

# Improved Motion Invariant Deblurring through Motion Estimation

Scott McCloskey

Honeywell Labs, USA

**Abstract.** We address the capture of sharp images of fast-moving objects, and build on the Motion Invariant photographic technique. The key advantage of motion invariance is that, unlike other computational photographic techniques, it does not require pre-exposure velocity estimation in order to ensure numerically stable deblurring. Its disadvantage is that the invariance is only approximate - objects moving with non-zero velocity will exhibit artifacts in the deblurred image related to tail clipping in the motion Point Spread Function (PSF). We model these artifacts as a convolution of the desired latent image with an error PSF, and demonstrate that the spatial scale of these artifacts corresponds to the object velocity. Surprisingly, despite the use of parabolic motion to capture an image in which blur is invariant to motion, we demonstrate that the motion invariant image can be used to estimate object motion *post-capture*. With real camera images, we demonstrate significant reductions in the artifacts by using the estimated motion for deblurring. We also quantify a 96% reduction in reconstruction error, relative to a floor established by exact PSF deconvolution, via simulation with a large test set of photographic images.

## 1 Introduction

Motion deblurring is one of the longest standing problems in computational photography, and has been addressed in a number of ways. Classic blind deblurring - estimating blur and then applying deconvolution - has long been used on images captured with traditional cameras. However, it is well-known that deconvolving very common motions - such as vehicles moving at a constant velocity - is ill-conditioned under traditional capture methods. In order to address this, various computational cameras have been proposed to capture *invertible* motion blur. The flutter shutter [18] does this by capturing a multiple exposure image, but pre-exposure velocity estimation is required to ensure invertibility [13].

In order to avoid pre-exposure motion estimation, Levin et al. [12] proposed *motion invariant photography*, whereby the camera captures an image in which the blur PSF is both invertible and approximately invariant to subject velocity. This approximate invariance holds over a range of velocities, so a reasonably good image can be produced without the need for pre-exposure velocity estimation. However, the approximate nature of the invariance results in certain artifacts (see Fig. 1) which are aesthetically unpleasant.



**Fig. 1.** (Top left) A motion invariant image, deblurred as in previous work. The invariance approximation results in artifacts, most notable in the wheel area. (Top right) Using our motion estimation algorithm, we avoid these artifacts and improve the quality of the reconstruction. (Bottom row) zoomed views of annotated regions, plus a traditional image of the moving object to convey the extent of motion.

We model motion invariant capture and deconvolution in order to demonstrate, for the first time, that these artifacts are indicators of the underlying velocity. Whereas the image captured by the camera is approximately motion invariant, we show that the artifacts in a processed version of that image are distinctive enough to estimate the object velocity. We propose a motion invariant motion estimation algorithm which determines velocity by matching edges to artifacts arising therefrom. This estimate is then be used to re-process the image, reducing artifacts and improving the final image quality. We demonstrate the qualitative improvements - as in Fig. 1 - using a real camera whose stabilizing lens is moved to implement motion invariance. By synthetically blurring and then processing images from a public dataset, we quantify an 96% reduction in reconstruction errors, relative to a floor established by exact PSF deconvolution, as a result of our blur estimation.

## 2 Previous Work

Our work builds on Levin et al. [12], who introduced motion invariance and later extend it to cases where the motion direction is not known [4], by capturing two images with orthogonal motion. A key step to 2D motion invariance is blur estimation *from a pair of images*; we estimate motion from a single parabolic exposure, and note that many of our results (e.g. Fig. 1) involve objects moving too fast to capture in multiple frames. Webster and Dorrel [21] argue for the inclusion of a variable opacity optical element in motion invariant cameras to reduce artifacts arising from PSF variance, but do not support their simulations with real images and do not model the effect of light loss through the additional optical element. More recently, other implementations of motion invariance have been described in the literature. In previous work, we [15] describe an implementation of motion invariance using the image stabilization hardware in a Canon

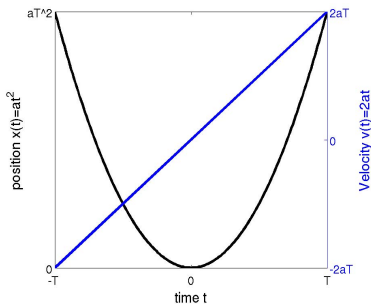
DSLR lens. The actual motion that the lens undergoes during exposure, combined with prior information about camera and/or subject motion, is shown to improve the quality of the deblurred image in [16]. Sonoda et al. [19] describe an implementation of motion invariance using a Liquid Crystal on Silicon (LCoS) element, though that implementation involves about three stops (90%) of light loss due to the combined effects of polarization, fill factor, reflectivity, and the reduction of the reflective region. All of the previous work in motion invariance deconvolves the PSF of a stationary object and, as we show, will introduce certain artifacts when objects are in motion. We introduce a post-capture motion estimation algorithm which greatly reduces these artifacts.

Besides motion invariance, several computational photographic techniques have been developed to address motion blur. Notably, coded exposure [18] addresses the lack of motion blur invertibility in traditional imaging, but its lack of invariance requires pre- and/or post-capture motion estimation. Coded exposure blur estimation has been developed using either edge priors [2] or natural image statistics [14]. Coded exposure and motion invariance have been compared with respect to noise performance and other criteria [1], though that analysis includes the artifacts which we reduce by motion estimation. Both coded exposure and motion invariance have analogous approaches for handling optical blur, as well. Coded aperture photography [11,20] selectively occludes light paths to the sensor, and has been used for extended depth of field imaging. Similar to motion invariance, flexible depth of field photography [17] captures a coded image using lens motion, as the lens moves along the optical axis to make defocus blur invariant over an depth range. More recently, Bando et al. [3] analyze these computational cameras in the framework of a time-varying light field, and come to the surprising conclusion that focal sweep is optimal for both defocus and motion blur. However, their motions are smaller than the ones considered here, and the analysis depends on the ‘infinite exposure’ assumption, and thus doesn’t account for motion invariant artifacts.

Despite the associated invertibility problems, there is a large amount of research addressing both blur estimation and deblurring of images from traditional cameras. Recently, work in this area has addressed spatially variant blur arising from camera rotations [22,5]. Interestingly, experiments by Köhler et al. with a dataset of hand-held camera motions [8] have shown that these methods don’t necessarily out-perform those based on a shift-invariant model of motion blur. In fact, the work of Xu and Jia [23] performed best despite assuming uniform blur. However, [8]’s evaluation results on blind deconvolution do not necessarily reflect performance on motion invariant deblurring, where the invariant PSF is known ahead of time. In our experiments, we use the method of Krishnan and Fergus [9] for the non-blind deconvolution of motion invariant images, and note that it performs well in the blind deconvolution evaluation.

### 3 Motion Invariant Capture

The objective of motion invariant image capture is to obviate PSF estimation, thus avoiding *blind* deconvolution, by capturing an image with a motion blur PSF



**Fig. 2. Motion invariant capture:** parabolic motion in position versus time (black curve/axis) capturing an image where velocities in a range are stabilized for the same period of time due to the linearity of velocity as a function of time (blue line/axis)

that does not depend on the velocity of a moving object. For an object with any particular velocity, of course, motion blur can be prevented by translating the lens/camera in the same direction and with the same speed as the object in order to stop motion, i.e. to ensure that its projection on the sensor does not move. The intuitive explanation of motion invariant capture is that, by translating the lens with constant acceleration along the object motion direction, all objects stopped momentarily, and these objects will have approximately the same PSF.

Motion invariance can be implemented by translating any of the camera body, sensor, or lens with constant acceleration. We use optical stabilization hardware to implement motion invariance by lens motion, as suggested by Levin et al. in [12,4]. In the 1D case, the lens moves along a line in the direction of expected motion; without loss of generality, we will discuss horizontal lens motion with initial rightward velocity. At the beginning of the image’s exposure, the lens translates right with a given velocity, and constant (negative) acceleration  $a$  is applied. During an exposure duration of  $2T$  milliseconds, with the time variable  $t \in [-T, T]$  for mathematical convenience, the acceleration causes the lens to come to a stop at  $t = 0$  and, at the end of the exposure period ( $t = T$ ), the lens has returned to its initial position with the same velocity magnitude (but in the opposite direction) as in the beginning. Though the motion of the lens is linear, this pattern is referred to as *parabolic motion* because the horizontal position  $x$  is a parabolic function of time  $t$ ,

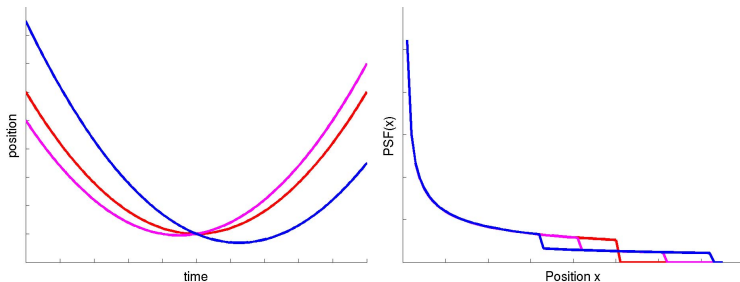
$$x(t) = at^2. \quad (1)$$

Figure 2 shows parabolic motion and lens velocity (the derivative of  $x$ ), illustrating the range of object velocities which are stabilized during exposure.

As in other work, blur in a motion invariant image is modeled as the convolution of a latent image  $I$  with a PSF  $B$ , giving the blurred image

$$I_b = I * B + \eta, \quad (2)$$

where  $\eta$  represents noise. While deconvolution methods based on this model can perform quite well in practice, the model is mostly limited to subject motion. Because motion arising from unsteady hands holding the camera generally involves some rotation [8], this model is best suited to stationary camera scenarios. The 1D motion invariance considered here is also limited to constant velocity



**Fig. 3.** (Left) Trajectories of points with different velocities as observed on the sensor of a lens undergoing parabolic motion. (Right) Using the PSF model of eq. 4, non-zero velocities result in clipped tails of the PSFs.

motion, which has been shown to hold approximately when the short exposure duration allows for good approximations of slight acceleration/deceleration [12].

As discussed in [12], there are tradeoffs in adopting motion invariant photography over traditional imaging. First, stationary objects in the scene will be blurred in the image due to the motion of the lens and, while deconvolving  $B$  is numerically stable, deblurring amplifies noise  $\eta$  everywhere. The second issue is that the convolution model of eq. 2 does not hold at occluding contours, so artifacts arise when both foreground and background are textured. Our experiments show that motion estimation reduces these artifacts. Finally, and most importantly for our purposes, the invariance of  $B$  to velocity is approximate, and fails entirely when the object’s projection on the sensor moves too fast to be stabilized. For different object velocities which are stabilized, their PSFs have some variation which leads to artifacts when the wrong PSF is deblurred.

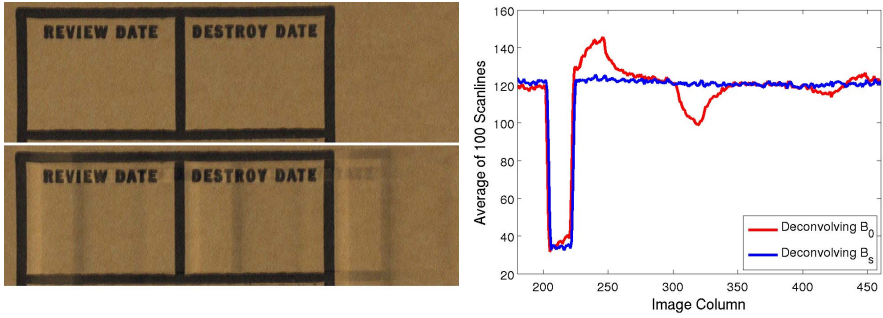
### 3.1 Motion Invariance Artifacts

In this section, we elaborate on the variation in the PSF induced by changing object velocity, and model the artifacts that arise therefrom. Consider the effect of subject motion of a point  $P$  on its PSF  $B$ , when that motion is linear with constant velocity  $s$ . During exposure, the motion of  $P$  on the sensor combines  $P$ ’s real-world motion and parabolic motion of the lens (eq. 1), as

$$x(t) = at^2 + st. \quad (3)$$

Fig. 3 shows these trajectories for three different values of  $s$ . Because the value of the PSF  $B_s$  at a particular sensor point is proportional to the amount of time  $P$ ’s reflected/emitted light lands there, we can derive the continuous PSF as in [12] (with slightly modified notation)

$$B_s(x) = \begin{cases} \left(a \left(x + \frac{s^2}{4a}\right)\right)^{-1/2} & \text{if } 0 \leq x + \frac{s^2}{4a} \leq a \left(T - \frac{s}{2a}\right)^2 \\ \frac{1}{2} \left(a \left(x + \frac{s^2}{4a}\right)\right)^{-1/2} & \text{if } a \left(T - \frac{s}{2a}\right)^2 < x + \frac{s^2}{4a} \leq a \left(T + \frac{s}{2a}\right)^2 \\ 0 & \text{otherwise} \end{cases} \quad (4)$$



**Fig. 4.** Example of motion invariant artifacts. (Top left) Reference image of a cardboard box. (Bottom left) When the motion invariant image of the moving box is deblurred with  $B_0$ , printed regions ‘echo’ in the image. (Right) Plots of the average intensity over 100 rows of pixels covering the right-most printed edge and its echo, when the motion invariant image is deblurred with  $B_0$  (red line) and the actual PSF  $B_s$  (blue line).

Note that, when  $s \neq 0$ , the change from the first to the second clauses results in a halving of the PSF, which is described as ‘tail clipping’. This arises because, while the lens returns to its original position by the end of the exposure period, the light from  $P$  will be displaced by a distance of  $2sT$ . Fig. 3 illustrates this tail clipping effect on the PSFs corresponding to the three parabolas shown.

While eq. 4 indicates that  $B$  depends on velocity, motion invariant images in all previous work have been deblurred using the PSF  $B_0$  of stationary objects. The differences between a moving object’s actual PSF and the one used in deblurring will introduce certain artifacts, as illustrated in Fig. 4. In this example, the printed lines on the box appear reasonably sharp, but regions which should appear uniform brown instead have intensity changes which appear as ‘echos’ of the printed lines. First, there is a vertical band of brighter pixels immediately to the right of the printed edges. Next there is a darker band of pixels about 100 pixels to the right of the printed edges. Finally, there are adjacent dark and bright bands beginning about 200 pixels to the right of the printed edges. The plot shows the average intensity over 100 rows of pixels spanning the right-most printed edge and its echo, to give a better sense of the artifact’s magnitude. We can avoid these artifacts by deconvolving the actual PSF  $B_s$  from the motion invariant image, as shown by the blue line in the plot.

What we will show in the remainder of this section is that - despite the objective of capturing an image which is invariant to object motion - the artifacts arising from deconvolving  $B_0$  are, themselves, a cue to subject motion. This surprising result is due to the fact that the effect of deconvolving  $B_0$  is the same as convolving the true latent image with an error PSF which has peaks at locations uniquely determined by the object velocity  $s$ . To show this, we express the convolution blur model of eq. 2 in the Fourier domain as

$$\hat{I}_b = \hat{I} \cdot \hat{B}_s + \hat{\eta}, \quad (5)$$

where symbols under the  $\hat{\cdot}$  are the Fourier representations of the image/PSF/noise and  $\cdot$  represents multiplication. We use  $\mathcal{F}$  to represent the Fourier transform, so  $\hat{I} = \mathcal{F}(I)$ . By deconvolving  $B_0$  from  $I_b$ , we compute a latent image estimate

$$J_0 = \mathcal{F}^{-1} \left( \frac{\hat{I}_b}{\hat{B}_0} \right) = I * \mathcal{F}^{-1} \left( \frac{\hat{B}_s}{\hat{B}_0} \right) + \mathcal{F}^{-1} \left( \frac{\hat{\eta}}{\hat{B}_0} \right). \quad (6)$$

So the latent image estimated by deconvolving  $B_0$  is itself a convolution of the true latent image with an error PSF

$$E_s = \mathcal{F}^{-1} \left( \frac{\hat{B}_s}{\hat{B}_0} \right), \quad (7)$$

plus the image noise amplified by deconvolution. Recall that  $B_0$  satisfies the invertibility conditions discussed in [18], so the deconvolution is numerically stable (i.e., there is no division by 0). Obviously, for stationary objects  $B_s = B_0$ , the error PSF is the  $\delta$  function, and the latent image estimate matches the true latent image (modulo amplified noise). When the object is moving ( $s \neq 0$ ),  $\hat{B}_s \neq \hat{B}_0$ , and the error PSF introduces the artifacts illustrated above.

Example error PSFs are shown in Fig. 5, using the same colors/speeds as in Fig. 3. This figure illustrates how the error PSF - and thus the artifacts in the initial latent image estimate - relate to the ground truth PSF. The negative peaks in the error PSF occur in two places: first, at the location where the PSF  $B_s$ 's tail clips

$$x + \frac{s^2}{4a} = a \left( T - \frac{s}{2a} \right)^2, \quad (8)$$

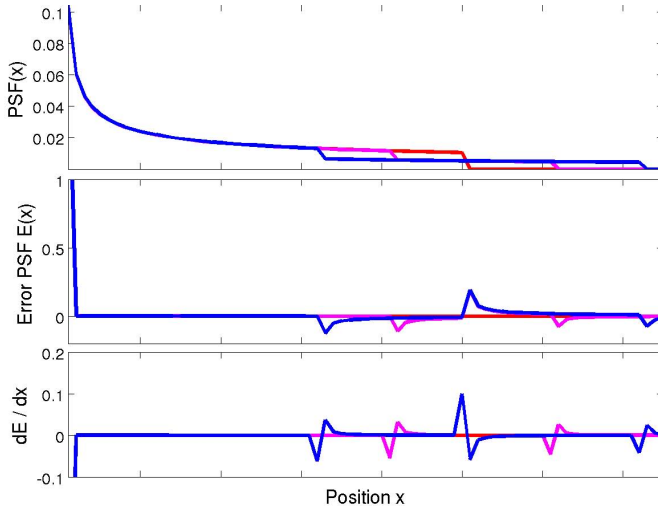
and, second, at the location where  $B_s$  goes to zero

$$x + \frac{s^2}{4a} = a \left( T + \frac{s}{2a} \right)^2. \quad (9)$$

The error PSF also has a positive peak at  $x = aT^2$ , where the  $B_0$  goes to zero. The key to understanding our motion invariant motion estimation algorithm is that the distance between the two negative peaks in the error PSF  $E_s$  is exactly  $2sT$ . At a high level, then, our algorithm for motion estimation computes the initial latent image estimate  $J_0$ , localizes artifacts in this image, and uses the peak-to-peak distance to estimate  $s$ . We then deblur the original motion invariant image again - this time with the exact PSF - to get an artifact-free latent image. The next section describes our method in greater detail.

## 4 Artifact-Based Motion Estimation

Having shown that deblurred motion invariant images contain artifacts which reflect the underlying object speed, it remains to be shown that these artifacts can be reliably detected. The box image shown in Fig 4 is a simple case, in that the background is a uniform intensity brown; a general solution must address



**Fig. 5.** (Top) Three motion invariant PSFs for velocities  $s = 0$ ,  $\frac{aT}{10}$ , and  $-\frac{aT}{5}$  in red, magenta, and blue, respectively. (Middle) The corresponding error PSFs  $E_s$ : the red curve is the delta function, whereas the blue and magenta curves have positive/negative peaks (and their positive peaks overlap). Comparing the top two plots, we see that the first negative peak in the error PSF coincides with the start of tail clipping, the second with the last non-zero entry, and the positive peak in between occurs at the length of the PSF  $B_0$ . (Bottom) The derivative of the error PSF introduces distinctive sign changes in the image’s gradient magnitude at these same locations.

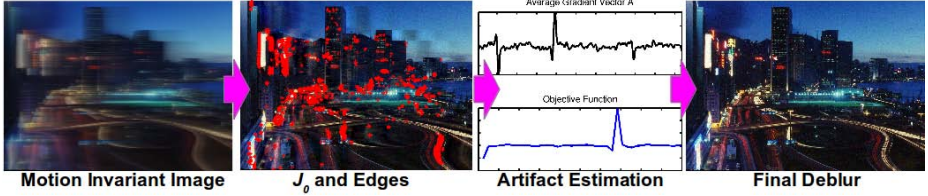
detection in regions of natural image texture. To handle this, we employ the well-known sparse gradient distribution used in [11] and elsewhere. One consequence of this prior is that, if we average the horizontal image gradient over a set of pixel locations *in a well-focused natural image*, the value will tend to 0 with an equal probability of negative and positive values. What happens when, instead of doing this to a well-focused natural image, we average horizontal gradient values in the initial latent image estimate  $J_0$ ? Per eq. 6,

$$\frac{dJ_0}{dx} = \frac{d\left(I * \mathcal{F}^{-1}\left(\frac{\hat{B}_s}{B_0}\right)\right)}{dx} + \frac{d\left(\mathcal{F}^{-1}\left(\frac{\hat{\eta}}{B_0}\right)\right)}{dx}. \quad (10)$$

The noise term, even after deconvolving  $B_0$ , is still zero mean and does not bias the gradient to positive or negative values. However, the first term is equivalent to convolving the sharp latent image (with no bias for positive or negative gradients) with the horizontal derivative of the error PSF. As shown in Fig. 5, this kernel induces a sign change in the derivative image *at the same three locations as the error PSF has peaks*.

As with other work in PSF estimation [7], and following the intuition of Fig 4, we use edge points in the image to infer the scale of the PSF. Specifically, we look for pronounced sign changes in the average gradient magnitude of pixels over a range of horizontal distances from edges in the image. If the object attached to





**Fig. 6.** An overview of our motion estimation method. The motion invariant image is deblurred using  $B_0$ . Edges are detected in the resulting image, and artifacts arising from them are used to estimate  $s$ . Finally, we deconvolve the exact PSF  $B_s$  from the original motion invariant image.

the edge was moving ( $s \neq 0$ ), there will be a zero-crossing of the gradient at a distance of  $aT^2$  from the edge, and additional zero-crossings separated by  $2sT$ .

The steps of our algorithm are as follows, and are illustrated in Fig. 6.

1. Compute the initial latent image estimate  $J_0$  by deconvolving  $B_0$ .
2. Find vertical edges in  $J_0$  using the Sobel detector. Discard edge points below the 80th percentile strength.
3. Compute  $H = \frac{dJ_0}{dx}$ .
4. Compute the average gradient sign vector  $A$ , weighted by the strength of edge points  $x$ , as

$$A(i) = \frac{\sum_x |H(x)| \text{sign}(H(x+i))}{\sum_x |H(x)|} \quad \forall i \in [0, 4aT^2]. \quad (11)$$

5. **If**  $\sum_{x=aT^2-1}^{aT^2+1} A(x)^2 < \tau$ ,  $\tilde{s} = 0$  and return; **else**

$$\tilde{s} = \underset{s \in S}{\text{argmax}} \sum A \cdot \frac{dE_s}{dx}. \quad (12)$$

The threshold  $\tau = 5 * 10^{-10}$  is used to determine whether there is a zero-crossing at  $aT^2$ . The space of potential velocities is quantized to the set  $S$  such that consecutive values shift the peaks in the error PSF  $E$  by 1 pixel and  $s \leq 2aT$ , which is the maximum velocity that the lens stabilized.

## 5 Experiments

In order to validate our motion estimation algorithm, and to demonstrate the improvement in image quality, we describe two experiments. First, we apply uniform blur PSFs representing a range of velocities to images from the IM2GPS [6] dataset in order to quantify motion estimation performance and image quality improvement in the presence of a wide range of image texture. In the second, we capture motion invariant images using a Canon 60D camera with a 100mm image stabilizing lens which has been modified to execute parabolic motion during exposure. Note that, with the exception of data presented in Sec. 5.1, all images shown in this paper are real images from our Canon camera.

**Table 1.** RMSE of images deblurred using  $B_0$  (showing the performance of existing MI deblurring),  $B_{\tilde{s}}$  (i.e., our estimate), and  $B_s$  (i.e., the ground truth PSF and lower bound of RMSE). Rows separate noise conditions, and we show separate RMSE performance for those cases where our motion estimate is inaccurate.

noise	test cases	$B_0$	$B_{\tilde{s}}$	$B_s$
$\sigma = 1$	all	14.52	10.77	10.66
$\sigma = 1$	$\tilde{s} \neq s$	14.75	15.40	10.48
$\sigma = 2.5$	all	18.96	16.24	16.11
$\sigma = 2.5$	$\tilde{s} \neq s$	18.99	19.53	16.01

## 5.1 Synthetic Data

In order to avoid the small sample size issues associated with presenting only real camera results, we first present results on the 163 landscape-oriented images from the IM2GPS [6] test set. We synthetically blur the entire image using the PSF  $B_s$  (eq. 4), with  $T = 50ms$  (i.e., a 100ms exposure time),  $a = 0.04^1$ , and  $s$  uniformly sampled in  $[0, 2aT]$ . We test 21 different values of  $s$  for each image. Blur is simulated using eq. 2, and we test two levels of Gaussian white noise  $\eta$  with  $\sigma = 1$  and 2.5 relative to 8 bit intensity values.

Under the  $\sigma = 1$  case, we estimate  $s$  to within the resolution of our quantization of  $S$  in 3,346 of 3,423 cases (98%). Under the  $\sigma = 2.5$  condition, we do so in 3,299 of 3,423 cases (96%). Most of the errors come from two images, where the algorithm always estimates  $\tilde{s} = 0$ . In one, the ‘latent image’ contains motion blur - a clear violation of the natural image prior. The second has no strong edges from which to estimate motion.

Table 1 shows the computed Root Mean Squared Error (RMSE) between the reference latent image and motion invariant images deblurred using  $B_0$  (showing the performance of existing MI deblurring),  $B_{\tilde{s}}$  (i.e., our estimate), and  $B_s$  (i.e., the ground truth PSF and lower bound of RMSE). When averaged over all image/velocity combinations, the use of our estimate in deblurring eliminates 96% of the RMSE relative to the use of the ground truth  $s$ . In the very few cases where our velocity estimate is off by one or more quantization level ( $\tilde{s} \neq s$ ), the resulting RMSE is only slightly worse than the state of the art.

## 5.2 Real Camera Images

Our motion invariant camera is made of a stock Canon 60D DSLR body with a modified Canon EF 100mm f/2.8L Macro IS lens. As described in [15], the standard mode of operation for image stabilization is a closed control loop where motion detected by sensors within the lens induces a compensating motion of the stabilizing element. Lens motion is achieved by modulating electromagnets,

<sup>1</sup> These values produce a PSF  $B_0$  with 100 non-zero entries. We choose this so the ratio of PSF length to image width is consistent with our camera; they are about twice as wide as IM2GPS images, and our camera produces a PSF  $B_0$  that’s 200 pixels in length.



**Fig. 7.** Our prototype Motion Invariance camera, based on an unmodified Canon 60D with a modified 100mm Macro IS lens. The hot shoe triggers our Texas Instruments MSP430F2618 microprocessor (located inside the metal box) to start parabolic lens motion.

and its position is recorded using two embedded position sensors. In order to drive the lens to parabolic motion, we break the control loop and decouple the stabilizing element from the motion sensor. We then add an independent Texas Instruments MSP430F2618 microprocessor which controls motion through pulse width modulation, and monitors its progress by reading the position sensors through 12-bit ADCs. Control loops running on our added microcontroller execute the desired parabolic motion, which is synchronized with the camera’s shutter via the hot shoe trigger. An image of the camera is shown in Figure 7, and additional detail about the hardware implementation is given in [16].

While the stabilizing lens accurately traces the desired parabolic trajectory, our real image PSFs differ from the analytic form due to a delay in the start of motion. The issue is that our motion starts based on the camera body’s hot shoe, which is designed for flash triggering. In order to avoid flash synchronization issues - particularly a flash firing when the shutter is not completely open - the hot shoe’s first signal occurs when the first curtain is completely open. On our 60D body, the transit of the shutter blade from bottom to top takes about 4ms, so the lens will be stationary for as much as the first 4ms of exposure. The example in Fig. 1 was a 100ms exposure, so the amount of time the lens is stationary is comparable to the amount of time that it stabilizes any moving object. When estimating blur using images from this camera, we add an additional delay term in  $B$  to model the correct error PSFs.

In addition to Fig. 1, the deblurred results in Fig. 8 show the improved image quality that we enable with our motion estimation algorithm. Images in the left column, which are generated by deconvolving  $B_0$ , have significant echo artifacts due to the use of the incorrect PSF. Our results, in the right column, have significantly reduced echo artifacts, based on deconvolution of the correct PSF as determined by our motion estimation algorithm.

## 6 Limitations

Like previous work in Motion Invariance, our method assumes that objects move in a linear direction which is known a priori. The linear motion assumption holds up well in practice as long as the camera is stationary, since moving objects have inertia which precludes rapid direction changes. A priori knowledge of the motion direction is available in many cases: vehicles drive within lanes on roads, people walk down hallways and along sidewalks, etc. The rear hubcap in the bottom row of Fig. 8, which is rotating rather than translating, violates this assumption and results in slight artifacts. As well, the limited dynamic range of the sensor



**Fig. 8. Experimental results with our real camera.** Left column shows the result of deconvolving  $B_0$ , as in previous motion invariant work. Right column shows the result of deconvolving  $B_s$ , after applying our motion estimation algorithm. Echo artifacts are noticeably reduced when deconvolving  $B_s$ . **Best viewed electronically.**

causes some artifacts in deblurred images; in our case, this is evident in the top row of Fig. 8, where the eye at the front of the bus has some artifact due to saturated pixels in the motion invariant image. In both cases, though, the severity of artifacts is reduced by our method relative to the existing approach.

As discussed above, motion invariance amplifies noise on stationary parts of the scene. By deconvolving a PSF  $B_s$  with  $s \neq 0$ , we may further exacerbate this issue (see the blue car in the top row of Fig. 8).

## 7 Multiple Moving Objects

While our algorithm is designed to estimate the velocity of a single, dominant object in the scene, we also note that the method performs well in cases where multiple objects have different velocities. This is partially due to the fact that the PSF depends only on the magnitude of object velocity, so  $B_{-s} = B_s$ . In the general case when there are two objects with velocities  $s_1$  and  $s_2$ , our algorithm



**Fig. 9. Multiple moving objects.** Here, the green train moves faster than the red, and its image velocity is even higher due to motion parallax. Our motion estimation algorithm finds the velocity of the green train, and our result (right image) significantly reduces echos of the wheels compared to the existing method (left image). Though the red train is deblurred with the PSF of the green train, this is still a better approximation than  $B_0$ , so artifacts are also reduced on the red train (e.g., reduced echo of the numeral 5 at the left edge of the coal car and reduction of artificial highlight to the right of the 5). **Best viewed electronically.**

will estimate the velocity of the larger object (i.e., the one with the most edge points), say  $s_1$ , due to the use of *argmax* in eq. 12. Because the point where the PSF’s tail clips moves continuously with velocity, our method will also improve (in the sense of RMSE) the estimate of the smaller object in cases where  $\|s_2\| > \|\frac{s_1}{2}\|$ . An example of this is shown in Fig. 9, where two moving objects have different velocities and our method produces better latent image estimates for both, as compared to the existing method. In future work, we hope to extend the method to incorporate explicit motion segmentation, perhaps along the lines of Levin [10], though we note that this is more challenging from motion invariant images due to the blurring of stationary scene elements.

## 8 Conclusion

Surprisingly, despite the use of parabolic motion to capture an image in which blur is decoupled from an object’s velocity, we have successfully demonstrated a motion estimation algorithm using motion invariant images. We further show that using our algorithm to determine which PSF should be deconvolved significantly reduces both quantitative RMSE and the severity of motion invariant artifacts, using both a large-scale synthetic experiment and validation with real camera images. Using this method, we get the benefit of deconvolving the exact PSF (in the 96+% of cases where we accurately estimate motion) *without* the need for pre-exposure velocity estimation, as in coded exposure. The key to enabling this is our modeling of the artifacts introduced by the traditional motion invariant approach, and quantifying the relationship between velocity and the spacing of these artifacts.

Relative to other work attempting to improve motion invariant image quality, notably [21], we achieve better image quality without the need for additional hardware elements or the associated light loss. Our prototype camera does not introduce additional elements in the optical path, and therefore maintains high light throughput.

Knowing that parabolic motion does not capture an image with strict motion invariance, and having shown that this can lead to significant artifacts, one may ask whether this approach is still worthwhile. We believe that it is, because the parabolic motion produces a PSF with a sharp peak which makes deconvolution well-posed without the need for pre-exposure motion estimation. When combined with our post-capture motion estimation, we can produce high-quality images that would not be possible with traditional camera hardware.

## References

1. Agrawal, A., Raskar, R.: Optimal single image capture for motion deblurring. In: *Computer Vision and Pattern Recognition*, pp. 2560–2567 (2009)
2. Agrawal, A., Xu, Y.: Coded exposure deblurring: Optimized codes for PSF estimation and invertibility. In: *Computer Vision and Pattern Recognition* (2009)
3. Bando, Y., Holtzman, H., Raskar, R.: Near-invariant blur for depth and 2d motion via time-varying light field analysis. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 13:1–13:15 (2013)
4. Cho, T.S., Levin, A., Durand, F., Freeman, W.T.: Motion blur removal with orthogonal parabolic exposures. In: *Int'l Conf. on Computational Photography* (2010)
5. Gupta, A., Joshi, N., Lawrence Zitnick, C., Cohen, M., Curless, B.: Single image deblurring using motion density functions. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) *ECCV 2010, Part I. LNCS*, vol. 6311, pp. 171–184. Springer, Heidelberg (2010)
6. Hays, J., Efros, A.A.: Im2gps: estimating geographic information from a single image. In: *Computer Vision and Pattern Recognition* (2008)
7. Joshi, N., Szeliski, R., Kriegman, D.: Psf estimation using sharp edge prediction. In: *Computer Vision and Pattern Recognition* (2008)
8. Köhler, R., Hirsch, M., Mohler, B., Schölkopf, B., Harmeling, S.: Recording and playback of camera shake: benchmarking blind deconvolution with a real-world database. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) *ECCV 2012, Part VII. LNCS*, vol. 7578, pp. 27–40. Springer, Heidelberg (2012)
9. Krishnan, D., Fergus, R.: Fast image deconvolution using hyper-laplacian priors. In: *NIPS* (2009)
10. Levin, A.: Blind motion deblurring using image statistics. In: *NIPS*, pp. 841–848 (2006)
11. Levin, A., Fergus, R., Fergus, R., Durand, F., Freeman, W.T.: Image and depth from a conventional camera with a coded aperture. In: *SIGGRAPH* (2007)
12. Levin, A., Sand, P., Cho, T.S., Durand, F., Freeman, W.T.: Motion-invariant photography. In: *SIGGRAPH* (2008)
13. McCloskey, S.: Velocity-dependent shutter sequences for motion deblurring. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) *ECCV 2010, Part VI. LNCS*, vol. 6316, pp. 309–322. Springer, Heidelberg (2010)

14. McCloskey, S., Ding, Y., Yu, J.: Design and estimation of coded exposure point spread functions. *IEEE Trans. Pattern Analysis and Machine Intelligence* 34(10), 2071–2077 (2012)
15. McCloskey, S., Muldoon, K., Venkatesha, S.: Motion invariance and custom blur from lens motion. In: *Int'l Conf. on Computational Photography* (2011)
16. McCloskey, S., Muldoon, K., Venkatesha, S.: Motion aware motion invariance. In: *Int'l Conf. on Computational Photography* (2014)
17. Nagahara, H., Kuthirummal, S., Zhou, C.Y., Nayar, S.K.: Flexible Depth of Field Photography. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part IV*. LNCS, vol. 5305, pp. 60–73. Springer, Heidelberg (2008)
18. Raskar, R., Agrawal, A., Tumblin, J.: Coded exposure photography: motion deblurring using fluttered shutter. *ACM Trans. on Graphics* 25(3), 795–804 (2006)
19. Sonoda, T., Nagahara, H., Taniguchi, R.-I.: Motion-invariant coding using a programmable aperture camera. In: Lee, K.M., Matsushita, Y., Rehg, J.M., Hu, Z. (eds.) *ACCV 2012, Part IV*. LNCS, vol. 7727, pp. 379–391. Springer, Heidelberg (2013)
20. Veeraraghavan, A., Raskar, R., Agrawal, A., Mohan, A., Tumblin, J.: Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing. In: *SIGGRAPH* (2007)
21. Webster, S., Dorrell, A.: Improved motion invariant imaging with time varying shutter functions. In: *Proc. of SPIE-IS&T Electronic Imaging*. SPIE, vol. 7876, pp. 787604–787604–8 (2011)
22. Whyte, O., Sivic, J., Zisserman, A., Ponce, J.: Non-uniform deblurring for shaken images. *Int'l Journal of Computer Vision* 98(2), 168–186 (2012)
23. Xu, L., Jia, J.: Two-phase kernel estimation for robust motion deblurring. In: *Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I*. LNCS, vol. 6311, pp. 157–170. Springer, Heidelberg (2010)