the scene. Nagahara *et al*. [14] move the sensor in the lateral direction during image capture to make the defocus PSF invariant to depth. However, the drawback is that the typical plane of focus due to lens is also blurred. By moving the camera in a parabolic fashion, Levin *et al*. [11] make the motion PSF approximately invariant to the speed of the object. Similarly, the drawback is that static parts of the scene are also blurred. Coded exposure [17] and coded aperture [23] techniques make the PSF invertible so that the resulting deconvolution process becomes well-posed. However, PSF estimation is still required.

PSF estimation and deblurring: Recent interest in computational photography has spurred significant research in PSF estimation and deblurring algorithms. Fergus *et al*. [5] use natural image statistics to estimate the PSF from a single blurred image. Joshi *et al*. [8] estimate non-parametric, spatially-varying blur functions by predicting the sharp version of a blurry input image. Yuan *et al*. [24] use both a short exposure image and a long exposure image to estimate the motion PSF and use them simultaneously for deblurring to handle camera shake. Recent work on deblurring algorithms [21, 25] have shown excellent results on images corrupted due to camera shake.

2. Blur estimation using alpha matting

Let $s(x, y)$ denote the image of the object if it was static and $h(x, y)$ be the motion PSF. Let $M(x, y)$ be a binary indicator function for the object[1]. When the object moves in front of the background $b(x, y)$, the captured blurred image I is given by the sum of blurred foreground object and partial background [17]

$$I = s * h + (1 - M * h)b. \tag{1}$$

Comparing with the familiar matting equation $I = \alpha F + (1 - \alpha)B$ [22], we get

$$B = b, \quad \alpha = M * h, \quad F = (s * h)/(M * h). \tag{2}$$

[1]We assume that the moving object is opaque and in sharp focus.

Note that the 'foreground' for the matting algorithm is not the actual object s, but the blurred object which depends on the PSF h. Although matting algorithms can handle complex α (such as hair, smoke etc.) and thus discontinuous I, they require both the foreground and background to be locally smooth or low frequency. For a traditional camera, PSF is a box function (h is low pass) and results in smooth foreground F. Previous motion blur estimation algorithms based on alpha matting have shown very good results on images captured using a traditional camera. However, deblurring is ill-posed due to h being low pass.

2.1. Coded exposure camera

The key idea of coded exposure is to open and close the shutter according to a pseudo-random binary code to preserve high spatial frequencies in the captured blurred image. Thus, the motion PSF h becomes broadband and deblurring is well-posed. However, this results in high frequency variations in the blur profile. Thus, alpha matting is not robust due to non-smooth 'foreground' and PSF estimation using transparency is also hard due to non-smooth alpha. Our goal is to design the code so that certain parts of the code result in smooth blur to help matting and PSF estimation, while overall the code is still invertible for good deblurring.

For rest of the paper, let $c(x)$ be the code, n be the code length, s be the total number of ones, t be the number of transitions, and r be the maximum number of continuous ones in the code. A traditional camera can also be characterized as a coded exposure camera with $s = r = n$ and $t = 0$. Note that the coded exposure camera loses light by a factor of $\frac{n}{s}$. The linear system corresponding to motion blur is given by $A\mathbf{x} = \mathbf{b}$, where A is the motion smear matrix, $\mathbf{x}$ is the unknown sharp image, and $\mathbf{b}$ is the blurred photo. Similar to [17], we use $f_{noise} = \texttt{mean}(A^T A)^{-1}$ for evaluating the increase in deconvolution noise.

2.2. Motion from blur

We first show that motion from blur constraint also holds for coded exposure camera. The constraint is given by [2]

$$\nabla\alpha \cdot \mathbf{k} = \pm 1, \tag{3}$$

where $\mathbf{k} = [k_x, k_y]$ denotes the blur vector[2]. This constraint assumes h to be a box filter (traditional camera). For coded exposure, h is a sum of shifted box functions of varying sizes. Thus, this constraint still holds for each set of continuous ones in the code. If the object moves with constant speed, the motion from blur constraint changes to

$$\nabla\alpha \cdot \mathbf{k} = \pm\frac{n}{s}, \quad \texttt{if} \quad c(x) = 1, \tag{4}$$

since PSF h is normalized to 1 ($\int h = 1$). When the code is zero, no light is integrated, and hence α remains constant ($\nabla\alpha = 0$) within that time period. Only for those parts of code which are 1, the constraint holds as shown in Figure 3.

2.3. Codes with similar deblurring performance

Codes having the same deblurring performance could differ significantly in their resulting blur profiles. Consider two $n = 31$ codes:

$$C_1 = 10101010111001111011101011111011,$$
$$C_2 = 11111111111110000100111010001111.$$

Both codes have the same number of ones ($s = 21$), and thus would allow the same amount of light. Figure 4 shows the magnitude of the frequency transform for both codes after zero padding. The minimum frequency transform magnitude is the same for both codes. In fact, the increase in deconvolution noise for C_1 and C_2 are 19.7 and 20.09 dB respectively (compared to 35.7 dB for traditional camera). Thus, these two codes will result in similar deblurring performance. However, they result in significantly different blur profiles. The number of transitions, t, for C_1 equals 18 compared to 8 for C_2 and C_2 has a long continuous string of ones ($r = 13$). As shown in Figure 4, the blur profile corresponding to C_2 will be smooth at one end, with minimum number of discontinuities compared with the blur profile corresponding to C_1. Thus, for the same deblurring performance, one could possibly choose a code which results in smooth blur for some *parts* of the entire motion blur. Since most alpha matting algorithms require local smoothness within a neighborhood (e.g., 3×3), minimizing the number of transitions in the code will reduce discontinuities in foreground and result in better alpha map estimation. Moreover, the smoothly changing alpha values within the same region also allows better gradients computation; thus facilitates PSF estimation.

3. PSF estimation and deblurring results

In this section, we show results and comparison for PSF estimation and deblurring on real datasets using traditional camera and coded exposure with codes C_1 and C_2. In all

[2]Assuming constant velocity object motion in image plane.

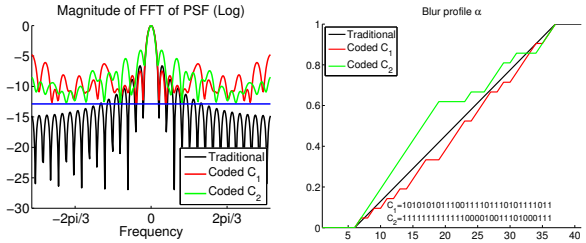

Figure 4. Two different codes C_1 and C_2 having same deblurring performance but different blur profiles. (Left) The magnitude of Fourier transform shows that although the minimum for C_1 and C_2 are same (blue line), C_1 attenuates low frequencies much more than C_2. (Right) C_2 has small number of transitions and long consecutive string of ones. This results in smooth blur profile for C_2 on one side which helps in PSF estimation. Note that since alpha is normalized to $[0, 1]$, the slopes of blur profiles for traditional and coded exposure are different.

results where we compare with a traditional camera, its exposure time is reduced by a factor of $\frac{n}{s}$ to ensure the *same* light level. The traditional camera image thus will have reduced blur by the same factor.

The PSF estimation algorithm follows [2], where first alpha matting is performed (using Levin *et al*. [10]) to obtain the alpha values. We further improve the MFB algorithm to handle the aperture problem as described below. As shown in [2], every pixel whose alpha gradient is non-zero, gives information about the blur direction and magnitude. In [2], first a set of locally consistent pixels are found and then RANSAC is applied to estimate the blur using (3) by computing the inliers.

Weighted least square (WLS) estimation: To handle the aperture problem, blurred edges of different edge directions should be present in the image as described in [2]. However, [2] uses all inliers equally to estimate the blur. We propose to cluster the inliers based on the α-gradient values, since α_x and α_y together give information about the edge direction. For example, if both α_x and α_y are larger than zero, the pixel belongs to an edge that is facing top right. Specifically, we divide the inliers into 8 clusters depending on whether the gradient α_x, α_y are $> \tau$, $< -\tau$, or $\in [-\tau, \tau]$, where τ is a threshold (e.g. 0.02). We ignore the cluster where both α_x, α_y are $\in [-\tau, \tau]$, since those pixels do not give any useful information in presence of noise. Then, we simply perform a WLS estimate on inliers, where the weights are the inverse of cluster sizes. This ensures that edges having different directions get equal weights in blur estimation, so that the estimation is not biased towards a particular edge direction.

Object moving at an angle: Figure 5 shows results on a toy motorcycle, where the motion is non-horizontal in the image plane. The captured blurred photos and deblurred results are shown in top and bottom rows respectively for traditional and coded exposure cameras using C_1 and C_2 codes. Note that the estimated PSF using C_2 is close to

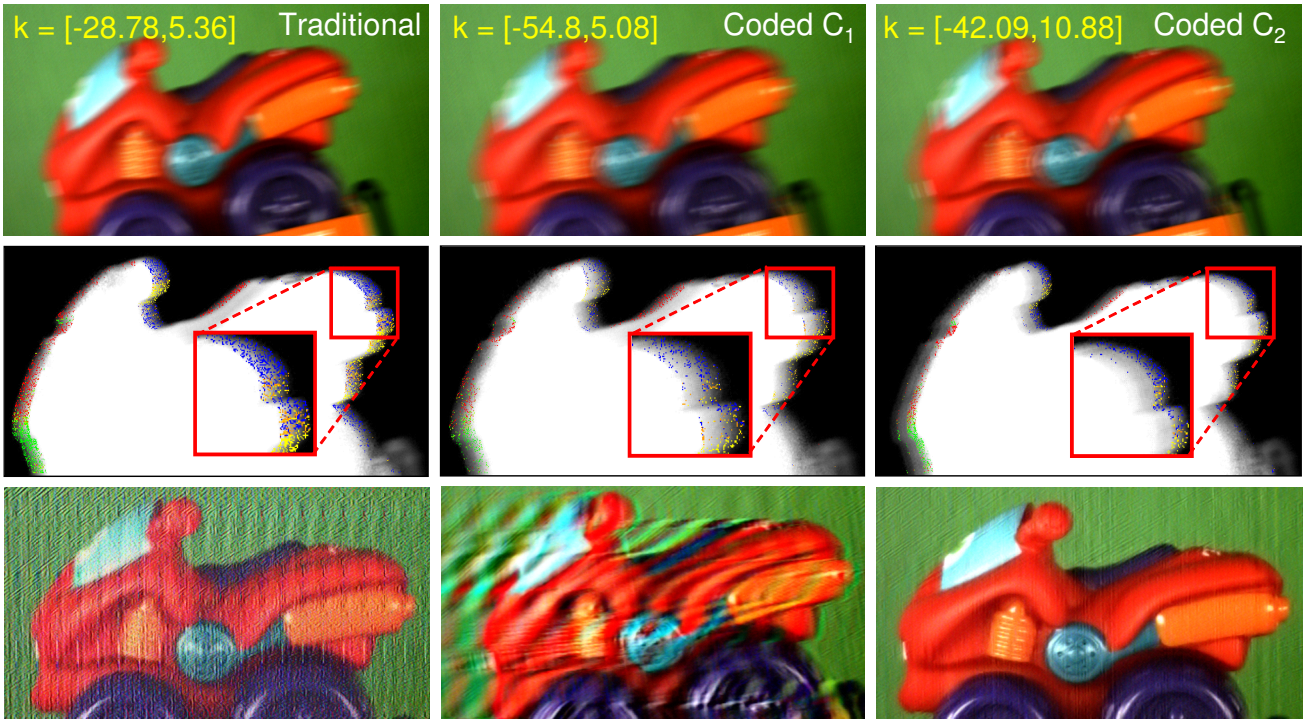

Figure 5. Motorcycle moving at an angle. (Top) Blurred photos. (Middle) Alpha maps with inliers. (Bottom) Deblurred results. PSF estimation for traditional camera is good but deblurring is poor due to non-invertible PSF. Bad PSF estimation for code C_1 leads to poor deblurring. For C_2, estimated PSF is good, as proved by the deblurring result. The ratio between the lengths of motion vectors **k** for coded and traditional exposure should be n/s=31/21=1.47. It is 1.48 for C_2, 1.88 for C_1. Input images are rotated using the estimated motion angle before deblurring to bring the motion horizontal. For C_1, incorrectly estimated angle cannot be used to rectify the input image.

ground truth, as shown by the good deblurring result. PSF estimation for traditional camera is also good but deblurring is bad due to PSF being non-invertible. Figure 5 (middle row) also shows inliers (different color for each cluster) obtained from MFB algorithm. For traditional camera, inliers span *all* parts of the blur as expected; while, for coded blur, the α-motion blur constraint only holds for those parts of the blur that correspond to 1's in the code, as described in Section 2.2. Note that for C_2, most of the inliers are present on one end of the blur corresponding to the long string of 1's in C_2. However, for C_1, inliers are scattered all over the blur which shows that alpha estimation and MFB algorithm was not successful. Figure 6 also shows the ground truth photo and the deblurring result for C_1 if the PSF estimated using C_2 is used. This clearly demonstrates that the deblurring performances for C_1 and C_2 are similar. However, C_2 assists in PSF estimation, while C_1 does not.

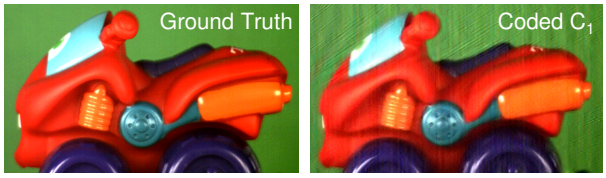

Figure 6. Motorcycle. (Left) Ground truth sharp image. (Right) Deblurring result for C_1 using the motion PSF estimated from C_2 shows that the deblurring performances are similar for C_1 and C_2, but PSF estimation fails using C_1 (see Figure 5 middle image in the bottom row.)

Non-uniform background: Figure 7 shows an example of a moving sticker in front of a non-uniform background. Again note that the estimated inliers for C_2 are restricted to those parts of the blur, which correspond to the long string of 1's. The deblurring results demonstrate that the motion estimation is good for C_2, but poor for C_1.

Complex object shape: Figure 8 shows another example on an complex shaped action figure. Even though the shape is complex, our algorithm successfully estimates the PSF using C_2 since it produces partial smooth blur. Fine features are recovered on the action figure using C_2 code.

Outdoor scene: Our approach also works on challenging outdoor scene as shown in Figure 1. Since the car is far away, it is assumed to be moving parallel to the image plane. A $n = 15$ code 111111100110001 with $r = 7$ was used to capture the photo. Note that the deblurring result recovers sharp features on the car.

4. Implementation and analysis

In [17], coded exposure was implemented using an external ferro-electric shutter placed in front of the lens of a SLR camera. The ferro-electric shutter from *DisplayTech*

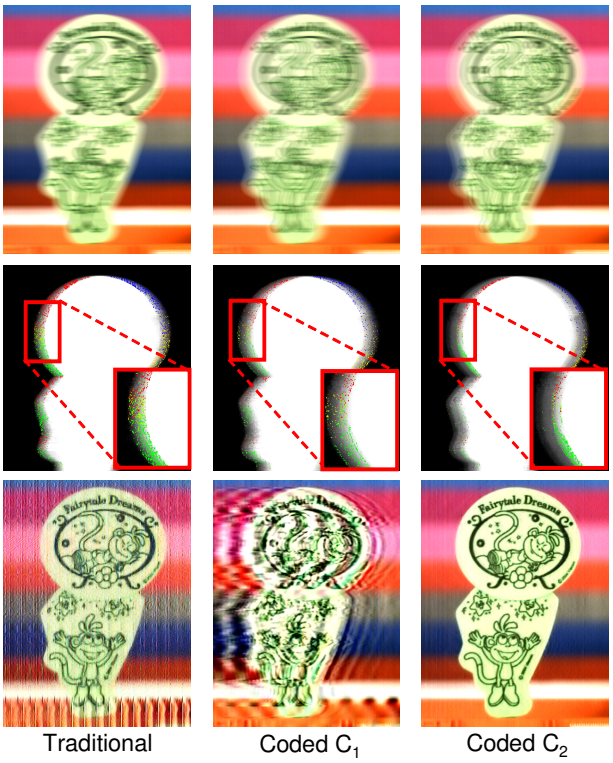

Figure 7. Non-uniform background. (Top) Blurred photos. (Middle) Alpha maps with inliers. Each color shows one of the 8 clusters. (Bottom) Deblurring results. Estimated **k** are [-26.56, 0.67], [-49.87, 1.13] and [-37.75, -0.33] for traditional, C_1 and C_2 respectively. The magnitude of the estimated motion vector for C_2 is 1.42 times of that of traditional exposure, close to the theoretical factor of n/s=31/21=1.47.

costs $\approx$ \$500 and requires an external micro-controller for control. In addition, the external shutter leads to vignetting in images and loses light even when it is transparent due to polarization. Instead, we implemented coded exposure on a consumer grade machine vision camera by *on-chip* fluttered integration with zero additional cost and avoided all the above issues. This can be achieved with any camera that supports IEEE DCAM Trigger mode 5. This trigger mode supports multiple pulse-width trigger with a *single* readout. We use the Dragonfly2 camera from PointGrey (`www.ptgrey.com`) (Figure 1). The camera is triggered using the parallel port of a PC. Each bit of the code corresponds to 1 ms of the exposure time in our implementation. Thus, for $n = 31$, total exposure time was 31 ms. To implement a particular code, the camera is triggered at $0 \rightarrow 1$ transition and held until the next $1 \rightarrow 0$ transition. For example, for code 11101000011, three triggers will be sent at 0, 4 and 9 ms and held for a duration of 3, 1 and 2 ms respectively. Note that the number of triggers is not equal to the number of 1's in the code; rather for each continuous set of 1's, one trigger is sent. For indoor datasets, we captured blurred photos of objects placed on a moving variable-speed toy train.

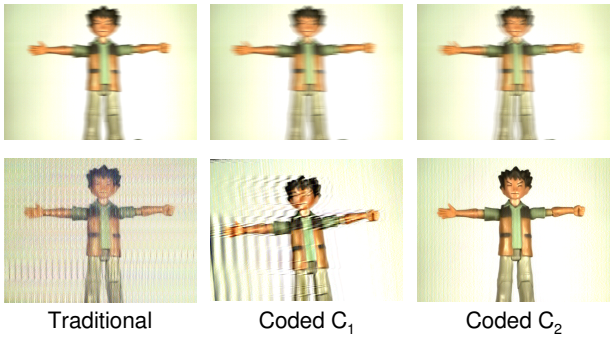

Figure 8. Action figure with complex shape. (Top) Input blurred photos. (Bottom) Deblurring Results. The estimated motion vectors were [-27.99, -1.00], [-49.99,-4.78] and [-41.14,0.22] for traditional, C_1 and C_2 respectively. Note that motion estimation using traditional camera and C_2 code is good. However, only C_2 achieves *both* PSF estimation and PSF invertibility.

4.1. Fast binary code search

We describe two approaches to search for the optimal code for a given code length n that satisfies the criteria for both PSF estimation and invertibility. These criteria are (a) minimize f_{noise}, (b) minimize t, (c) maximize s, and (d) maximize r. Depending on the application, other approaches could be used. Note that the first and the last bit of the code have to be 1, otherwise the code will reduce to a code of smaller length. Thus, in general the search space is of order 2^{n-2} for code length n. For small n, the search space is small and all possible codes can be tested. For larger n, if the search space is large ($> 10^6$), we randomly sample 10^6 codes from the search space for testing.

In the first approach, we fix $s = s^{th}$ and set a threshold (f^{th}_{noise}) on the maximum deconvolution noise that can be tolerated (e.g., 20 dB). We find all codes for which $f_{noise} \leq f^{th}_{noise}$. The search space is equal to $\binom{n-2}{s-2}$. We sort these codes according to t and pick the first code which has the

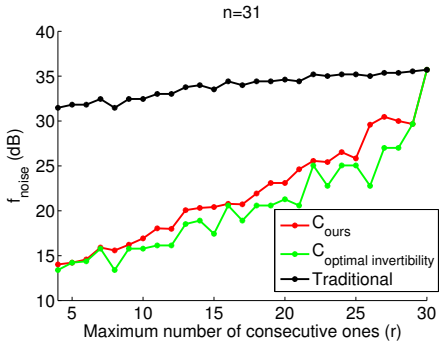

Figure 9. Deblurring performance of our proposed codes and optimal invertible codes [17] with respect to r for the same light level. The proposed codes help PSF estimation and are much better than traditional camera in terms of deconvolution noise. The increase in noise with respect to optimal invertible codes is small.

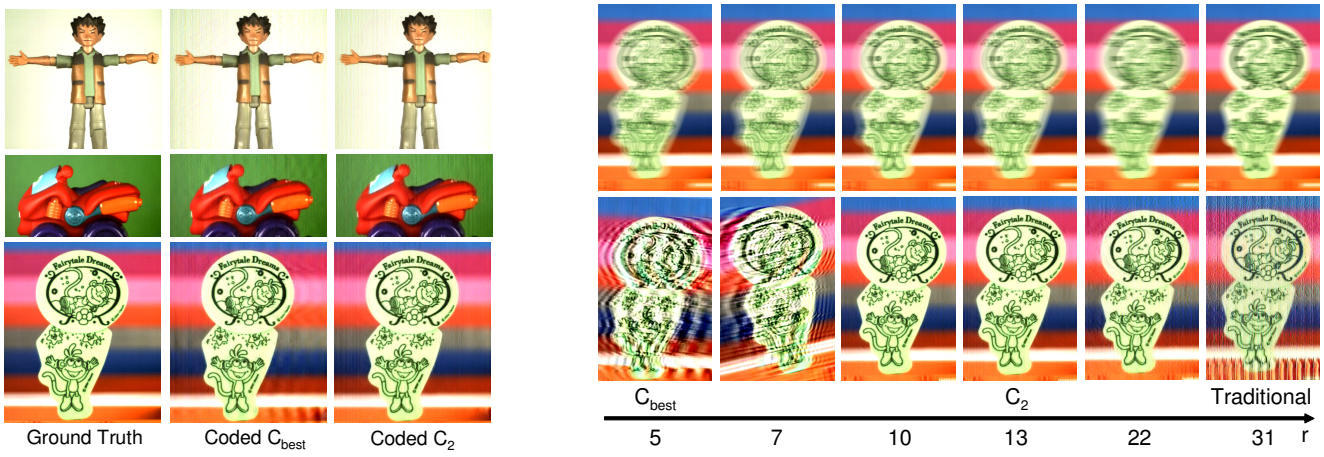

Figure 10. (Left) Visual deblurring comparison of C_2 versus C_{best} on real datasets. Note that the proposed code C_2 gives similar deblurring performance compared to optimal code C_{best}. (Right) PSF estimation capability for $n = 31$ codes with increasing r (decreasing number of $0 \rightarrow 1$ transitions). For small r, PSF estimation fails leading to poor deblurring results. Note that $r = 5$ for C_{best}, and thus optimal invertible code may not give good PSF estimation. As r increases, PSF estimation improves, but PSF invertibility degrades.

maximum r in the sorted list. A second faster approach is to first set r, the continuous number of ones in the code. For simplicity, let the first r bits be ones. The search space is reduced to 2^{n-r-2} (the $(r+1)^{th}$ bit has to be 0 and the last bit has to be 1). Among these codes, we choose those whose $f_{noise} \leq f_{noise}^{th}$ and $s = s^{th}$, and pick the one with minimum t. If no code satisfies the criteria, r is decreased by one and the search is repeated.

The code C_2 described in Section 2.3 is found using the second approach for $n = 31$ and $r = 13$ by testing only $2^{31-13-2} = 65,536$ codes in 6.7 seconds on a standard PC. For this code, $s = 21$ and searching the code using the first approach requires testing $\binom{31-2}{21-2}$=20.03 million codes, which is 305.6 times more than that of the second approach.

4.2. Analysis

We compare the proposed codes with the optimal code for **PSF invertibility** for the *same* light level. The optimal invertible code [17] simply minimizes f_{noise} without considering PSF estimation (r and t). Obviously, the proposed codes will lead to more deconvolution noise, but the increase in deconvolution noise is small and visually the deblurring results are comparable. For example, for $n = 31$, the optimal invertible code $C_{best} = 10111101101010001101111100111111$ was found using [17]. The deconvolution noise for optimal code C_{best} is 18.52dB compared to 20.05dB for C_2.

Figure 9 compares the f_{noise} for the proposed codes and optimal invertible codes for r varying from 1 to n. For a given r, we obtain our code using the approach described in Section 4.1. We record the s value and then find the optimal invertible code which has the same s value (for the same light level). The f_{noise} for a traditional exposure with the same light level is also plotted in black. When $r = n$, there is only one code (all ones) and all three curves meet. In general, the optimal invertible codes have r around $3-5$, so the green and red curves meet at low r values. The plot shows that the increase in f_{noise} using proposed codes is small and the proposed codes are significantly better than the traditional camera in terms of deconvolution noise. Figure 10 (left) shows visual deblurring comparisons on real datasets for C_{best} and C_2 using the same motion PSF (estimated from photos captured using C_2). Note that the deblurring results are visually similar.

PSF estimation: We analyze the PSF estimation capability of the proposed codes for different values of r for a given n. As r increases, the code becomes similar to a traditional camera ($r = n$) and becomes favorable for PSF estimation, but f_{noise} increases significantly. Smaller values of r ($r \approx 5$) result in significant noise in the estimation of alpha values. In Figure 10 (right), we show results using codes having different r values. In general, we found that codes with $r \geq n/3$ work well for PSF estimation.

5. Discussions

We have focused on binary valued codes; however **continuous valued codes** can improve both PSF estimation and invertibility. As shown in [23], continuous valued codes perform better than binary codes in terms of deconvolution noise, since they could avoid the sharp transitions of a binary code and result in smoother blur. In fact, optimizing such codes will be easier using continuous optimization compared to the discrete search used for binary codes. To enforce smooth blur, a penalty on the spatial gradients of the code can be applied, similar to the regularization techniques. However, their implementation is not straightforward using external shutters or trigger-based cameras. It could be achieved by controlling the A/D gain *during* the

exposure time according to the code, but would require changes at the chip level. We have focused on spatially-invariant PSF, but the proposed codes could also be used for **affine motion** using variations of the MFB algorithm. Our approach shares the same limitations of the alpha matting algorithm (e.g., low-frequency background) and requires a few brushes for matting initialization. Combining information from **multiple images** captured with same or different codes will further help in matting and PSF estimation.

Conclusions: PSF estimation is as important as PSF invertibility for motion deblurring. A traditional camera results in smooth blur which is easier to estimate, but makes the PSF non-invertible. A coded exposure camera makes the PSF invertible but results in sharp discontinuities in the blur and degrades PSF estimation. We showed that both criteria of PSF estimation and invertibility can be achieved by carefully designing the code. We proposed design rules based on minimizing the transitions and maximizing the number of continuous ones in the code for good PSF estimation and described two schemes for searching such codes. We analyzed the performance of the proposed codes in comparison with the optimal invertible codes. We also described how coded exposure can be implemented on machine vision sensors without any additional cost and presented real experimental results that showed the effectiveness of the proposed codes for PSF estimation and invertibility.

Acknowledgements We thank Ramesh Raskar for stimulating discussions. We also thank Jay Thornton, Keisuke Kojima, and Haruhisa Okuda, Mitsubishi Electric, Japan, for help and support.

References

[1] R. Accorsia, F. Gasparinib, and R. C. Lanza. Optimal coded aperture patterns for improved snr in nuclear medicine imaging. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 474:273–284, 2001.

[2] S. Dai and Y. Wu. Motion from blur. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 1–8, June 2008.

[3] E. R. Dowski and W. Cathey. Extended depth of field through wavefront coding. *Appl. Optics*, 34(11):1859–1866, Apr. 1995.

[4] E. Fenimore and T. Cannon. Coded aperture imaging with uniformly redundant arrays. *Appl. Optics*, 17:337–347, 1978.

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

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

[7] J. Jia. Single image motion deblurring using transparency. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 1–8, June 2007.

[8] N. Joshi, R. Szeliski, and D. Kriegman. PSF estimation using sharp edge prediction. In *Proc. Conf. Computer Vision and Pattern Recognition*, June 2008.

[9] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM Trans. Graph.*, 26(3):70, 2007.

[10] A. Levin, D. Lischinski, and Y. Weiss. A closed-form solution to natural image matting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(2):228–242, 2008.

[11] A. Levin, P. Sand, T. S. Cho, F. Durand, and W. T. Freeman. Motion-invariant photography. *ACM Trans. Graph.*, 27(3):1–9, 2008.

[12] C.-K. Liang, T.-H. Lin, B.-Y. Wong, C. Liu, and H. H. Chen. Programmable aperture photography: multiplexed light field acquisition. *ACM Trans. Graph.*, 27(3):1–10, 2008.

[13] L. Lucy. An iterative technique for the rectification of observed distributions. *J. Astronomy*, 79:745–754, 1974.

[14] H. Nagahara, S. Kuthirummal, C. Zhou, and S. Nayar. Flexible Depth of Field Photography. In *Proc. European Conf. Computer Vision*, Oct 2008.

[15] S. K. Nayar, V. Branzoi, and T. Boult. Programmable imaging using a digital micromirror array. In *Proc. Conf. Computer Vision and Pattern Recognition*, volume 1, pages 436–443, 2004.

[16] H. Poor. *An Introduction to Signal Detection and Estimation*. Springer-Verlag, 1988.

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

[18] N. Ratner and Y. Y. Schechner. Illumination multiplexing within fundamental limits. In *Proc. Conf. Computer Vision and Pattern Recognition*, June 2007.

[19] W. Richardson. Bayesian-based iterative method of image restoration. *J. Opt. Soc. of America*, 62(1):55–59, 1972.

[20] Y. Y. Schechner, S. K. Nayar, and P. N. Belhumeur. A theory of multiplexed illumination. In *Proc. Int'l Conf. Computer Vision*, volume 2, pages 808–815, 2003.

[21] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. *ACM Trans. Graph.*, 27(3):1–10, 2008.

[22] A. R. Smith and J. F. Blinn. Blue screen matting. In *SIGGRAPH*, pages 259–268, 1996.

[23] A. Veeraraghavan, R. Raskar, A. Agrawal, A. Mohan, and J. Tumblin. Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing. *ACM Trans. Graph.*, 26(3):69, 2007.

[24] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum. Image deblurring with blurred/noisy image pairs. *ACM Trans. Graph.*, 26(3):1, 2007.

[25] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum. Progressive inter-scale and intra-scale non-blind image deconvolution. In *SIGGRAPH '08: ACM SIGGRAPH 2008 papers*, pages 1–10, New York, NY, USA, 2008. ACM.

[26] A. Zomet and S. Nayar. Lensless imaging with a controllable aperture. In *Proc. Conf. Computer Vision and Pattern Recognition*, pages 339–346, 2006.