

Velocity-Dependent Shutter Sequences for Motion Deblurring

Scott McCloskey

Honeywell ACS Labs, Golden Valley, MN, USA
scott.mccloskey@honeywell.com

Abstract. We address the problem of high-quality image capture of fast-moving objects in moderate light environments. In such cases, the use of a traditional shutter is known to yield non-invertible motion blur due to the loss of certain spatial frequencies. We extend the flutter shutter method of Raskar et al. to fast-moving objects by first demonstrating that no coded exposure sequence yields an invertible point spread function for all velocities. Based on this, we argue that the shutter sequence must be dependent on object velocity, and propose a method for computing such velocity-dependent sequences. We demonstrate improved image quality from velocity-dependent sequences on fast-moving objects, as compared to sequences found using the existing sampling method.

1 Introduction

In challenging photographic situations, where ambient illumination is low or subject/camera motion is high, blur is a significant image quality problem for both consumer photographic and computer vision applications. Both optical and motion blur have been studied in the literature, and the limits of de-blurring are well understood. With respect to motion blur, it is well-known that the use of a traditional open/closed shutter results in motion blur that is non-invertible. That is, the use of a traditional shutter precludes the recovery of a sharp image information at certain spatial frequencies, and images processed by de-convolution will contain significant reconstruction artifacts. In order to address this shortcoming, recent work in computational photography has advocated the use of non-traditional capture methods to ensure invertibility of blur in the captured images. The use of coded exposure has demonstrated an ability to produce images with good contrast at all spatial frequencies without significant artifacts.

The fundamental idea of the flutter shutter approach is to open and close the shutter several times during image capture in order to produce an image with invertible motion blur. The motion blur is considered to be invertible if the associated modulation transfer function (MTF - the Fourier magnitude of the point spread function [PSF]) is greater than zero for all spatial frequencies. For such blurs, the deconvolution process is well-posed and the sharp image can be recovered from the blurred camera image, as illustrated in Fig. 1.

Though the shutter's fluttering pattern is one determinant, the effective PSF also depends on the motion of the object. As with other current work with motion blur, we assume that the object follows a linear trajectory with constant

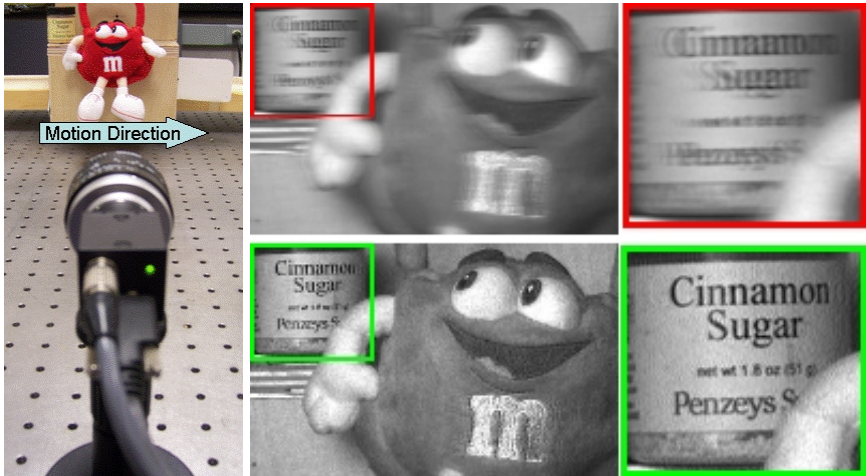


Fig. 1. We employ an off-the-shelf camera to capture coded exposure imagery of high-speed motion using velocity-dependent shutter sequences. The coded exposure image (top right) is then de-blurred to give a sharp image (bottom right) without artifacts.

(unknown) velocity, and that either blur is uniform or that blurred regions have been segmented in advance. Under these assumptions, a given flutter pattern can generate any of a family of PSFs, depending on the object’s velocity and motion direction. Though this dependency has been mentioned in [1], we contribute the first analytic characterization of the relationship, and demonstrate that no fluttering pattern can generate a family consisting entirely of invertible PSFs. We therefore argue that fluttering patterns must be designed and selected for a particular velocity. We provide theoretical motivation for our algorithm to generate velocity-dependent fluttering patterns, which is shown to improve image quality on reconstructions of fast-moving objects. We also consider (previously ignored) read-out noise due to the use of an electronic shutter.

2 Related Work

Numerous methods in the category of blind deconvolution [2] have been presented to mitigate the effects of motion or optical blur in images. Though these methods may be successful relative to certain aesthetic objectives, they are fundamentally limited by the blurred input images. The image of an object moving along a straight line with constant velocity is equivalent to a sharply-focused image convolved with a 1D rectangular point spread function. The magnitude of the Fourier transform of such a PSF (known as the Modulation Transfer Function or MTF) is small at middle and high spatial frequencies, and goes to zero at several frequencies. As a result, contrast of a motion-blurred object will be significantly muted at the middle and high spatial frequencies, and there will be no contrast at a number of *lost frequencies* (the frequencies at which the MTF

vanishes). These spatial frequencies are lost when captured through a traditional shutter, and post-processing the image cannot recover them and may instead introduce artifacts. Such images can, however, be processed by learning methods that use outside information (priors, etc.) to produce visually pleasing images [3,4]. While successful in that regard, hallucinating image content is inappropriate for many applications, e.g. forensics, that require the recovery of the true scene.

Given the incremental improvements of camera sensitivity, researchers have begun to use computational photographic methods to achieve fundamental new gains. Hasinoff and Kutulakos [5,6] propose light-efficient photography as a faster way of capturing images with large depth of field from multiple images. Telleen *et al.* [7] combine multiple, poorly-exposed images from a hand-held camera to produce low-noise images of stable image regions. Ben-Ezra and Nayar [8] use a hybrid camera to simultaneously acquire high-resolution/low frame rate and low-resolution/high frame rate videos; the point spread function estimated from the low resolution video is used to deblur the high resolution video. Synthetic apertures [9] have been shown capable of acquiring both scene radiance and depth in a single image; scene depth can subsequently be used to deblur optically-defocused regions of the scene, increasing depth of field. Levin *et al.* [10] acquire and process uniformly motion-blurred images with an invertible PSF by moving the camera during exposure, assuming a priori knowledge of the motion direction.

We extend the fluttering shutter method of Raskar, *et al.* [11] and do not require a priori knowledge of the motion direction. The flutter shutter approach chooses a shutter timing pattern with the intent of optimally preserving image content at all spatial frequencies, and preserving those frequencies at a nearly uniform level of contrast. Because the effective PSF is zero-padded, the MURA pattern [12] is not necessarily optimal. Raskar’s shutter timing pattern is found by dividing the acquisition time into several *chops* of uniform duration, and by assigning a label of open or closed shutter to each of the chops subject to a constraint on the total exposure. Representing the timing pattern as a binary sequence (with 1s and 0s corresponding to open and closed shutter chops, respectively), the search for an optimal sequence is carried out by random sampling and a fitness function computed on the Fourier magnitude of the binary sequence, *which is assumed to be the effective MTF*. Agrawal and Xu [13] present a method to determine chop sequences that are optimal with respect to both invertibility and ease of blur estimation via alpha matting, but continue to conflate the PSF and the chop sequence. As we demonstrate in the next section, the equivalence of the binary chop sequence and the PSF only holds at a particular velocity and, at all other velocities, the invertibility of the effective PSF cannot be guaranteed.

3 Velocity Dependence

The fundamental notion behind the fluttering shutter concept is that the open/close sequence of the shutter should be chosen to give a PSF that passes all spatial frequencies with nearly uniform levels of contrast. In addition to the

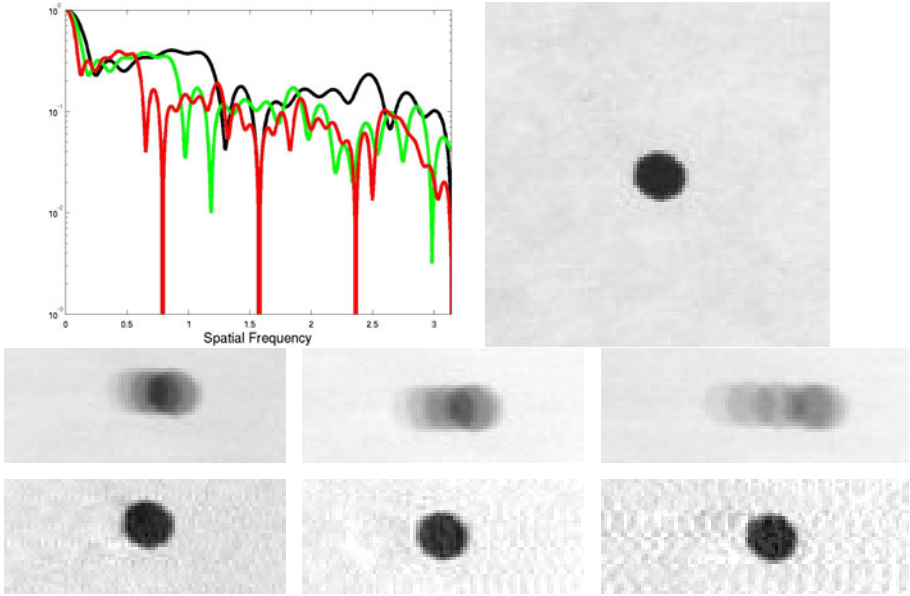


Fig. 2. For a particular fluttering sequence, the effective PSF/MTF depends on subject velocity. (Top Left) The effective MTFs of motion through a particular fluttering shutter for object velocities of 3 pixels per ms (black), 4p/ms (green), and 6p/ms (red). (Top Right) Reference image of stationary target. (Lower Images) Coded exposure images [top] of a dot moving left to right and de-blurred images [bottom]. Though the reconstructed image quality is good for object speeds of 3p/ms (bottom left), there are lost frequencies at 6p/ms (bottom right) and the reconstruction has noticeable artifacts.

binary chop sequence $S(t), t \in \{0, 1, \dots, N-1\}$ ¹, the fluttering pattern is specified by the duration t_{chop} of each chop. As such, the exposure time of the flutter shutter image is $t_{chop} \sum S(t)$ and the total acquisition time (the time from the start of the first open chop to the end of the last) is Nt_{chop} . By convention, the fluttering pattern is chosen to have an acquisition time no greater than twice the exposure time, i.e. no fewer than half of the $S(t)$ are open shutter chops. In order to implement an arbitrary chop sequence on a particular camera, it is necessary for the camera to support open and closed shutter periods as short as t_{chop} , a constraint that we discuss further in Section 4.

Though the fluttering sequence is one determinant, the effective PSF of motion blur also depends on the object's velocity on the image sensor. Though the velocity on the image sensor depends both on the object's real-world velocity and its distance from the camera, it is the image velocity that determines the PSF. Because of the PSF's dependence on velocity, a particular fluttering sequence

¹ By convention, the chop sequence begins and ends with a 1 representing open shutter chops, i.e. $S(0) = S(N-1) = 1$.

defines a family of PSFs, as illustrated in Fig. 2. A notable member of this family, which we refer to as the ‘nominal’ PSF, is effective when the object moves over a range of N pixels during the course of exposure with a fluttering sequence composed of N chops. In this case the effective PSF (call it B_N) is equal to a scaled version of the chop sequence S ,

$$B_N(t) = \frac{S(t)}{\sum S(t)}, \text{ for } t = 0, 1, \dots, N - 1. \tag{1}$$

In this case, which has been considered in [11,13], the Fourier transform \widehat{B}_N of the PSF B_N will be the same as that of the chop sequence S (up to a scale factor). Presuming that S was chosen to preserve all frequencies, this nominal PSF is invertible and the sharp image can be recovered.

In the general case, however, the PSF is a stretched version of the chop sequence and may not be invertible. In fact, no chop sequence can generate a family consisting of invertible PSFs for all velocities. In particular, if the object moves over $2N$ pixels during the course of exposure, the effective PSF will be

$$B_{2N} = \frac{1}{2 \sum S(t)} * [S(0) S(0) S(1) S(1) \dots S(N - 1) S(N - 1)], \tag{2}$$

where $*$ represents an element-wise multiplication of the sequence. As we will now demonstrate, B_{2N} suffers lost frequencies that cannot be recovered post-hoc.

Lemma 1. *Let S be an arbitrary chop sequence of length N . The effective PSF B_{2N} for an object that moves over $2N$ pixels during exposure will have a lost frequency at $k = \frac{N}{2}$.*

Proof. Let $A = \frac{1}{2 \sum S(t)}$.

$$\begin{aligned} \widehat{B}_{2N}(k) &= A \sum_{t=0}^{N-1} S(t) \left(e^{-i\frac{2\pi}{N}k(2t+1)} + e^{-i\frac{2\pi}{N}k2t} \right) \\ &= A \sum_{t=0}^{N-1} S(t) e^{-i\frac{2\pi}{N}kt} \left(e^{-i\frac{2\pi}{N}k(t+1)} + e^{-i\frac{2\pi}{N}kt} \right) \\ \widehat{B}_{2N}\left(\frac{N}{2}\right) &= A \sum_{t=0}^{N-1} S(t) e^{-i\pi t} \left(e^{-i\pi(t+1)} + e^{-i\pi t} \right) \\ &= A \sum_{t=0}^{N-1} S(t) e^{-i\pi t} 0 = 0 \square \end{aligned} \tag{3}$$

It can similarly be shown that PSFs of the general form

$$B_{\kappa N} = \frac{1}{\kappa \sum S(t)} * \left[\underbrace{S(0) \dots S(0)}_{\kappa \text{ times}}, \dots, \underbrace{S(N - 1) \dots S(N - 1)}_{\kappa \text{ times}} \right], \tag{4}$$

will have $\kappa - 1$ lost frequencies. As well, it can be shown that for any object that moves over more than $2N$ pixels, the MTF will have at least one zero in the

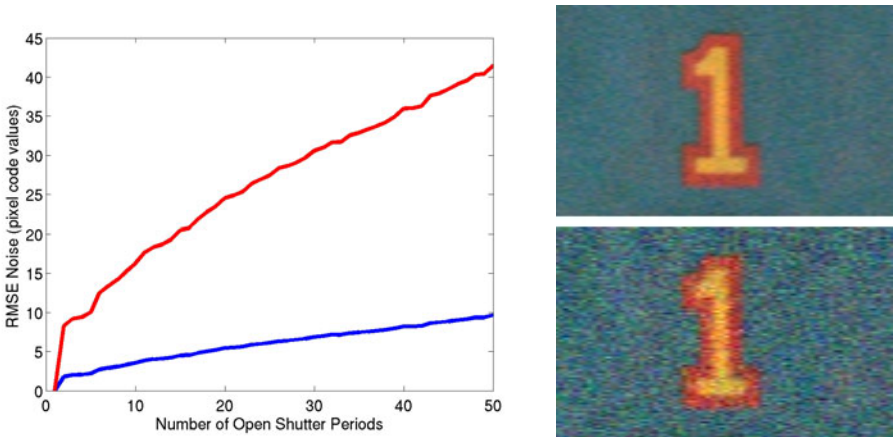


Fig. 3. Use of an electronic shutter for coded exposure imposes read-out noise proportional to the number of open shutter periods. (Left) Plots shows root mean squared error (RMSE) due to read-out noise in the captured image (blue), and the de-blurred image (red). For all captures, the exposure time is fixed at 30ms. (Top right) Reconstructed patch derived from an image captured with a physical shutter (taken from [11]). (Bottom right) Reconstructed image patch derived from an image with simulated read-out noise corresponding to 13 open chop periods, i.e. representing the electronic shutter implementation of the sequence given in [11].

range $0 \leq k \leq \frac{N}{2}$. The implication of this result is that the invertibility of coded exposure blur depends on the velocity; one cannot expect a flutter sequence designed for motion over N pixels to perform well when the velocity produces an effective motion of $2N$ pixels. We demonstrate the image quality implications of this in the Fig. 2 and the experiments of Section 6.

In order to capture images with invertible motion blur, the shutter’s fluttering pattern must be selected according to the object velocity. The use of an inappropriate fluttering pattern may cause artifacts in the processed images, as illustrated in Fig. 2. As such, it is necessary to compute different fluttering patterns for specific velocities. One naive way to ensure invertibility of blur is to shorten the duration t_{chop} of each chop to compensate for higher than expected velocity. There are two problems with this approach, namely that (1) the exposure time would be reduced (and noise increased) and (2) the camera hardware may not support the shortened chop duration. Before describing our method to determine velocity-dependant fluttering patterns for each exposure time, we first discuss hardware limitations on the fluttering sequence.

4 Hardware Considerations

Though the original flutter shutter work [11] was demonstrated with a custom-made camera, more recent work [13] (including our own) has employed off-the-shelf

cameras from Point Gray Research. Several of their cameras support flutter shutter image acquisition through an external trigger and multiple-exposure capture mode. This mode captures a single image whose exposure is accumulated over a pre-set number of pulses of variable duration. Because the camera lacks a physical shutter, the CCD sensor is cleared at the beginning of each pulse and at the end of a pulse the charge is added to the accumulated exposure. This transfer imposes read-out noise at the end of each open shutter period of a flutter shutter capture, a fact that has not been noted elsewhere. The noise level in the coded exposure image is proportional to the number of open shutter periods, as shown by the blue line in Fig. 3. Because de-convolving the flutter shutter PSF from the captured image amplifies noise, images de-blurred from those captured with more open shutter periods will have still more noise, as shown by the red line. The two images in Fig. 3 illustrate the difference in reconstructed image quality between a physical shutter implementation (top) and electronic shutter implementation (bottom) of the sequence given in [11]. In order to avoid such noise-related degradation, we bias our shutter finding method to favour sequences with fewer open shutter periods, as described in the next section.

A second result of the lack of a physical shutter is that there are constraints on both the minimum open shutter pulse length and the minimum time between pulses. These constraints depend on the image format and frame rate; for the Flea[®]2 camera used in our experiments, the 800-by-600 pixel grayscale image mode at 15Hz is the least restrictive, requiring pulse lengths of at least $1\mu s$ and at least $1.21ms$ between pulses. This second constraint is quite restrictive in light of the use of randomly-sampled binary chop sequences. In the event that the chop sequence contains the subsequence 101, this means that t_{chop} cannot fall below $1.21ms$, meaning that (for example) the minimum acquisition time of the 52 tap sequence given in [11] is 62ms and the minimum exposure time is 31ms. In the event that the image velocity of an object is 4 pixels per ms (a velocity that we consider in our experiments), the effective PSF would be 248 pixels long. Because of edge effects in the de-convolution, less than half of our 800-by-600 images could be recovered in this scenario.

5 Shutter Finding Method

Though it is possible to employ rejection sampling to find sequences without a 101 subsequence, this strategy would be extremely inefficient, as the frequency of random binary strings without a 101 subsequence decreases exponentially with the sequence length. For 32 element chop sequences, more than 98% of all sequences have a 101 substring, and for 52 element sequences the proportion is more than 99.9%. Use of rejection sampling, therefore, would add a factor of 50 to 1000 to the time required to find a suitable sequence. Instead of attempting to find a good sequence by sampling random sequences and rejecting those that can't be implemented, our method *constructs* a near optimal sequence that respects given hardware constraints.

We attempt to find an optimal PSF with respect to reconstructed image quality. In previous work [11], the minimum contrast in the MTF and variance

of the MTF are mentioned as optimization criteria. We add a third criteria, mean contrast in the MTF and, when targeting a camera with an electronic shutter, a fourth term (number of open shutter periods) to limit read-out noise. Numerically, the fitness of a given PSF is a weighted sum of these terms,

$$F(B) = w_1 \min_k(|\hat{B}(k)|) + w_2 \text{var}_k(|\hat{B}(k)|) + w_3 \text{mean}_k(|\hat{B}(k)|) + w_4 C, \quad (5)$$

where C represents the number of open shutter pulses.

In order to find reasonable values for these weights, we have simulated all PSFs for $N = 16$ and measured the RMSE of the reconstructed image in the presence of Gaussian noise (including a noise component proportional to C). By computing these errors for 5 random images from the Corel dataset, we find the optimal weights in the least-squares sense, and set $w_1 = 0.1$, $w_2 = -0.2$, $w_3 = 3.4$, and $w_4 = -0.1$.

Because Lemma 1 tells us that no single sequence can be expected to produce a good coded exposure image for all velocities, our method determines a unique flutter sequence for each combination of subject velocity and total required exposure time. Sequences can be completely specified by the open shutter segments, each segment having a duration and start time. The segments are constrained to be non-overlapping, have durations and spacings that respect hardware constraints, and have a total open shutter duration that equals the required exposure time. Our method builds a flutter pattern by first determining the segment durations and then determining each segment’s start time. As we will show, the choice of segment durations (without start times) gives an upper bound to the contrast at all spatial frequencies; it determines the envelope of the MTF.

Lemma 2. *Let B be an arbitrary PSF of length N , let B^1, B^2, \dots, B^C represent N -length PSFs such that $B(t) = \sum_{c=1}^C B^c(t)$, and let $\overleftarrow{B}^1, \overleftarrow{B}^2, \dots, \overleftarrow{B}^C$ represent shifted versions of the B^c such that its first non-zero entry appears at $t = 0$ (see Fig. 4). The sum of the MTFs of the \overleftarrow{B}^c is an upper bound of the MTF of B .*

Proof.

$$\begin{aligned} \|\widehat{B}(k)\| &= \left\| \sum_{t=0}^{N-1} B(t) e^{-i \frac{2\pi}{N} kt} \right\| = \left\| \sum_{t=0}^{N-1} \sum_{c=1}^C B^c(t) e^{-i \frac{2\pi}{N} kt} \right\| \\ &\leq \sum_{c=1}^C \left\| \sum_{t=0}^{N-1} B^c(t) e^{-i \frac{2\pi}{N} kt} \right\| \\ &= \sum_{c=1}^C \|\widehat{B}^c(k)\| = \sum_{c=1}^C \|\widehat{\overleftarrow{B}^c}(k)\| \equiv \|\|\widehat{B}(k)\|\| \square \end{aligned} \quad (6)$$

We denote this final quantity $\|\|\widehat{B}\|\|$ for future reference. This insight allows us to significantly limit the search space of potential flutter sequences, investigating only those composed of chop durations that might result in an invertible PSF.

Our algorithm produces a shutter sequence given the required exposure, subject velocity (given in pixels per ms), and hardware constraints on the shortest

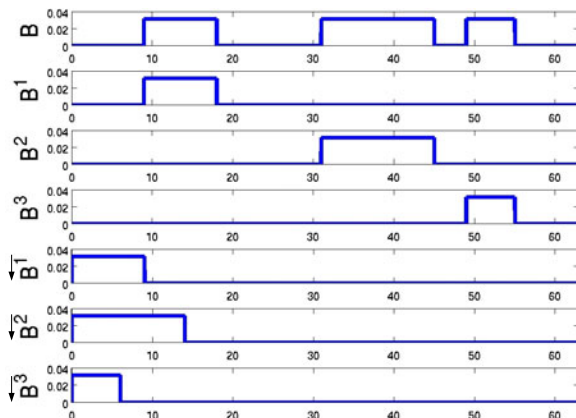


Fig. 4. PSF B (top row), its decomposition into chops (rows 2-4), and the shifted versions of these chops (rows 5-7). Lemma 2 shows that the MTF of B is bounded at each spatial frequency by the sum of the MTFs of the shifted chops.

chop duration c_{chop} and shortest period between open shutter periods c_{gap} . We find an optimal shutter sequence by first ranking each combination of open shutter durations and then by exploring arrangements of these in priority order, as in Algorithm 1. The search is terminated either when the fitness of the current best shutter sequence is greater than that of the envelope of the next combination of open shutter chop durations or when an optional timeout is reached. We describe the steps in greater detail in the remainder of this section.

Our method first determines all partitions of the required exposure time into sets of valid open shutter chop durations. We take t_{chop} to be the larger of either the minimum integration constraint or the value $\frac{1}{v}$, where v is the object's image velocity expressed in pixels per ms. The set of potential open chop lengths is taken to be all integer multiples of this shortest duration, and we compute all partitions of the exposure time into open shutter chops from this set. For each such partition, we compute the MTF envelope $\|\|\widehat{B}\|\|$ and measure the fitness of that envelope. This gives a ranking of each partition based on the *potential* fitness of a PSF corresponding to a shutter timing with that set of chop durations.

In the order of this ranking, we consider each partition and attempt to find the matching set of open shutter chop start times that produces the best PSF. We do this (in the `BestSequenceOfPartition` function) by starting with a random set of start times, such that the open shutter chops do not overlap and the required time between them is maintained. We then pursue a hill climbing strategy of computing several deformations of the current start times, selecting the one with the best fitness, and iterating. We repeat this process for several random sets of start times, and keep the sequence corresponding to the overall best PSF.

The ranked list of partitions is traversed until the fitness of the envelope corresponding to the next best partition is less than the fitness of the current

Input: Exposure time T , velocity V , constraints c_{gap} and c_{chop}
Output: Shutter sequence S and t_{chop}
 $t_{chop} = \max(1/V, \min(c_{chop}, c_{gap}))$;
 $\text{numOpenChops} = \lceil T/t_{chop} \rceil$;
 find all partitions of numOpenChops ;
 sort partitions by decreasing $\lceil \|\hat{B}\| \rceil$;
 $S = \text{zeros}(1, 2 * \text{numOpenChops})$;
foreach *partition* P **do**
 Compute $\lceil \|\hat{B}\| \rceil$ for P ;
 if $\lceil \|\hat{B}\| \rceil \leq \text{Fitness}(S)$ **then**
 | break;
 end
 $\hat{S} = \text{BestSequenceOfPartition}(P, V)$;
 if $\text{Fitness}(S, V) \leq \text{Fitness}(\hat{S}, V)$ **then**
 | $S = \hat{S}$;
 end
end

Algorithm 1. Overall shutter finding method

best PSF. Additionally, a time limit can be incorporated to produce a result within a budgeted amount of computation. At the completion of this process, the timing sequence with the highest PSF fitness is stored for use when the given object velocity and exposure time are required.

For our experiments, we have used this method to produce fluttering patterns for a wide range of combinations of subject velocity and required exposure time. These fluttering patterns are pre-computed and stored on the computer controlling the camera.

6 Experiments

In order to validate the claim that our velocity-dependant fluttering sequences provide increased image quality relative to the existing sampling method, we have carried out a number of experiments on real moving objects. Our coded exposure images are processed using the example code from [11] to produce a sharp image of the moving object. Because of the issues described in Sec. 4, we cannot use the 52 chop sequence given in [11] for comparison. Instead, we employ Raskar’s sampling method to determine new fluttering patterns with $t_{chop} \geq 1.25ms$, the shortest chop length allowable by the camera. For relatively short exposure times, the search space of potential binary sequences is small, and can be searched exhaustively. For a $4ms$ exposure time, for instance, using $t_{chop} = 1.33ms$ requires three open chops and the search space has only $2^6 = 128$ elements, of which the optimal sequence is 1101. We use this to capture an image of an object moving at 4 pixels per ms, and present the de-blurred results in Fig. 5 (left column).

Because the object moves through more than one pixel per t_{chop} , the extent of the PSF is greater than the sequence’s nominal PSF of $[0.33 \ 0.33 \ 0 \ 0.33]$,

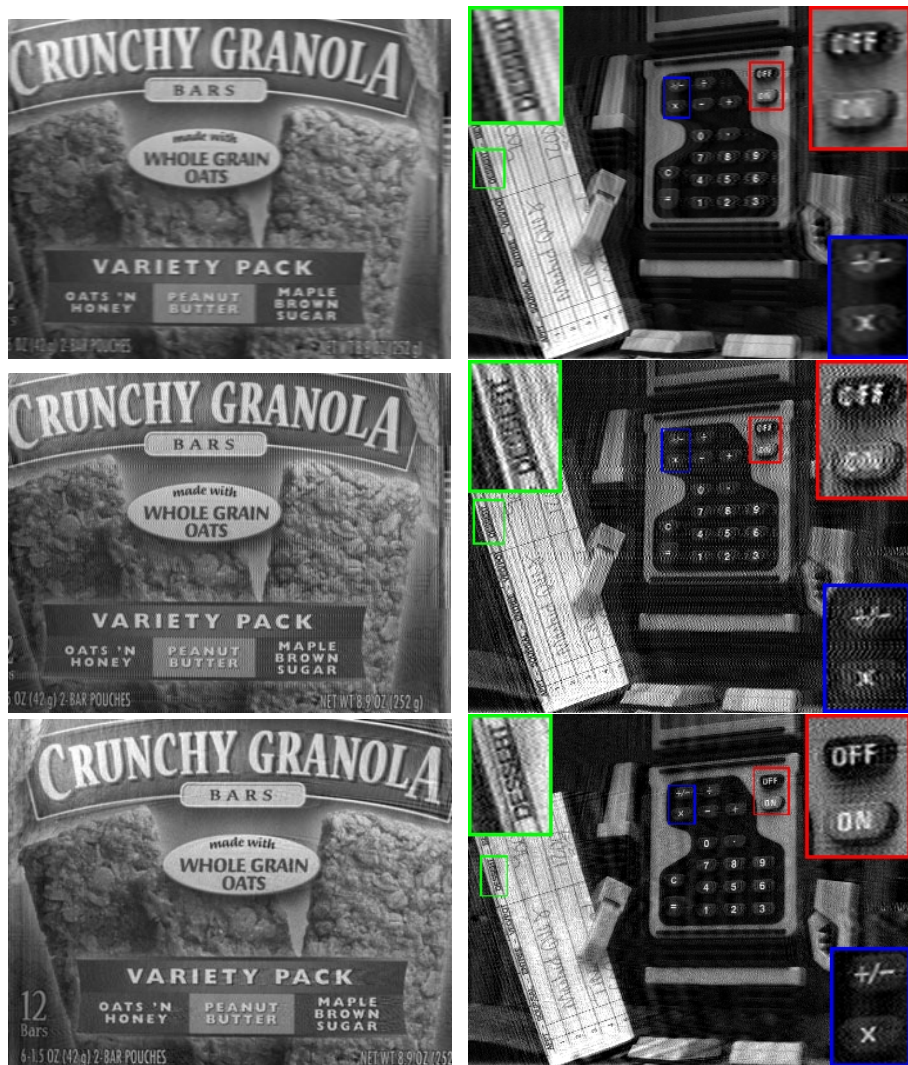


Fig. 5. Comparing de-blurred results to the existing method. (**Top row**) De-blurred flutter shutter image using the sampling and de-blurring methods of [11]. (**Centre row**) The same image, de-blurred using the effective PSF. (**Bottom row**) De-blurred flutter shutter image, acquired using a fluttering sequence determined by our method. Images in the left column are $4ms$ exposures of an object moving at 4 pixels per ms; image in the bottom row has 32 pixels of blur. Images in the right column (with annotated insets) are $8ms$ exposures of an object moving at 4 pixels per ms; image in the bottom row has 60 pixels of blur.



Fig. 6. (Top) Flutter shutter image captured of a car driving down a residential street, with 43 pixels of blur. (Bottom) Reconstructed image, in which details on the car are preserved; note that the static background has artifacts from the deconvolution. In this case, the street's speed limit serves as a strong prior on object velocity, obviating the need for explicit pre-capture velocity estimation. Exposure time is 20ms with velocity of 1.1 pixel per ms; shutter sequence is 1001001001100110001000111100011111111, with $t_{chop} = 1ms$.

and there are thus two choices for de-blurring. The approach taken in [11] is to re-sample the image to a smaller one in which the nominal PSF is the effective PSF. The nominal PSF is then de-convolved from this image, and the result is re-sampled in order to produce an image of the same size as the input. In this case, where the effective PSF is more than 4 times the length of the nominal PSF, the de-blurred image (top row of Fig. 5) has soft edges due to the final up-sampling by a factor of 4. In order to avoid this re-sampling step, we could instead de-convolve the effective PSF from the input image directly, as shown in Fig. 5 (middle row). As predicted by Lemma 1, however, this effective PSF is non-invertible and the resulting image has noticeable artifacts.

The de-blurred result of our shutter sequence is shown in Fig. 5 (bottom row), and avoids the artifacts due to lost frequencies while still preserving sharp edges in the scene. The shutter sequence used to capture the coded exposure image was 100000011111111000000010000011100000111, with $t_{chop} = 0.25ms$.

Fig. 5 (right column) shows a capture with an 8ms exposure time and object velocity of 4 pixels per ms. The fluttering pattern determined by sampling

is 110110000101 with $t_{chop} = 1.33ms$, and our computed shutter sequence is 111100001111100000111110000111110000100001111 with $t_{chop} = 0.33ms$. As before, the de-blurred image resulting from our shutter sequence maintains high-frequency information without significant lost frequency artifacts, whereas the fluttering pattern derived from sampling gives either soft edges or significant reconstruction artifacts, depending on the de-blurring approach. All three images in this column show artifacts at occlusion edges, similar to the images of [11].

Though we have argued for, and demonstrated the benefits of, velocity-dependant shutter sequences for motion capture, we have not presented a pre-capture velocity estimation method for shutter selection. While such a method would be helpful, it is not necessary in all situations. Fig. 6 shows, for example, the captured and de-blurred images of a car driving down a residential street. In this case, the street’s posted speed limit serves as a strong prior on a vehicle’s velocity, obviating the need for explicit motion estimation before image capture. It is unlikely that any vehicle will travel at twice the posted speed limit, meaning that the lost frequencies predicted by Lemma 1 are an unlikely problem. One can imagine, however, that a shutter sequence providing optimal reconstructions of a residential street will perform poorly on a highway.

7 Conclusions and Future Work

We have presented a method for determining velocity-dependant shutter sequences to capture coded exposure imagery of fast-moving objects. We have demonstrated that these shutter sequences produce higher quality de-blurred imagery than those determined by the existing random sampling method. This algorithm is motivated by the (heretofore unnoted) observation that a particular shutter sequence gives rise to a *family* of PSFs, with the effective PSF determined by the object’s velocity. We contribute an analytic proof that no shutter sequence can be devised that produces a family of invertible PSFs and that, in particular, a shutter sequence will produce non-invertible blur when the velocity is more than twice a nominal velocity. Our method for determining the optimal shutter sequence for a given combination of exposure time and object velocity is based on a priority search over the space of potential sequences, and features a termination condition that ensures optimality. We have also noted, measured, and incorporated a term in our optimisation to account for the fact that implementations of the flutter shutter based on electronic shutters incur read-out noise proportional to the number of open shutter periods.

Throughout these experiments, we have used manual estimation of the object velocity in order to select the appropriate fluttering sequence. In order to apply our method in unconstrained settings, this step should be performed automatically before initiating image capture. Though this step is non-trivial, we expect that the large body of literature on tracking and motion estimation will yield a workable approach, given two facts. First, real-world moving objects have inertia which precludes sudden changes of direction and velocity. Second, we note that all cameras already have meters that estimate a quantity (illumination) that potentially changes much quicker than velocity. It should also be noted that many

cameras/lenses already have sensors that provide real-time motion estimates for optical image stabilisation, and that accurate velocity estimation obviates the need for explicit blur estimation from the image, as the shutter pattern and estimated velocity combined determine the extent of the PSF.

Acknowledgements

This material is based upon work supported by the U.S. Army Biometrics Task Force under Contract No. W91CRB-09-C-0013. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author and do not necessarily reflect the views of the U.S. Army. Flea[®]2 is a trademark of Point Grey Research, Inc. All brand names and trademarks used herein are for descriptive purposes only and are the property of their respective owners.

References

1. Agrawal, A., Raskar, R.: Optimal single image capture for motion deblurring. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 2560–2567 (2009)
2. Haykin, S.: Blind Deconvolution. Prentice-Hall, Englewood Cliffs (1994)
3. Jia, J.: Single image motion deblurring using transparency. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
4. Shan, Q., Jia, J., Agarwala, A.: High-quality motion deblurring from a single image. In: ACM SIGGRAPH (2008)
5. Hasinoff, S.W., Kutulakos, K.N.: Light-efficient photography. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 45–59. Springer, Heidelberg (2008)
6. Hasinoff, S.W., Kutulakos, K.N., Durand, F., Freeman, W.T.: Time-constrained photography. In: ICCV (2009)
7. Telleen, J., Sullivan, A., Yee, J., Wang, O., Gunawardane, P., Collins, I., Davis, J.: Synthetic shutter speed imaging. *Computer Graphics Forum* 26, 591–598 (2007)
8. Ben-ezra, M., Nayar, S.K.: Motion deblurring using hybrid imaging. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 657–664 (2003)
9. Levin, A., Fergus, R., Fergus, R., Durand, F., Freeman, W.T.: Image and depth from a conventional camera with a coded aperture. In: ACM SIGGRAPH (2007)
10. Levin, A., Sand, P., Cho, T.S., Durand, F., Freeman, W.T.: Motion-invariant photography. In: ACM SIGGRAPH (2008)
11. Raskar, R., Agrawal, A., Tumblin, J.: Coded exposure photography: motion deblurring using fluttered shutter. *ACM Trans. Graph.* 25, 795–804 (2006)
12. Caroli, E., Stephen, J., Cocco, G., Natalucci, L., Spizzichino, A.: Coded aperture imaging in x- and gamma-ray astronomy. *Space Science Reviews* 45, 349–403 (1987)
13. Agrawal, A., Xu, Y.: Coded exposure deblurring: Optimized codes for psf estimation and invertibility. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 2066–2073 (2009)