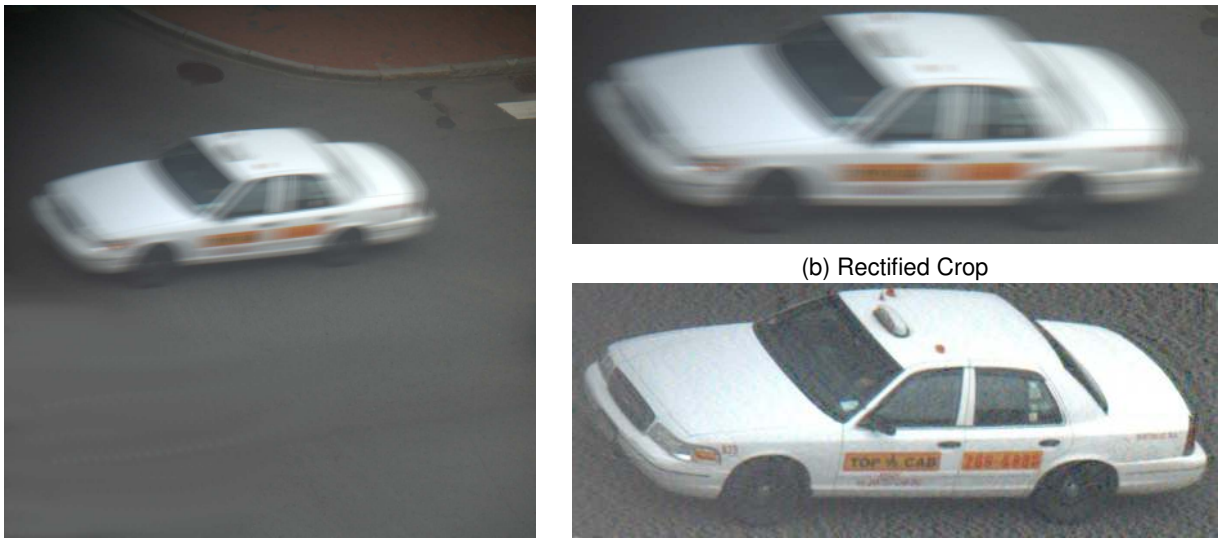
Coded Exposure Photography: Motion Deblurring using Fluttered Shutter

Ramesh Raskar* Amit Agrawal Jack Tumblin
Mitsubishi Electric Research Labs (MERL), Cambridge, MA Northwestern University



(a) Blurred Image (b) Rectified Crop (c) Deblurred Image

Figure 1: Coded exposure enables recovery of fine details in the deblurred image. (a) Photo of a fast moving vehicle. (b) User clicks on four points to rectify the motion lines and specifies a rough crop. (c) Deblurred result. Note that all sharp features on the vehicle (such as text) have been recovered.

Abstract

In a conventional single-exposure photograph, moving objects or moving cameras cause motion blur. The exposure time defines a temporal box filter that smears the moving object across the image by convolution. This box filter destroys important high-frequency spatial details so that deblurring via deconvolution becomes an ill-posed problem.

Rather than leaving the shutter open for the entire exposure duration, we "flutter" the camera's shutter open and closed during the chosen exposure time with a binary pseudo-random sequence. The flutter changes the box filter to a broad-band filter that preserves high-frequency spatial details in the blurred image and the corresponding deconvolution becomes a well-posed problem. We demonstrate that manually-specified point spread functions are sufficient for several challenging cases of motion-blur removal including extremely large motions, textured backgrounds and partial occluders.

1. Introduction

Despite its usefulness to human viewers, motion is often the bane of photography: the clearest, most detailed digital photo requires a perfectly stationary camera and a motionless scene. Relative motion causes motion blur in the photo. Current practice presumes a 0^{th} order model of motion; it seeks the longest possible exposure time for which moving objects will still appear motionless. Our goal is to address a first-order motion model: movements with constant speed rather than constant position. Ideally, the camera would enable us to obtain a sharp, detailed record of each moving component of an image, plus its movement.

This paper takes first steps towards this goal by recoverably encoding large, first-order motion in a single photograph. We rapidly open and close the shutter using a pseudo-random binary sequence during the exposure time so that the motion blur itself retains decodable details of the moving object. This greatly simplifies the corresponding image deblurring process. Our method is not fully automatic: users must specify the motion by roughly outlining this modified blurred region. We then use deconvolution to compute sharp images of both the moving and stationary components within it, even those with occlusions and linear mixing with the background.

Deconvolution to remove conventional motion blur is an old, well-explored idea, but results are often disappointing. Motion blurred images can be restored up to lost spatial frequencies by image deconvolution [Jansson 1997], provided that the motion is shift-invariant, at least locally, and that the blur function (point spread function, or PSF) that caused the blur is known. However, image deconvolution belongs to the class of ill-posed inverse problems for which the uniqueness of the solution cannot be established, and the solutions are oversensitive to any input data perturbations [Hadamard 1923] [Tikhonov and Arsenin 1977]. In comparison, the proposed modification of the capture process makes the deblurring problem well-posed.

*e-mails: [raskar,agrawal]@merl.com, jet@cs.northwestern.edu.
Web: http://www.merl.com/people/raskar/deblur

1.1. Contributions

The motion deblurring problem involves three parts: capture-time decision for motion encoding, PSF estimation and image deconvolution (Figure 2). The key novelty of our methods stem from modifying the capture-time temporal integration to minimize the loss of high spatial frequencies of blurred objects.

- We compute a near-optimal binary coded sequence for modulating the exposure and analyze the invertibility of the process,
- We present techniques to decode images of partially-occluded background and foreground objects,
- We show how to handle extremely large degrees of blur.

We do not perform PSF estimation, but instead rely on easy user interactions to estimate the blur path. Image deconvolution is ill-posed for traditional blur, but coded exposure makes the problem well-posed for small as well as large degrees of blur. Thus, even a simple least squares approach can be used for image deconvolution.

Limitations Our technique is limited to scenes that meet the constant radiance assumption. While points in the scene may move or occlude each other, their intensities must remain constant throughout the exposure time: spurious highlights as in Figure 14 produce artifacts. In addition, coded exposure necessarily reduces the open-shutter exposure time by about half (one f-stop), even without photographed motion. However, the resulting photo is visually similar to an un-coded photo taken with a traditional camera.

Our current implementation uses an external liquid-crystal shutter, but several machine vision cameras support external binary trigger input for on-off control of light integration on the camera sensor chip. Although our techniques cover only a subset of possible object and camera motions, we demonstrate that several cases exist for which our techniques are very effective. These techniques look beyond the traditional camera-shake examples and primarily address object motion within the scene.

1.2. Related Work

Capture-time Solutions Short exposure time reduces blur, but increases noise and needlessly penalize static areas of the image. High speed cameras can capture fast motion, but require expensive sensing, bandwidth and storage. A high speed camera also fails to exploit the inter-frame coherence, while our technique takes advantage of a simplified model of motion. These cameras often require brilliant scene lighting. Edgerton and others [Edgerton 1951-1963] have shown visually stunning results for high speed objects using a modest exposure time but an extremely narrow-duration flash. Flash, however, is impractical in outdoor or distant scenes. In addition, it captures an instant of the action and fails to indicate the general movement in the scene.

Smarter Cameras To overcome camera shake, some newer cameras optically stabilize their lenses. Adaptive optical elements controlled by inertial sensors compensate for camera motion to reduce blur [Canon 2006][Nikon 2005]. Alternatively, some CMOS cameras perform multiple high-speed frame captures within normal exposure time, enabling multiple image-based motion blur removal [Liu and Gamal 2001]. These methods are able to produce clear and crisp images, given a reasonable exposure time. Ben-Ezra and Nayar [2004] proposed a hybrid imaging system that accurately estimates the PSF using an auxiliary low resolution high frame rate camera, enabling deblurring even after long exposure times. These methods compensate for camera motion but do not usually respond to object motion within the scene.

Figure 2: Motion deblurring involves three steps. Our key contribution changes the first stage: temporal shutter modulation reduces loss of high spatial frequencies, permitting use of simple well-posed linear solutions for the last stage. We rely on user-assisted methods to estimate the point-spread function.

Video Analysis Several authors have proposed to remove motion blur using video cameras, where PSF is estimated by combining partial information from successive video frames of a single camera [Schultz and Stevenson 1996][Bascle et al. 1996] or from frames captured by multiple co-located cameras with overlapped exposure time [Shechtman et al. 2002]. This staggered exposure approach also assisted a novel reconfigurable multi-camera array [Wilburn et al. 2005].

Post-processing Solutions Image restoration methods such as blind deconvolution or Wiener filtering [Yitzhaky et al. 1998] attempt to estimate the PSF from the blurred image alone and use the PSF to deconvolve the image. More sophisticated algorithms such as the iterative restoration approach presented by Tull and Katsaggelos [1996] can further improve the quality of restored images.

Without estimating PSF, Jia et al. [2004] improved a short exposure noisy image by using color constraints observed in a long exposure photo. Blind deconvolution is widely adopted to enhance a single blurred image, based on various assumptions applied to the PSF [Kundur and Hatzinakos 1998]. PSF estimation remains a challenging problem for arbitrary motions. Even when the PSF is known, deblurred results have amplified noise and resampling/quantization problems. High frequencies are usually lost in the deblurred results and deblurring is often limited to blur from small movements. Our techniques can gracefully handle a very large degree of motion blur using frequency preserving coded temporal sampling.

Coded Sampling Binary and continuous codes are commonly used in signal processing for modulation and they act as broadband, limited-power substitutes for an ideal impulse signal. These include chirps that sweep the carrier over a wide frequency band during the pulse interval. Maximum length sequences (m-sequences) and Modified Uniformly Redundant Arrays (MURA) are popular choices for coding and decoding by circular convolution. Coded-aperture astronomical imaging uses MURA codes [Gottesman and Fenimore 1989] to improve the signal to noise ratio while capturing X-ray and gamma-ray wavelengths unsuitable for conventional lenses. Broadband codes find wide application in spread spectrum coding for noise-robust communication and in code division multiplexing (CDMA) that minimizes interference between adjacent broadcast channels. Acousticians have used m-sequences to design two-dimensional panels that exhibit minimal sound diffraction [Trevor J. Cox 2003]. However, as shown in later sections, motion blur corresponds to zero-padded circular convolution, and the zero padding makes MURA codes non-optimal.

2. Image Deconvolution

Consider the problem of deblurring a 1-D signal via deconvolution. The goal is to estimate the signal $S(x)$, that was blurred by a linear system's point-spread function $P(x)$. The measured image signal $I(x)$ is then known to be

$$I(x) = P(x) * S(x), \tag{1}$$

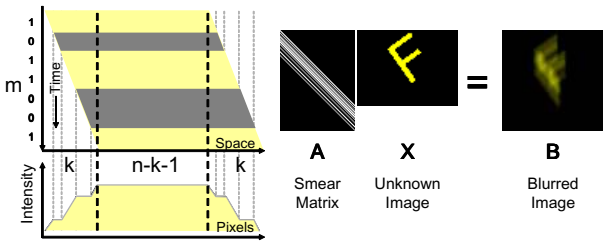


Figure 3: The 1-D motion blur process. (Left) A time-to-space projection for a moving object of size n, with blur of k pixels, chopped with a binary code of length m and the intensity profile of the $n+k-1$ blurred pixels. (Right) A linear system to transform the unknown image into a blurred photo.

with $*$ denoting convolution. In the ideal case, a good estimate of the image, $S'(x)$, can be recovered via a deconvolution filter $P^+(x)$.

$$S'(x) = P^+(x) * I(x). \tag{2}$$

In the case of band-limited point-spread functions or point-spread functions with incomplete coverage of the Fourier domain, information is lost and therefore, deconvolution is not possible. For example, capturing an image with exposure duration T is equivalent to a convolution with a box filter in the temporal domain. We therefore call the resultant alteration a *flat blur*. In the frequency domain, the signal is multiplied by a band-limited sinc function with zeros at the intervals of $2\pi/T$ and significant attenuation at most other frequencies (Figure 5). To overcome this problem, several previous algorithms choose their reconstruction from the range of possible solutions using an iterative maximum-likelihood estimation approach. A well-known class of techniques follow the Richardson-Lucy algorithm [Richardson 1972][Lucy 1974] which uses a statistical model for image formation and is based on the Bayes formula. The Richardson-Lucy algorithm is a non-linear ratio-based method that produces non-negative gray level values.

The iterative deconvolution technique is applicable for whole-motion blur and assumes the complete signal $I(x)$ is available, but it fails to handle cases where different parts of the scene have different PSFs. For example, in the case of a moving object on a static textured background, the background contribution to the blurred image is different from the foreground object smear.

3. Coded Exposure

Rather than leaving the shutter open for the duration of the exposure, we 'flutter' it open and closed in a rapid irregular binary sequence. We call the resultant alteration a *coded blur*. The fluttering toggles the integration of motion on and off in such a way that the resultant point spread function, $P(x)$, has maximum coverage in the Fourier domain. Although the object motion is unknown *a priori*, the temporal pattern can be chosen so that the convolved (blurred) image $I(x)$ preserves the higher spatial frequencies of moving objects and allows us to recover them by a relatively simple decoding process.

3.1. Motion Model

For greater generality, we describe convolution using linear algebra. Let B denote the blurred input image pixel values. Each pixel of B is a linear combination of the intensities in the desired unblurred image, X and can be written as

$$AX = B + \eta. \tag{3}$$

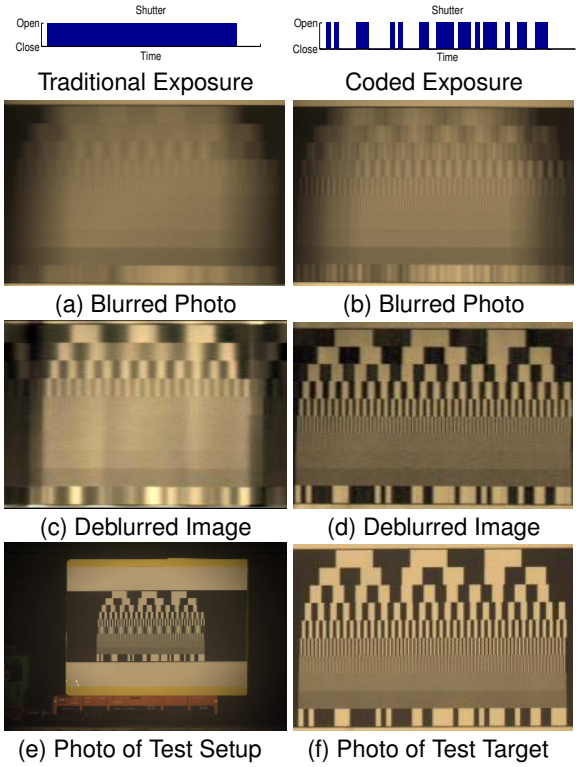


Figure 4: Key idea behind coded exposure is to temporally sample the motion with minimum loss of spatial frequencies. (a,b) Photos of a moving target pattern with varying spatial frequencies show that motion blur in conventional exposure loses high spatial frequencies, but coded exposure preserves those frequencies with attenuated magnitudes. (c,d) The deblurred results show recovered spatial frequencies. (e,f) Images of the static scene.

The matrix A, denoted as the *smearing matrix*, describes the convolution of the input image with the point spread function $P(x)$ and η represents the measurement uncertainty due to noise, quantization error, and model inaccuracies. For two-dimensional PSF's, A is block-circulant while for one-dimensional PSF, A is circulant. For simplicity, we will describe the coding and decoding process for a one-dimensional PSF case.

Given a finite exposure time of T seconds, we subdivide the integration time into m time slices, called *chops*, so that each chop is T/m seconds long. The **on/off chop pattern** then is a binary sequence of length m. The motion blur process is a time to space projection (Figure 3), where, in the one-dimensional motion case, the motion in T seconds causes a linear blur of k pixels. Hence, within one single chop's duration, the smear covers k/m pixels.

Consider a simple case of an object moving downwards in front of a black background and evaluated along a vertical scan line (Figure 3). If the PSF is length k in image pixel coordinates, a pixel at a location (u,v) in the first chop is smeared linearly up to the pixel $(u,v+k-1)$. If the object length along the direction of motion is n pixels, then the total blur width is w where $w = (n+k-1)$.

Our goal is to find the best estimate of the n pixels from the observed $n+k-1$ pixels. The smear matrix A can be obtained as follows. Each pixel in the unknown image X contributes to a total of k pixels after smearing. The first column of circulant matrix A is the PSF vector of length k followed by $n-1$ zeros. And each column is obtained from the previous one by cyclically shifting the entries one step forward. Therefore, in case of a black background,

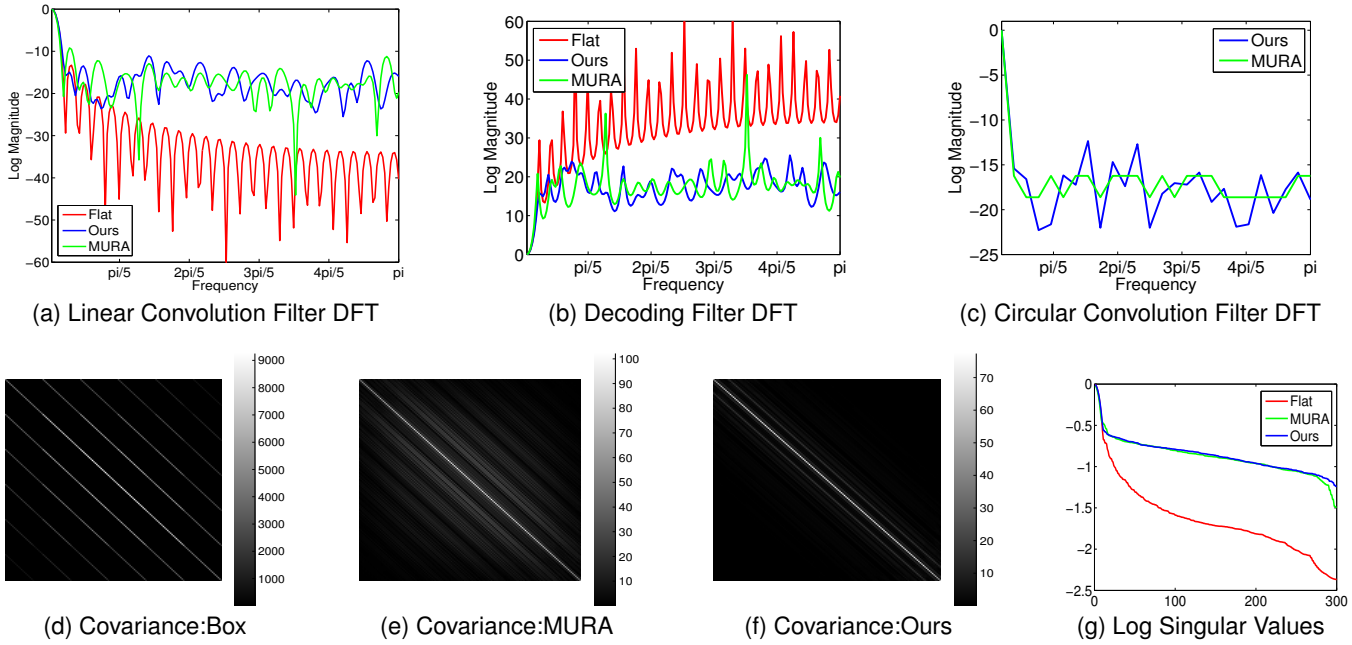


Figure 5: Frequency, covariance and matrix conditionality analysis of codes used for linear convolution. Each 52-length code is padded with 300 zeros. (a) The DFT of the box filter (flat blur) is a sinc function and frequencies with sharp dips are lost. The MURA code filter is better than box filter but the variance of the frequency magnitudes is high compared to our code. Our coded filter is more efficient with a nearly constant broadband response. (b) The DFT of decoding filters show that box and MURA filters require significant amplification of certain high frequencies. (c) For comparison, we show that for circular convolution (i.e. without zero padding), MURA code actually has a flatter frequency response than our code. (d,e,f) Noise covariance matrix of X corresponding to smear matrix A has large diagonal and sub-diagonal entries for box and MURA code. The sub-diagonal entries in covariance matrix indicate ringing and banding artifacts in the deblurred results. (g) The condition number can be analyzed using singular values of the smearing matrix.

the linear convolution with $P(x)$ (or multiplication by the circulant matrix A) is equivalent to a circular convolution with a PSF vector of length k padded with $n-1$ zeros. In practice, since X has only n unknown values in the smear direction, one can build up an over-constrained least square system by truncating A to keep only its first n columns. Thus, the size of A becomes $(n+k-1)\times n$. In the case of flat blur, the time-to-space projection of an input signal of length n with constant values creates a response with a trapezoidal intensity profile. The ramps have a span of k pixels each and the plateau is $n-k-1$ pixels long. For coded blur, the overall intensity profile shape is still trapezoidal, but the shutter's rapid flutter changes the ramps to a more jagged shape as shown in Figure 3.

3.2. Code Selection

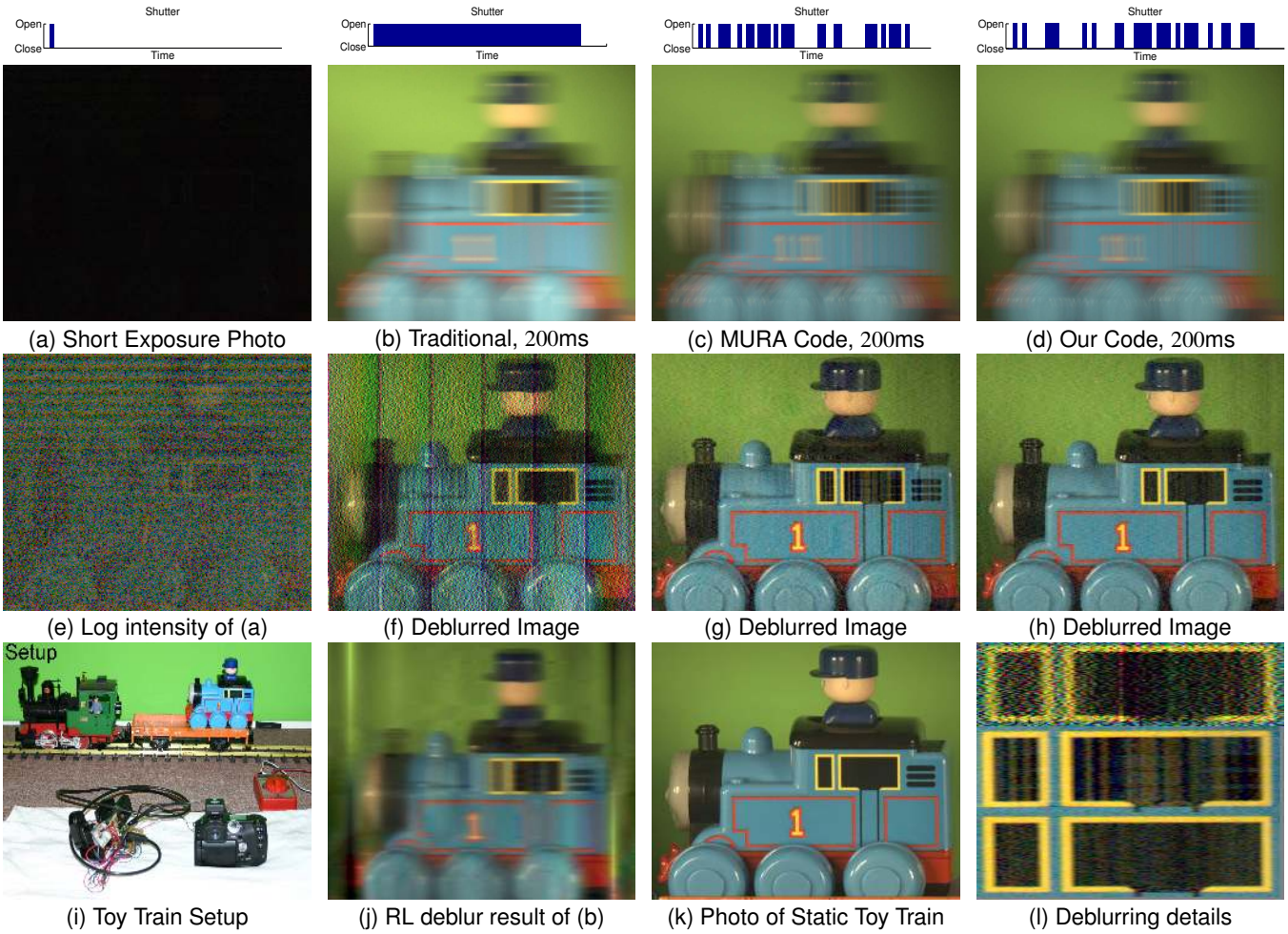
Our goal is to select a temporal code that improves the invertibility of the imaging process. We analyze the invertibility by studying the condition number of the coding matrix and the variance of the frequency spectrum of the code. The invertibility of the smearing matrix A, in the presence of uncertainty and noise, can be judged by the standard matrix conditioning analysis. The condition number is the ratio of the largest to the smallest singular value and indicates the sensitivity of the solution X to the noise in the input image B. We note that the eigenvalues of a circulant matrix comprise the magnitude of the DFT of the first column of the circulant matrix and that each column in A is the PSF vector padded with zeros. Based on this observation, we choose a coded sequence with a broadband frequency response so that the corresponding **condition number** for the smearing matrix is as small as possible.

In theory, we could modulate the opacity of the filter continuously over time to achieve a broadband frequency response, e.g. using a chirp like function. However, in practice a binary (on/off) opacity switching with fixed chop duration is much easier to implement. Choices for **broadband binary codes** include Walsh-Hadamard codes, maximum length sequences and MURA codes. The MURA sequence may seem the obvious choice as its discrete Fourier transform is flat. However, for motion blurring, circular convolution occurs with the PSF vector of length k padded with $n-1$ zeros, where n is the size of the object in pixels. As discussed below, a MURA is not optimal for zero padded patterns, and prompted our search for the best possible code.

The DFT of a MURA pattern without zero padding is flat. However, the DFT can resolve with exactness only the discrete frequencies. There is spectral leakage for components falling between the DFT bins. Zero-padding results in greater resolution in the frequency components and shows the weakness of MURA patterns.

Because the decoding involves inversion of the frequency spectrum, we also add a smoothness constraint to our search for the best binary chop pattern. The frequency response should have low **variance** so that a mis-estimation of the PSF will not cause strong changes in amplification for nearby but incorrect spatial frequencies during decoding. Note that the frequency response of both the box filter sequence and the padded MURA sequence (Figure 5) includes deep dips or zeros, producing a high variance for both. These spikes in the frequency domain leads to the spurious amplification of frequencies during deconvolution.

Finally, one must decide the sequence length m. As described later, an ideal chop-count is the one equal to the blur size k. Ideally, the camera would feature an auto-flutter mode to decide m on-the-fly based on sensed optical flow, a form of motion-adaptation similar to the auto-focus feature. Given our hardware constraints, we



(a) Short Exposure Photo (b) Traditional, 200ms (c) MURA Code, 200ms (d) Our Code, 200ms

(e) Log intensity of (a) (f) Deblurred Image (g) Deblurred Image (h) Deblurred Image

(i) Toy Train Setup (j) RL deblur result of (b) (k) Photo of Static Toy Train (l) Deblurring details

Figure 6: Comparison with other exposure settings: short exposure, traditional shutter exposure, MURA code and our code. The blur k in each case is between 118 and 121 pixels ($\approx 16\%$ of n). (a,b,c,d) Shutter sequence and corresponding photos used as input images. (e) Log intensity for short exposure. (f,g,h) Deblurred results using a linear solution. (i) Experimental setup with toy train. (j) Flat blurred image deblurred using Richardson-Lucy (RL) algorithm. (k) Photo of static toy train. (l) Enlarged regions taken from deblurred results for flat (top), MURA (middle) and coded exposure (bottom). Datasets and source code available at `http://www.merl.com/people/raskar/deblur/`.

settled for a compromise value by experimentation, choosing a sequence of $m = 52$ chops with 50% duty cycle, i.e., with 26 ones and zeros. The first and last bit of the code should be 1, which results in ${}^{50}C_{24} \approx 1.2 \times 10^{14}$ choices. Among them, there are a multitude of potential candidates with acceptable frequency magnitude profile but different phase. We computed a **near-optimal code** by implementing a randomized linear search and considered approximately 3×10^6 candidate codes. We chose a code that (i) maximizes the minimum of the magnitude of the DFT values and (ii) minimizes the variance of the DFT values. The near-optimal code we found is

1010000111000001010000110011110111010111001001100111.

The plot in Figure 5 demonstrates that the chosen code is a significant improvement over padded MURA code. The deblurred images in Figure 6 shows banding and artifacts for flat blur and MURA coded blur as compared to coded blur using our code.

4. Motion Decoding

Given the estimated PSF, we can deblur the captured high resolution image using existing image deconvolution algorithms. However, in several cases described below, we discovered that adding more constraints is difficult via deconvolution, and instead a linear algebra approach is more practical.

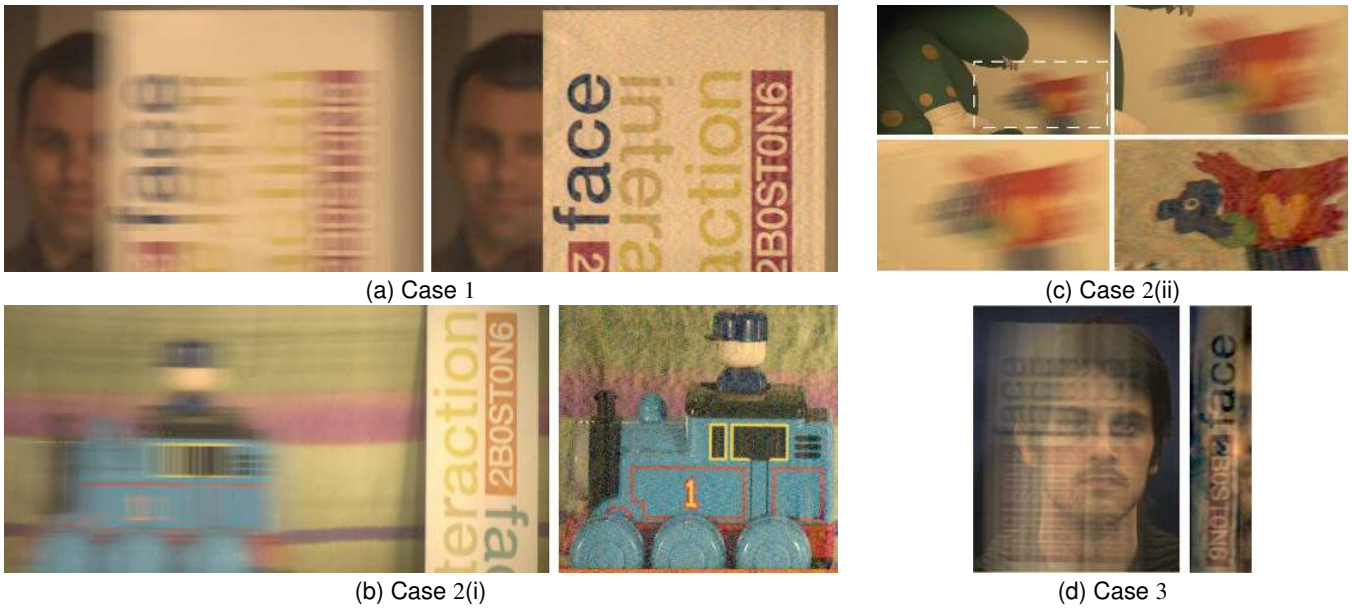
4.1. Linear Solution

We use a least-square estimation to solve for the deblurred image $\widehat{X}$ as

$$\widehat{X} = A^{+}B, \tag{4}$$

where A^{+} is the pseudo-inverse of A in the least-square sense. Since the input image can have a motion blur k different from m, we first expand/shrink the given blurred image by factor m/k. We then estimate X and scale it back by k/m. All the images in this paper have been deblurred using this simple linear approach with no additional post-processing.

In the following sections, we focus on one dimensional PSFs. Motion of real-world objects within a frame tends to be one dimensional due to energy and inertial constraints. We refer to the one dimensional line-like paths for motion as *motion lines*. Note that scene features on a given motion line contribute only to pixels on that motion line and therefore the motion lines are independent. The solution for each motion line can be computed independent of other motion lines. In the explanation below, without loss of generality,



(a) Case 1 (c) Case 2(ii)

(b) Case 2(i) (d) Case 3

Figure 7: Different types of backgrounds. (a) Case 1: The estimated background shows continuity with the unblurred part of the face. (b) Case 2(i): With a striped background aligned with motion lines, we can recover the moving train assuming a single unknown value for the background per motion line. (c) Case 2(ii): The user specifies an approximate crop which has strong gradients in the background. By keeping only the low gradients, this cropped part is left with only the blurred pixels, which is decoded. (d) Case 3: The narrow piece of paper (with Boston06 logo) smears across the face. The unblurred foreground estimate, shown on the right, has distortions due to ambiguities in the solution.

the motion lines are assumed to be oriented along the horizontal scan lines. However, in examples such as camera shake, the PSF is typically a collection of 1-D manifolds in 2-D and we show how our method can extend to these PSF's as well.

In general, deblurring algorithms need to know which pixels belong to the blurred portion of the image and which belong to the unblurred portion. A misclassification can lead to decoding errors where the unblurred background pixels will contaminate the results along the entire motion line. The regions can be classified into blurred and unblurred areas using several automatic methods. In the case of static video cameras, a cutout for the moving object can be obtained via frame differencing or by comparing the position of the moving object in two successive frames. In an interactive deblurring process, the user specifies an approximate axis-aligned bounding box around the blurred object. The length of the box, w_b, would typically be slightly greater than the blur width $w\,(=n+k-1)$. The user specifies the length of blur, k, which indicates to the program that the shape of the object is $n' = w_b - k + 1$. The value n' can be considered a close approximation of n. If the length differs along each motion line, the user specifies a cut-out rather than a bounding box.

4.2. Background Estimation

We now address the problem of motion blur due to an opaque object moving in front of a stationary (non-blurred) but non-zero-valued background. This is a commonplace but difficult case because the moving object blends with the background and it is, therefore, not sufficient to know the moving object's PSF to deblur the image. One also needs to estimate the background simultaneously. We explore this problem, classify the cases and show that in some instances, the unknown background visible at the edges of the blurred object can be recovered during the deblurring process. In the case of a non-zero background, the blurred image is given by

$$B = AX + A_g X_g, \tag{5}$$

where X is the moving foreground object, X_g is the static background and A_g is the background attenuation matrix. A_g is a diagonal matrix whose elements attenuate the static background. A_g can be written as

$$A_g = I - \mathtt{diag}(A * I_{(n+k-1)\times 1}), \tag{6}$$

where $I_{q\times 1}$ is a vector of length q with all 1's and $\mathtt{diag}(\mathbf{v})$ returns a square matrix by placing the vector $\mathbf{v}$ on the main diagonal.

The analysis of background estimation is based on the number of background pixels, g, that contribute to the blurred region. In the blurred region of size $(n+k-1)$, when $n > k$, the background is visible only near the edges and contributes to only $2k$ pixels. However, when $n < k$, the object smears more than its length and hence the background is partly visible in all the blurred pixels. Hence $g = \min(2k, n+k-1)$. Given observations at $(n+k-1)$ pixels, we must estimate a minimum of $n+2k$ values. The additional $k+1$ unknowns can be estimated by adding constraints on the object motion and on the complexity of the texture corresponding to the background image.

We consider the five critical cases for simultaneously estimating foreground X and background X_g.

1. Wide object with known shape and textured background: $n > k$.
2. Wide object with unknown shape: $n > k$ and (i) constant background; (ii) textured background but with edges of the texture outside the blur region; (iii) textured background with edges of the texture passing through the blur region.
3. Narrow object blurred over a distance greater than its length, i.e. $n < k$.

Figure 7 shows examples on the above background cases. In case 1, we constrain a specified region of length $n' = n$ so that the specified cutout precisely marks the shape of the blurred object. As we know the precise location of the ramps of the trapezoidal contribution of the background pixels, we can write the background attenuation matrix A_g. We estimate up to $k-1$ values from among the $2k$ possible values for the background by making a simplifying assumption that the background is low frequency, by assuming each consecutive pair of background pixels (on a motion line) to have the same color. Figure 7 shows recovery of the partially occluded face. In this case, the shape of the moving rectangular sheet of paper is specified by the user. The geometric and photometric continuity of the recovered low-frequency background (right part of the face) is validated by presenting it together with the original un-attenuated portion of the background (left part of the face).

In case $2(i)$, the shape of the object is unknown. When the background is constant, we can treat the background as part of the moving object that creates the smeared values. Only the task of estimating the single unknown background color remains. As each motion line is independent, we can recover a different background color for each motion line. Figure 7 demonstrates that we can decode a striped cloth that follows motion lines and still reliably recover the foreground image.

In case $2(ii)$ the background is not constant, but we can still recover the foreground if the edges of the background are *outside* the blurred region. We use a gradient domain method to eliminate those variations in the background. The highest possible gradient in the foreground is attenuated by a factor of $m/2$ or more due to blurring. Hence, all the gradients along motion lines with a magnitude greater than a threshold are likely to be background edges and are set to zero. The motion line is reconstructed by integrating the modifying gradients starting from the foreground region outwards. The modified motion lines are then decoded. Figure 7 (top right) demonstrates that we are able to deblur by eliminating those edges in the background which do not pass through the blur. However, in case $2(iii)$, the background edges are mixed with the foreground blur and gets highly attenuated. One can only hallucinate or synthesize the background texture for this case.

Finally, in case 3 ($n < k$), every blurred pixel has a contribution from background. This case does not have a unique solution. Since the background attenuation is non-zero for all the pixels, one can get multiple solutions for X and X_g that will result in the same blurred image. Specifically, given a solution X_1 and X_{g1}, one can find another solution X_2 and modify the background so that

$$AX_1 + A_g X_{g1} = AX_2 + A_g X_{g2}. \quad (7)$$

X_{g2} is then given by $X_{g2} = (A_g)^{-1}(A(X_1 - X_2) + A_g X_{g1})$. Note that when $n > k$, the background is not seen completely and A_g is not invertible, while A_g is invertible for $n < k$. Figure 7 shows an example where a thin piece of paper moves in front of the face, along with one of the possible solutions.

Although we have focused on linear or iterative estimation of the physical values, more imaginative solutions can be obtained for visually pleasing results. Such techniques include texture synthesis, image inpainting, structure propagation, or capture of an unoccluded background or "clean plate" image. Note that all the results presented here are simple least squares solutions without any post-processing, demonstrating the strength of the coded exposure technique.

4.3. Simple Motion Generalization

Applying image warping can permit our technique to decode a much broader set of simple motions that project as affine transforms, in plane rotations around a fixed center, as well as movement in perspective along lines that meet at a vanishing point. While the PSF of the coded blur is initially nonlinear and position-dependent, most of the linear motions can be warped to produce an image with spatially invariant uniform-length displacement vectors aligned with image scanlines (e.g. Figure 8). As motion blur follows this same displacement vector field, the warped image provides uniform-width coded-blur regions that are now suitable for decoding. To produce the final result, we simply apply the inverse warp to return the decoded image to its original geometric form.

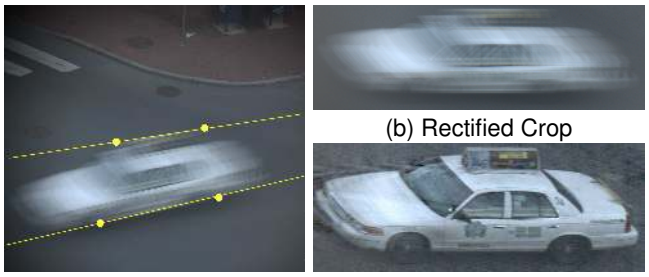


Figure 8: Simple motion generalization. (a) Photo of a fast moving vehicle with a blur of $k = 235$ pixels, which is 24% of the vehicle length in the image. We click on 4 points shown in yellow to specify motion lines. (b) An approximate crop of the rectified motion lines. (c) Despite vignetting on the left, spatial high frequencies are recovered in the deblurred result.

Figure 9: Two dimensional PSF due to camera motion. For a known PSF, coded exposure method also extends to 2D blur from camera shake. (a) We include a point-light source (LED) in the scene to estimate the 2D PSF. (Inset) Enlarged view of the LED blur. (b) Deblurred result retains facial features.

In the case of perspective warp, rectification can be applied after estimating the vanishing point of motion lines. Following rectification, all the warped motion lines are parallel in the camera image space. We used this technique for Figure 1, Figure 8 and several other example images. In-plane rotations (e.g. a rotating fan or swinging pendulum) create motion lines that form concentric circles around the center of rotation. These can be handled by deblurring in polar coordinates.

5. Applications

Camera Motion Camera motion is usually resolved by using a gyro-based stabilization of the optics. Although not a prime focus of this paper, it is also possible to deblur blurred images caused by camera shake. In this case, the PSF is complex and a separate method may be needed to estimate the 2D PSF. This can be achieved via accelerometers or gyroscopes embedded in the cam-

(a) Blurred Image (b) Deblurred Image

Figure 10: Motion deblur in presence of partial occluders. (a) A moving car occluded with a pole. We deblur the image without considering pixels occluded by the pole. (b) Deblurred image.

era or with an auxiliary low resolution high frame rate camera [Ben-Ezra and Nayar 2004]. But because we assume shift invariance (all points share the same PSF) only a single feature needs to be tracked. For our experiment, we placed a point light source (light emitting diode (LED)) in the field of view and used its smear as our PSF. Figure 9 shows the complex PSF captured via the static light source along with the blurred and the deblurred result. Note that all the sharp features of the face are recovered.

Partial Occlusion Removal Motion blur widely distributes the colors of a moving object along its motion lines. If the moving object is partially occluded by a narrow and stationary foreground object, we can still recover colors for all the partially occluded portions of the moving object. Techniques such as image inpainting hallucinate the possible pixel values [Wang and Adelson 1994][Black and Anandan 1996], but encoded blur allows us to encompass a completely observed system.

Returning to the 1-D PSF case, if the blur width size is w and the partial occluder length is d then of the $n+k-1$ pixels recorded, only $n+k-1-d$ pixels are available to reconstruct the n pixels. If blur size k is larger than the occluder length d, then deblurring can be done. Figure 10 shows the recovered car from the blurred image with a pole as the occluder.

6. Implementation and Analysis

Prototype We built a simple device to capture coded-blur photographs (Figure 11) and used it for all the images shown in this paper. We began with an 8 megapixel Canon Pro 1 digital still camera and interfaced it to a PIC microcontroller connected to the camera's hotshoe. The PIC controls a *DisplayTech* ferro-electric shutter placed over the camera lens, and drives the shutter to follow the binary coded sequence, when triggered by the hotshoe signals. The ferro-electric shutter's on/off contrast is high (1000 : 1) and switching time is less than 100 microseconds. Nearly all our pictures were taken with a total exposure time of 200 milliseconds and a coded exposure sequence of 52 chops. We used a LGB 72402 toy electric train (www.lgb.com) with adjustable repeatable speed to acquire controlled datasets. The datasets and source code are available at the project website [RAT 2006].

Noise Analysis Let us compare the noise in the decoded image using coded and traditional exposure techniques. For a linear system $Ax = b$, assuming IID Gaussian noise of mean 0 and variance σ^2 in b, the covariance matrix C_x of noise in x is given by

$$C_x = \sigma^2 (A^T A)^{-1}. \quad (8)$$

Figure 5 shows the absolute of the covariances matrices (assuming $\sigma^2 = 1$). For traditional exposure (flat blur), C_x has large off-diagonal entries, indicating that noise in any one pixel severely affects several other pixels in the deblurred image. The maximum value of C_x (for object size $n = 300$ pixels) is 9270.9, corresponding to noise amplification of 39.67db. For the 52 chop sequence that we have used, the effect of the off-diagonal entries in the corresponding C_x matrix is reduced. The maximum value in C_x corresponding to our coded sequence is 77.6, which gives a noise amplification of only 18.9db.

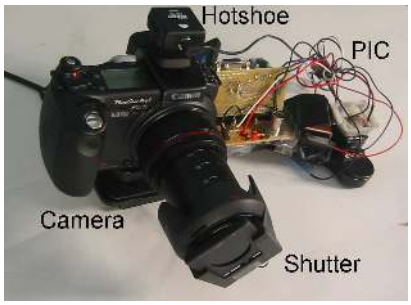


Figure 11: Our prototype coded-exposure camera with ferro-electric liquid crystal shutter.

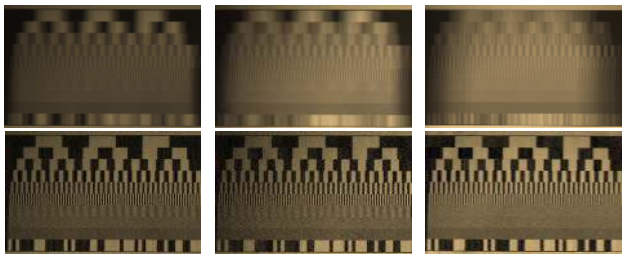


Figure 12: For a given code size, m, the resolution of the deblurred image is related to blur size, k. (Top Row) Blurred images corresponding to $k = 44, 78$ and 193 respectively for $m = 52$. (Bottom Row) Corresponding deblurred images. Note that at $k = 193$, high frequencies (bottom fourth row) are not recovered, while these are recovered when $k = 44$.

Our system can deblur the image to the extent of the motion within a single chop. Let us compare to an image captured with an exposure of a single chop, which is equal to T/m seconds. As the cumulative exposure time for coded exposure is roughly $T/2$, SNR is potentially better by $m/2$ in the blurred region. However, the key advantage with respect to short exposure photo is that in the areas without motion blur (which do not need to be deblurred), our system can record a sharp image with reduced noise.

Resolution Analysis A test target with varying spatial frequencies (Figure 4) was used to measure which frequencies can be reliably recovered. As expected the flat blur case not only fails to recover higher frequencies, but it also erroneously amplifies noise in lower frequencies.

Let us analyze the choice of the code length. A long code (large m) subdivides the exposure time finely and allows decoding of a large amount of blur, but it proves ineffective for small amounts of blur. Conversely, a short code has a longer duration per chop and blur within a single chop cannot be resolved. We would like to keep the ratio k/m close to 1 pixel per chop to achieve best possible sampling of the blur. Figure 12 shows the input images and the corresponding deblurred results for the test pattern with varying spatial frequencies for blur sizes 44, 78 and 193. Note that when the blur is 193 pixels, high frequencies (bottom fourth row) cannot be recovered. Thus, as the blur size k differs from chop count m, the decoding fails to resolve very fine frequencies. However, in practice, we do not need to know the blur size to determine the code length. Using our 52 chop code, we were able to handle motion blur ranging from $k = 27$ to $k = 300$ pixels.

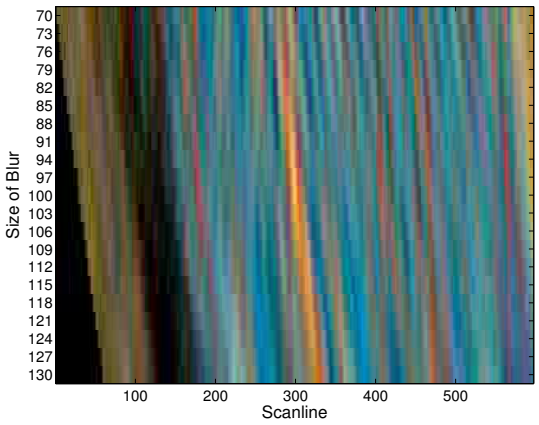


Figure 13: The one-dimensional search for blur size k. The figure shows deblurred values on a scanline for different values of k for the toy train image (Figure 7(b)). Notice that the image is sharp at the correct value of $k = 97$ but at incorrect k values, solutions have spurious high frequencies.

7. Discussion

Deblurring via a fluttered exposure sequence has several advantages while suffering from limitations as described below.

7.1. Benefits

Coded imaging methods such as those used in astronomy or tomography, typically sense a raw image that is meaningless to a human observer. Fortunately, in our case the coded exposure image is useful even if it is not decoded. Compared to flat blur, the image is, in the worst case, half blurred at half the intensity. The coded sequence is easy to implement because it requires toggling of binary opacity as opposed to a more complex continuous opacity control. We hope camera manufacturers can implement the binary switching feature directly on a sensing chip.

7.2. Limitations

Our technique is dependent on the knowledge of the PSF. However, PSF estimation is an active area of research in computer vision and image analysis. An ideal method would compute the degree of blur (defined by blur length k) as well as the motion lines (e.g. vanishing points in perspective warped motion). For the 1-D case, we perform a linear search in the blur width k. Figure 13 shows that the solution for incorrect values of k exhibits high frequency bands, but at the optimal value $(k = 97)$ the image is sharp. Image variance analysis is commonly used to create automatically re-focused photos from light-fields. However, during decoding of the fluttered exposure photos, the incorrect k does not produce a low frequency output. Hence, a new algorithm is required to estimate the correct k automatically and this remains an open area of research.

Large occlusions and dis-occlusions break the spatially-invariant PSF assumption, because a given object patch may only contribute to a partial PSF. Other view dependent effects such as specularities and non-diffuse BRDF can also cause problems. We currently cannot handle transparent or translucent objects that create a coupling or superposition of PSFs at two different scales. Figure 14 shows an example of a transparent teapot on a green background. The recovery fails at strong specular highlights but the least-square estimate does recover outlines of the transparent teapot and highlights which are not saturated in the blurred image. In other cases,

(a) Blurred Image (b) Deblurred Image

Figure 14: Limitations of our technique. The blurred image of a teapot has view dependent properties such as transparency and specularities. Our method fails to recover a sharp image at several locations.

we have noticed that within a reasonably short finite duration some view-dependent objects show minimal distortions of PSF.

For the linear mixing model, the camera response must be linear. Therefore gamma correction, saturation, under-exposure and other non-linearities must be corrected or removed. Spatial distortions such as vignetting and radial distortion may also interfere if left uncorrected. However, these effects appear small: in our examples we did not perform any corrections beyond linearizing the camera response,

7.3. Future Directions

The nature of coded blur photography points to a range of new research problems.

Exploring the Codes We have analyzed the coded exposure in discrete frequency domain and via matrix conditioning analysis. However, the relationship among the various elements: code sequence, code length, the blur length and the corresponding noise after decoding requires further investigation in the continuous domain. The code may have applications in other areas where a linear mixing model is inverted during decoding.

Deconvolution via coded exposure exhibits similarities to code division multiplexing and de-multiplexing of a single communication channel. Advances from the CDMA world in simultaneous orthogonal codes or channel reception with background noise may improve and broaden results in coded blur photography. The coding and reconstruction has several similarities with tomography and coded-aperture imaging, and exploiting this connection may yield further benefits in temporal image processing.

Effects Extension to **video** cameras and motion video could spur future research by exploiting frame-to-frame coherence. Overlapped chops for two or more cameras might permit very fine temporal resolution as well. Similarly, methods to decode blur from overlapping motions of more than one object will allow more complex occlusion effects. Focus and depth also affects blur size, but in two dimensions, suggesting that coded focusing combined with coded exposure might yield a decodable depth map based on maximized local variance in the image.

PSF Estimation Coded exposure may make PSF estimation within a single frame easier as it preserves higher frequencies. The coherence in successive frames can also be used for better background modeling.

Camera Hardware We used an externally toggled opacity to gather images for our results, but coded-exposure **sensor chip implementation** is straightforward. For example, Pointgrey cameras support a 'multiple exposure pulse width mode' where the integration time for all pixels is globally controlled by a binary waveform via external trigger input [Pointgrey Research 2006]. With on-chip

fluttering, the R, G and B pixels might each use a different binary code that exploits the Bayer grid (color sensor interleaving) for finer spatio-temporal resolution. We have explored constant duration codes, but variable width codes may prove beneficial in video to adapt to intra-frame motion.

The codes can be used for **strobed lighting flashes**. Strobed preflashes are already employed in scene analysis and red-eye reduction. Narrow duration flash is typically used to freeze scene motion, but a coded flash sequence will provide greater ability to control motion sensing. In ambient light, the resultant image is a linear combination of two PSF's; flat blur due to ambient lighting and coded blur due to strobed lights. An orthogonal coding of coded flashes and coded exposure in a multi-camera multi-illumination setup may prove useful for recovering moving silhouettes, self-shadowing, and occlusion ordering.

Finally, our work may inspire a manual or automatic **motion knob** on cameras. Similar to auto-focus systems, the camera can have an auto-flutter facility wherein the camera electronics can determine on-the-fly the best code sequence length and duration. An ultrasound sensor or an auxiliary low resolution camera [Ben-Ezra and Nayar 2004] can trigger fluttering by detecting and measuring object motion. As the coded sequencing preserves the look and feel of the conventional flat motion-blurred image, we feel that the camera manufacturers could add fluttered integration without annoying uninterested consumers.

Acknowledgements We would like to thank the anonymous reviewers and several members of the Mitsubishi Electric Research Labs for their suggestions. We also thank Hideaki Nii, Joe Marks, Joseph Katz and Rama Chellappa for helpful discussions and support.

References

BASCLE, B., BLAKE, A., AND ZISSERMAN, A. 1996. Motion deblurring and super-resolution from an image sequence. In *ECCV*, vol. 2, 573–582.

BEN-EZRA, M., AND NAYAR, S. 2004. Motion-based Motion Deblurring. *IEEE Trans. on Pattern Analysis and Machine Intelligence 26*, 6 (Jun), 689–698.

BLACK, M. J., AND ANANDAN, P. 1996. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. In *Computer Vision and Image Understanding (CVIU)*, vol. 63, 75–104.

CANON, 2006. What is optical image stabilizer? http://www.canon.com/bctv/faq/optis.html.

EDGERTON, H., 1951-1963. Rapatronic Photographs. http://simplethinking.com/home/rapatronic_photographs.htm.

GOTTESMAN, S. R., AND FENIMORE, E. E. 1989. New family of binary arrays for coded aperture imaging. *Applied Optics 28*, 20 (Oct), 4344–4352.

HADAMARD, J. 1923. *Lectures on the Cauchy Problem in Linear Partial Differential Equations*. Yale University Press, New Haven, CT.

JANSSON, P. 1997. *Deconvolution of Image and Spectra*, 2nd ed. Academic Press.

JIA, J., SUN, J., TANG, C.-K., AND SHUM, H.-Y. 2004. Bayesian correction of image intensity with spatial consideration. In *ECCV*, vol. 3, 342–354.

KUNDUR, D., AND HATZINAKOS, D. 1998. A novel blind deconvolution scheme for image restoration using recursive filtering. *IEEE Trans. on Signal Processing 46*, 2 (Feb), 375–39.

LIU, X., AND GAMAL, A. 2001. Simultaneous image formation and motion blur restoration via multiple capture. In *Proc. Int. Conf. Acoustics, Speech, Signal Processing*.

LUCY, L. 1974. An iterative technique for the rectification of observed distributions. *Journal of Astronomy 79*, 745–754.

NIKON, 2005. Precise camera-shake compensation at every angle. http://www.nikon.co.jp/main/eng/portfolio/about/technology/nikon_technology/vr_e.

POINTGREY RESEARCH, 2006. PGR IEEE-1394 Digital Camera Register Reference. http://www.ptgrey.com.

RAT, 2006. Coded-exposure datasets. http://www.merl.com/people/raskar/deblur/.

RICHARDSON, W. 1972. Bayesian-based iterative method of image restoration. *J. Opt. Soc. of Am. 62*, 1 (January), 55–59.

SCHULTZ, R. R., AND STEVENSON, R. L. 1996. Extraction of high-resolution frames from video sequences. In *IEEE Trans. on Image Processing*, vol. 5, IEEE, 996–1011.

SHECHTMAN, E., CASPI, Y., AND IRANI, M. 2002. Increasing space-time resolution in video. In *ECCV*, Springer-Verlag, London, UK, 753–768.

TIKHONOV, A. N., AND ARSENIN, V. I. 1977. *Solutions of ill-posed problems [Metody resheniia nekorrektnykh zadach]*. Halsted Press, New York.

TREVOR J. COX, P. D. 2003. Engineering art: the science of concert hall acoustics. *Interdisciplinary Science Reviews 28*, 2, 119–129.

TULL, D. T., AND KATSAGGELOS, A. K. 1996. Iterative restoration of fast-moving objects in dynamic image sequences. *Optical Engineering 35*, 12 (Dec), 3460–3469.

WANG, J., AND ADELSON, E. 1994. Representing moving images with layers. *IEEE Trans. Image Processing 3*, 5 (Sept), 625–638.

WILBURN, B., JOSHI, N., VAISH, V., TALVALA, E.-V., ANTUNEZ, E., BARTH, A., ADAMS, A., HOROWITZ, M., AND LEVOY, M. 2005. High performance imaging using large camera arrays. *ACM Trans. Graph. 24*, 3, 765–776.

YITZHAKY, Y., MOR, I., LANTZMAN, A., AND KOPEIKA, N. 1998. Direct method for restoration of motion-blurred images. *J. Optical Society of America A (Optics, Image Science and Vision) 15*, 6 (June), 1512–1519.

Appendix: Source Code

```
% Motion Deblurring Source Code for a Simple Case Shown in Figure 3
% Solves for 1D motion blur assuming object is moving from top to bottom
% Using the 52 element binary sequence for fluttering

m = 52; % Coded Sequence length
CodeSeq=double('1010000111000001010000110011110111010111001001100111')-'0';
% Read input image
InputImage = double(readpfm_color('InputTaxi.pfm'));
[H,W,CH] = size(InputImage);
k = [235]; % Assume Blur Size in pixels = 235

% Resize image height by m/k so that the effective blur is m
InputImage1 = imresize(InputImage,[ceil(H*m/k) W],'bicubic');

% Get object size, n, knowing blurredSize = n + m - 1
n = size(InputImage1,1) - m + 1;

% Foreground: Get A matrix for foreground which encodes the blurring
Af = ComposeMotionBlurMatrix(CodeSeq, n);

% Background: bk is contribution vector of bkgrnd in blurred img for all pix
bk = abs(1 - Af*ones(n,1));

% Assume constant background in first m and last m pixels (effective blur is m)
bkLeft = zeros(size(bk)); bkLeft(1:m)=bk(1:m);
bkRite = zeros(size(bk)); bkRite(end-m+1:end)=bk(end-m+1:end);
```

```
% Ready to Solve AX=B for each color channel
A = [Af  bkLeft  bkRite];
for colorchannel = 1:CH
    B = InputImage1(:,:,colorchannel); % coded img for channel
    X = A\B; %Least square solution
  OutColorImage(:,:,colorchannel) = X(1:end-2,:);
End
% Expand/shrink the deblurred image to match blur size of k
OutColorImage = imresize(OutColorImage,[H-k+1 W],'bicubic');
```

```
function [A] = ComposeMotionBlurMatrix(CodeSeq,n)
CodeSeq = CodeSeq(:)/sum(CodeSeq);
k = size(CodeSeq,1);
ZeroPaddedCodeSeq = sparse([CodeSeq; zeros(n-1,1)]);
A = gallery('circul',ZeroPaddedCodeSeq)';% Create Circulant Matrix
A = A(:,1:n); % Keep first n columns out of (n+k-1) columns
```