

Motion Invariance and Custom Blur from Lens Motion

Scott McCloskey, Kelly Muldoon, and Sharath Venkatesha
Honeywell ACS Labs
1985 Douglas Drive North, Golden Valley, MN, USA

{scott.mccloskey, kelly.muldoon, sharath.venkatesha}@honeywell.com

Abstract

We demonstrate that image stabilizing hardware included in many camera lenses can be used to implement motion invariance and custom blur effects. Motion invariance is intended to capture images where objects within a range of velocities appear defocused with the same point spread function, obviating the need for blur estimation in advance of de-blurring. We show that the necessary parabolic motion can be implemented with stabilizing lens motion, but that the range of velocities to which capture is invariant decreases with increasing exposure time. We also show that, when that range is expanded through increased lens displacement, lens motion becomes less repeatable. In addition to motion invariance, we demonstrate that stabilizing lens motion can be used to design custom defocus kernels for aesthetic purposes, and can replace lens accessories.

1. Introduction

In recent years, image stabilization has become a popular feature of Digital Single Lens Reflex (DSLR) cameras, with wide product offerings from the major camera manufacturers. The objective of image stabilization is to prevent motion blur arising from movement of a photographer's hand during image capture. Implementations vary by manufacturer, but the two different categories - lens-based stabilization and sensor shift stabilization - can both be thought of as shifting the relative position of the sensor to the camera's optical axis *without* inducing a change in the optical axis's orientation. At a high level, lens or sensor motion is induced to counter-act camera motion, and stabilizing hardware consists of two elements:

1. Motion sensors which detect horizontal and vertical motion relative to the sensor plane.
2. A floating lens element (or the image sensor) that is moved in a plane orthogonal to the optical axis, in order to compensate for camera motion.

Given the prevalence of such stabilization hardware in existing cameras, using these elements to implement computational photographic techniques has the potential to speed deployment of developed techniques. We have implemented hardware modifications and the firmware necessary to control the floating lens element of a Canon Image Stabilizer (IS) lens, in order to execute a pre-determined sequence of lens shifts. When a subject's velocity is known *a priori*, this can be used to induce a compensating motion in a stationary camera to prevent blur in the captured image. Unfortunately, since pre- or in-exposure estimation of a subject's velocity is non-trivial, such an approach requires significant computing resources and can't be performed with a camera's limited computing budget.

In order to obviate velocity- and blur-estimation, Levin *et al.* [9] have proposed motion invariant image capture for moving subjects. In order to demonstrate the concept, prototype cameras were developed based on whole camera rotation [9] and sensor shifting [3] using custom hardware. In both cases, image stabilization hardware has been mentioned as a preferred implementation of motion invariant image capture.

In this paper, we describe an implementation of motion invariant image capture using existing image stabilization hardware. We present results and analysis of this system, and point out behaviors of stabilization hardware that should be considered for motion invariance and other uses. We then demonstrate that lens motion can be used to customize blur shape, producing non-traditionally blurred images for aesthetic purposes.

2. Related Work

Motion blur, the motivating application of image stabilization, has long been studied in computer vision and image processing. The problem of blind deconvolution - that is, estimation and removal of blur from a single image - has the longest history. Numerous blind deconvolution methods [4] have been presented to mitigate the effects of blur in images. Recent work has concentrated on learning methods [5, 15] and the handling of spatially-varying blur [7] from

traditionally-acquired images.

There are also a number of approaches to capture additional information with a motion-blurred image in order to improve the performance of blur estimation. Ben-Ezra and Nayar [2] 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 then used to de-blur the high resolution video. Joshi *et al.* [6] added inertial sensors to a DSLR to improve estimation performance, and to enable spatially-varying blur.

Eschewing traditional image capture, others have advocated alternative capture techniques that simplify motion de-blurring in various ways. Raskar *et al.* [13] have advocated coded exposure using a fluttering shutter, through which linear, constant-velocity motion produces an invertible PSF that can be removed through deconvolution. As reviewed in Sec. 3, Levin *et al.* [9] have proposed capturing an image while the lens/sensor is translated along a line parallel to subject motion, producing a blur PSF that does not depend on the velocity of a moving subject and therefore obviating blur estimation. Cho *et al.* [3] propose a 2D extension of this method based on two exposures with orthogonal translations.

There are analogous approaches to these methods for handling defocus blur. Whereas coded temporal exposure was used for motion de-blurring, spatially coded apertures have been used by Levin *et al.* [8] and Veeraraghavan *et al.* [16] to simplify defocused image restoration. Whereas sensor motion orthogonal to the optical axis can be used to capture an image with invariant blur over different velocities, Nagahara *et al.* [12] have shown that sensor motion *along* the optical axis can be used to capture images with defocus blur that is invariant to camera/object distance.

With respect to the modification of an image’s point spread function, the objective is more aesthetic. Controlled defocus is often used by professional photographers to isolate subjects within a short depth of field, or within a spatial region (e.g., a center filter). In the recent literature, authors have proposed methods to artificially reduce an image’s depth of field, presuming that a camera’s widest available aperture is insufficiently large. Mohan *et al.* [11] reduce the depth of field optically with a custom-modified camera, by shifting the camera and/or lens during image capture. Bae and Durand [1] reduce depth of field digitally post-capture, by performing depth estimation and signal processing. As discussed in Section 6, our work differs from existing methods in that it produces depth invariant optical effects with existing hardware elements.

3. Motion Invariance

The objective of motion invariant image capture is to obviate PSF estimation before deblurring, by capturing an im-

age with a motion blur PSF that does not depend on the real-world velocity of a moving object. For an object with a particular velocity, of course, motion blur can be prevented by translating the lens in the same direction 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, objects with velocities in a certain range will have the motion of their projections stopped at some point, and these objects will have the same PSF.

Motion invariance can be implemented by translating any of the camera, sensor, or lens with constant acceleration. In [9], camera rotation was employed, and in subsequent work [3] sensor motion was employed. We use optical stabilization hardware to implement motion invariance by lens motion, as suggested by the authors in [9, 3]. 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 is applied. During exposure, the negative acceleration causes the lens to come to a stop and, at the end of the exposure duration, 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. It is shown in [9] that this parabolic motion is unique in its ability to cause invariant motion blur. In practice [3], the parabolic motion is approximated by several segments of constant velocity motion.

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

$$I_b = I * B + \eta, \quad (1)$$

where η represents noise. The blur PSF B caused by parabolic motion is of the form

$$B(i) = \begin{cases} \frac{1}{\sqrt{i}} & \text{for } i = 1, 2, \dots, N, \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where N is the length of the PSF, which is determined by calibration. The PSF B is normalized to have unit power.

There are three stated disadvantages of motion invariant photography [9]. The first is that stationary objects in the scene will be blurred in the image due to the motion of the lens and, while de-convolving B is numerically stable, de-blurring amplifies noise η everywhere. The second is that the invariance of B to velocity is only over a range of velocities, and that range is determined by the velocity of the lens element at the beginning and end of its motion. The

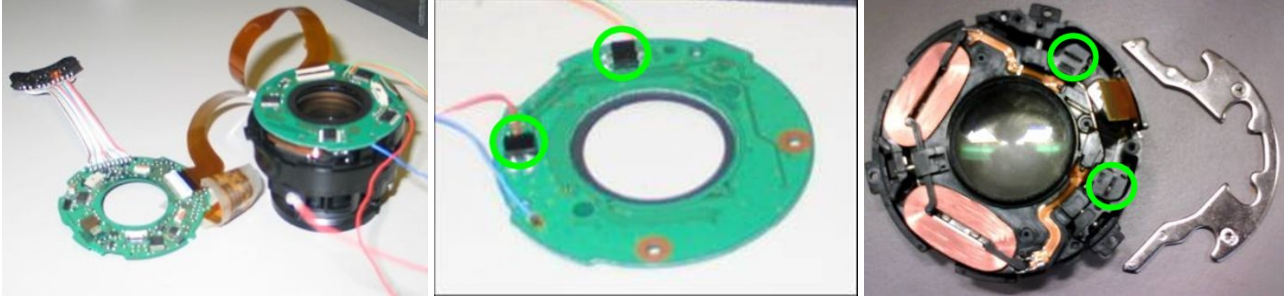


Figure 1. Stabilizing elements of the Canon EF70-200mm F/4L IS USM lens. (Left) We sever the flex cable connection between the position-sensing and motion compensating lens sub-systems. (Center) Lens motion is measured in the circled chips by the position of a laser spots relative to two orthogonal axes of motion. (Right) The floating element projects laser points (from circled slots at 1 and 4 o'clock), and moves the encased lens element using magnets (under red coils) and a yoke.

third is that the convolution model of eq. 1 does not hold at occluding contours, resulting in artifacts when both foreground and background have significant texture.

One disadvantage that has not been previously noted is that the range of velocities over which motion invariance is achieved *decreases* with increasing exposure time. The intuitive explanation can be understood with two constraints:

1. The maximum displacement of the lens element is limited by the size of the lens cavity.
2. The lens element must remain in motion throughout the exposure.

Simply put, if two motion invariant captures use the full range of lens motion, the one with longer exposure time must move slower, and thus will not be able to stop some of the higher velocity motion stopped by the shorter exposure. This is unfortunate, since longer exposure times are the motivating factor for much of the work on motion blur; short exposures with a traditional camera generally avoid motion blur without computational methods.

In order to quantify this dependence, let d be the maximum displacement of the lens element, and T be the exposure time of a motion invariant image. The parabolic motion $x(t) = X_0 + \frac{4d}{T^2} (t - \frac{T}{2})^2$ is approximated with N_s constant-velocity segments of equal duration $\frac{T}{N_s}$. The maximum velocity of the lens element happens at the beginning of the capture¹, where

$$\begin{aligned}
 v_{max} &= \frac{x(0) - x(\frac{T}{N_s})}{\frac{T}{N_s}} \\
 &= \frac{dN_s (1 - 4(\frac{1}{N_s} - \frac{1}{2})^2)}{T} \\
 &= \frac{4d}{T} \left(1 - \frac{1}{N_s}\right).
 \end{aligned} \tag{3}$$

From this, we can see that when exposure time T increases, the maximum velocity decreases. Given the additional de-

¹The velocity at the end of the capture has the same magnitude but is in the opposite direction.

pendence on N_s , it is tempting to think that increased T can be compensated by increasing the number N_s of segments. If T doubles (i.e. exposure time increases by one stop), the value of the term $(1 - \frac{1}{N_s})$ must double to compensate; it cannot, since the value of this term is limited to the range $[\frac{1}{2}, 1)$. Note that the value v_{max} is the velocity of the lens element, and the corresponding bounds on the range of *subject* velocities depends on the subject velocity and lens parameters.

4. Lens Motion Control

As mentioned in the introduction, image stabilizing hardware consists of two elements. In our Canon lens, those elements are: position sensors that record camera motion, and a stabilizing lens element that is moved in order to counter that motion. Fig. 1 shows these hardware elements extracted from the lens that we have modified. The standard mode of operation for image stabilization is a closed control loop where motion detected by sensors within the camera induces a compensating motion of the floating lens element.

The first step in the implementation is to disconnect the flex cable in order to prevent the stabilizing element from receiving signals from the lens processing board. We then add an external control signal interface connecting 12-bit ADCs to the position sensor, and an independent microcontroller to drive the IS lens. Control loops running on the independent microcontroller take commands from a host computer and create the desired lens motions.

Fig. 2 shows a block diagram of the IS lens control implementation that was used for the development and experimentation. We see that the IS lens position is determined by 2 optoelectronic position sensors shown in the upper left hand corner in green. These Position Sensitive Detectors (PSDs) are made up of a monolithic photo diode with a uniform resistance in one direction. This allows for high resolution position sensing with fast response time. Two

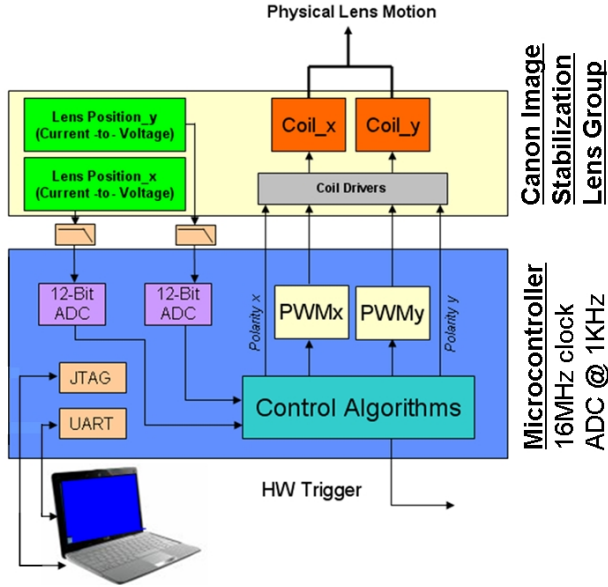


Figure 2. **Lens motion system overview.** Elements in the top (light yellow) box are existing hardware elements from the Canon lens. Elements in the lower (blue) box are executed in the microcontroller added in our modification. The green boxes correspond to the green-circled elements in Fig. 1, and orange boxes correspond to the red coils.

PSDs are used in the system one for sensing position in the X direction and one for Y. The voltage output of the sensors are tapped off the system, low-pass filtered and then are sampled by internal 12-bit ADCs on a custom board containing a 16 bit microcontroller. The sampling rate for the sensors is 1 KHz. The ADC samples are the inputs to a control algorithm which drives two Pulse Width Modulation (PWM) signals. The PWMs are fed into the coil drivers which provide a control current through the lens coils causing the physical motion of the IS lens in both X and Y directions. There are polarity signals associated with X and Y position coil drivers that control the direction (left/right and up/down). These polarities control the X and Y directions. A host computer is used for programming the microcontroller through a JTAG port. A UART is used to inspect data and to input desired locations into the system.

The implementation of the control algorithms are shown in Fig. 3. A classic PID^2 controller is applied to both X and Y positions in the system. The Y controller is different from X in that it has an offset component to compensate for the effects of gravity. Desired lens positions are fed to the X and Y controllers and the system runs until the position errors approach zero. Because we use a 12 bit ADC, the lens has 4096-by-4096 discrete positions over a range of approximately 3.5mm-by-3.5mm. In early experimentation, we found that movement between points in the cen-

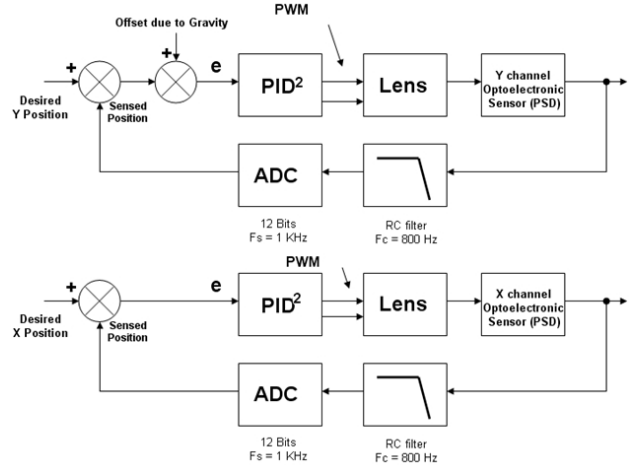


Figure 3. Control loops for lens X and Y position.

tral 3096-by-3096 region was relatively uniform, whereas movement between points in the periphery was not. As a result, we restrict our motion to the central region, and refer to displacements d in terms of the number of lens positions moved.

There are two types of commands used to approximate parabolic motion: one to move the lens element immediately to its starting position, and N_s commands to move the lens element to subsequent positions, linearly interpolating its position over a given time window. The camera's exposure is triggered by the embedded software when the first command is issued. The lens is connected to a monochrome Redlake camera with a 4872-by-3248 pixel sensor.

Because motion of the lens is implemented using a real-time control algorithm, certain artifacts are to be expected. Because the lens element has inertia while in motion, and because that motion must have constant velocity, the lens will overshoot its destination. In addition, since the lens is effected by gravity, vertical motions will have different accelerations in the up and down directions. When parabolic motion is approximated using constant-velocity segments, these transient errors will occur at segment boundaries. In order to illustrate these artifacts, we have captured images of a point light source (an LED) moving with constant horizontal velocity v_x as seen with different lens motions. In this setup, the horizontal position is a function of time t and starting position X_0 as $x(t) = X_0 + tv_x$. Fig. 4 (left) shows the motion of the light as viewed from a lens translating vertically with constant velocity v_y , thus vertical position $y(t) = Y_0 + tv_y$ relative to starting position Y_0 . In such an image, the LED should translate with constant velocity along a line $x = X_0 + \frac{v_x}{v_y}(y - Y_0)$, but transient errors from this trajectory are observed (see inset).

Fig. 4 (right) shows the same LED motion as viewed

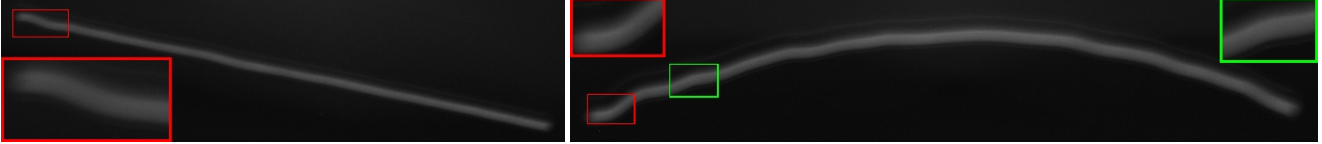


Figure 4. Point trajectories for motion calibration. Left image shows a point light source translating along a horizontal line while the lens translates along a vertical line (both with constant velocity). Ringing can be observed near the start of motion in the image’s upper left. Right image shows the point light source translating with constant velocity along a horizontal line while the lens undergoes parabolic motion in y/t , as approximated by $N_s = 8$ constant-time linear segments. The ringing at the beginning of each such segment produces the wobbles in the otherwise parabolic trajectory.

from a lens capturing an image with motion invariance in the vertical direction over exposure time $T = 300ms$. We have parabolic motion in y/t , with $y(t) = Y_0 + \frac{4d}{T^2}(t - \frac{T}{2})$. Thus, the point’s trajectory in the image should be parabolic, and deviations from this can be observed.

5. Experiments: Motion Invariance

In this section, we describe several experiments to illustrate the trade-offs of motion invariance, based largely on eq. 3. In particular, we will illustrate that

1. When exposure time T increases, the range of object velocities over which motion invariance can be achieved is reduced, as discussed in Sec. 3.
2. When lens displacement d increases, the range of velocities is increased at the expense of reconstructed image quality of stationary objects.
3. The reliability of repeatable lens motion depends on the lens displacement d , but not the number N_s of constant-velocity segments used to approximate parabolic motion.

We first demonstrate that, though Fig. 4 illustrates that our lens motion system does not perfectly implement parabolic motion, it provides a good approximation of the desired motion. We demonstrate this by inspecting images of a stationary scene through both a stationary lens and one undergoing parabolic motion with $d = 250$. In this example we approximate parabolic motion with $N_s = 14$ intervals of constant velocity, and we calibrate the motion invariant PSF off-line using this image pair. Analytically, we would expect that the difference between the two images is described by convolution with a kernel of the form of eq. 2. Fig. 6 shows captured image intensities along a scan-line of an image containing a bright bar feature on a dark background, using a stationary lens (blue line) and parabolic motion (green line). Using a PSF of eq. 2 with $N = 44$ (the best fit value), we synthetically blur the image from the stationary lens, and plot the resulting simulated blur scan-line

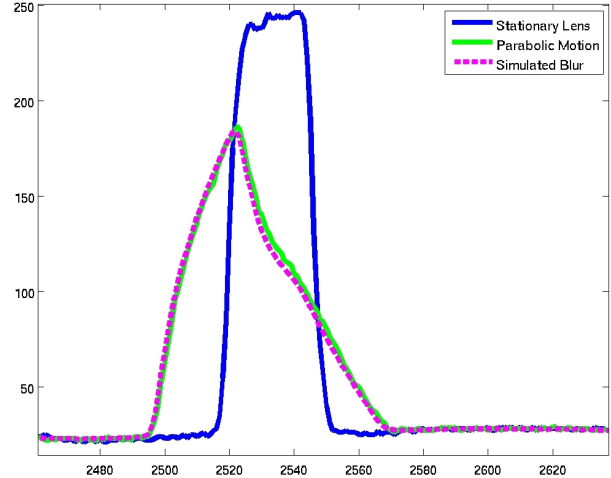


Figure 6. Intensity profiles of a bar image feature in scan lines of images with stationary lens (blue), parabolic lens motion (green), and synthetic parabolic motion applied to the stationary image (magenta), indicating good agreement between the motion model and actual images.

(dashed magenta line)². There is good agreement between the simulated blur and lens motion blur. Fig. 5 shows the captured motion invariant image (left), the estimated latent image found using this PSF and Lucy-Richardson deconvolution [14, 10] (center), and a reference image taken without lens motion (right). Notwithstanding faint artifacts at the right edge of the head, reconstructed image quality is good despite deviations from true parabolic motion.

5.1. Changing Exposure Time T

To illustrate that the range of invariant velocities is reduced when exposure time T is increased, we have captured and de-blurred images of an object moving with velocity $\approx .3 \frac{m}{s}$ for $T = 300ms, 400ms, \text{ and } 500ms$. In each case, the displacement $d = 1000$ positions, and parabolic motion is approximated with $N_s = 20$ constant-velocity segments. Because the displacement is the same in all three

²Note that the lens motion induces a spatial shift in addition to the PSF. We remove this shift with manual feature identification.

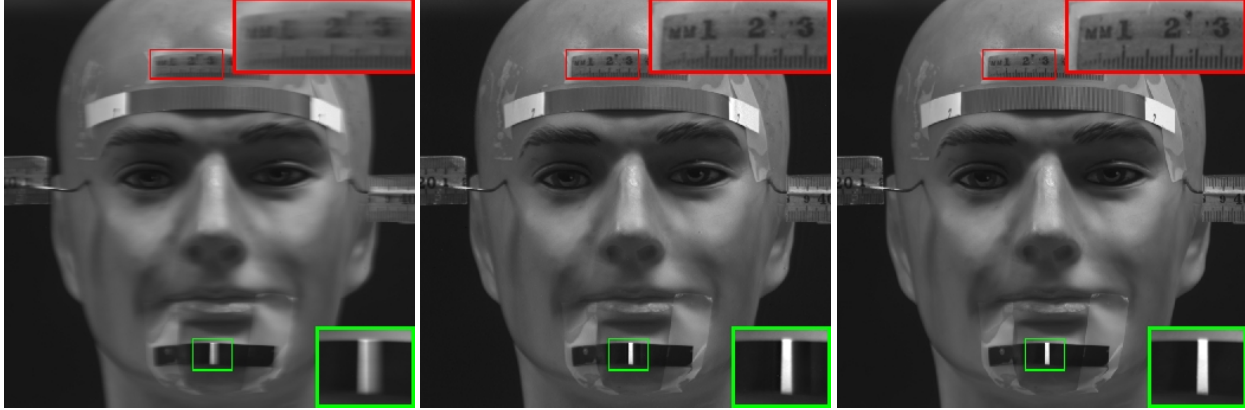


Figure 5. (Left) image of a stationary object through parabolic lens motion. (Center) de-blurred version. (Right) reference image of object without lens motion.

cases, the PSF observed at stationary objects is the same, with $N = 178$. Fig. 7 shows the motion invariant capture and de-blurred image for the three exposure times of a head translating on a motion stage. In the $T = 300ms$ and $T = 400ms$ cases, the PSF on the moving head is approximately the same size as the PSF on the stationary white bars in the corners. That is to say that the motion invariance in these two images covers the velocity of the moving head and, though there are artifacts near the occluding contour, reconstructed image quality is reasonably good. In the $T = 500ms$ case, on the other hand, the PSF on the moving head is clearly larger than the PSF on the stationary background (observe the width of the intensity ramp between neck and background), and the reconstruction fails because the head is not within the range of motion invariance.

5.2. Changing Lens Displacement d

Given that increasing exposure time limits the range of motion invariance, and that increasing the number N_s of segments can't totally counter this effect, eq. 3 suggests that displacement d should be maximized in order to cover the widest range of velocities. This is problematic, though, since the size N of the PSF is proportional to d . Fig. 8 (left) shows the Modulation Transfer Functions (MTF) of three different motion invariant blur kernels over a range of N . We see that, for higher N , contrast in the captured image is more muted. As a result, greater amplification is needed for these larger N , resulting in worse noise and stronger spatial artifacts in the reconstructed images. Fig. 8 also shows de-blurred images for $d = 500, 1000, 1500$ ($N = 91, 176, 267$ pixels, respectively), in which these effects can be observed.

5.3. Motion Repeatability

Motion invariant imaging implicitly assumes that the motion of the camera/sensor/lens used to induce the in-

d	RMSE	N_s	RMSE
500	0.8943	12	1.2021
1000	0.9451	16	1.0922
1500	1.1160	20	1.1245
2000	1.1414	24	1.0801
2500	1.2553	28	1.1106
		32	1.1648

Table 1. Assessing the impact of parameters d and N_s on motion repeatability.

variant PSF is repeatable. Because the invariant PSF is characterized once and used to de-blur subsequently captured images, it is important that the motion reliably produce the same PSF. In our early experiments, we noticed that this condition does not always hold as strictly as we would like, and found anecdotal evidence that the motion variability depended on certain parameters. In order to assess repeatability as a function of these parameters, we carried out an experiment where we captured 10 images under parabolic motion for each combination of $d \in \{500, 1000, 1500, 2000, 2500\}$ and $N_s \in \{13, 17, 21, 25, 29, 33\}$, for a total of 300 images. All images were acquired with $T = 300ms$. For each combination of d and N_s , we find the PSF B of the form of eq. 2 that best fits the 10 captured images, given a reference sharp image. We then measure the root mean squared error (RMSE) of the feature used in Fig. 6 relative to its synthetically-blurred appearance given B and the sharp image. Table 1 shows the RMSE as a function of both d and N_s , from which we can see that increasing d reduces the repeatability of lens motion. This makes sense, as larger values of d with fixed T force the lens to move over greater distances in a given time, increasing overshoots. There is no consistent effect of N_s on motion repeatability.

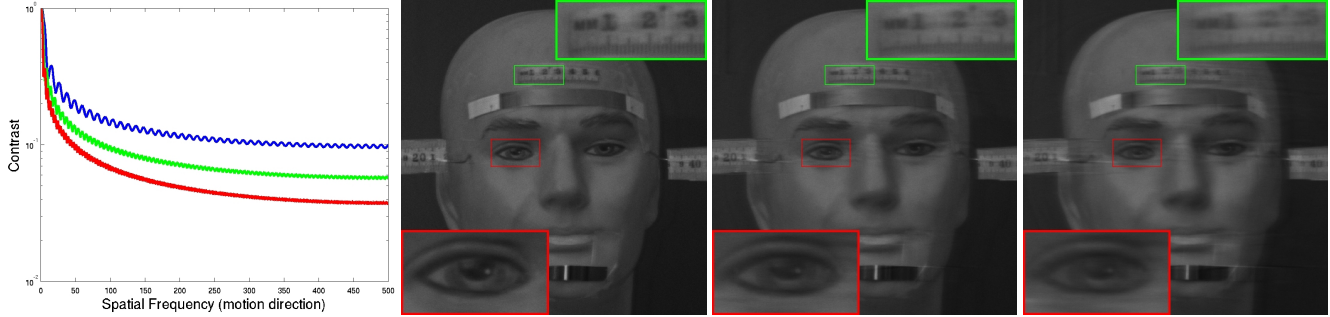


Figure 8. Performance of motion invariance with increasing d . Left plot shows the MTFs of motion invariant PSF for $d = 500$ (blue), 1000 (green), and 1500 (red). Images show de-blurred stationary head for $d = 500, 1000, 1500$ (left to right), with reducing contrast (notably in green inset).

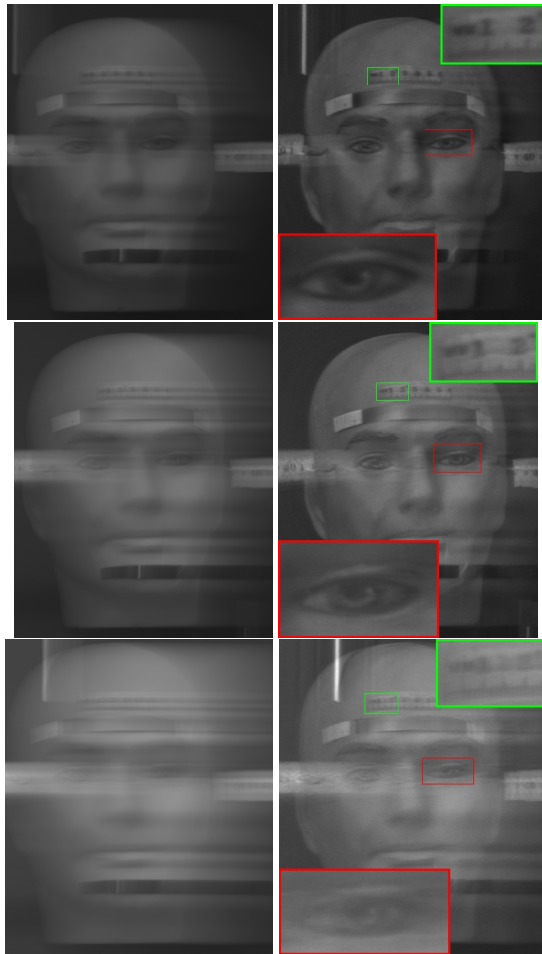


Figure 7. Performance of motion invariance with increasing T . Rows, from top to bottom, have $T = 300, 400, 500$ ms. (Left column) motion invariant image of head moving with velocity $\approx .3 \frac{m}{s}$. (Right column) image of left column, deblurred with motion-invariant PSF.

6. Custom Blur

There are several special effects filters that produce defocus PSFs that advance various aesthetic aims. In this section, we demonstrate that a lens’s stabilizing element can be used to produce similar effects and custom blur without the need for additional equipment.

Fig. 9 shows how, by moving the lens’s stabilizing element during capture, we can emulate a cross screen filter. In this case, the lens element traces out a cross shape during a small portion (10%) of the image’s integration time, and remains stationary for the remaining 90%. Near image highlights, this has the effect of creating streaks, but other objects are sharply rendered because the lens is mostly stationary. Fig. 9 (bottom) shows an image taken through an optical cross screen filter, which produces a similar effect under the same illumination. Compared to the optical filter, our lens motion implementation produces a sharper rendering of mid-tone objects (see green insets). Whereas separate lens accessories are needed to create different patterns using physical filters, the use of lens motion sequences adds this as a feature at no additional cost.

An important distinction between previous work [1, 11] and the effects generated by motion of the stabilizing element is that the blur does not scale with object distance. Because the lens motion does not induce a change in orientation of the optical axis, there is no motion parallax. As a result, motion of the stabilizing element can’t be used to implement a synthetic aperture, nor can it be used to distinguish between objects at different depths.

7. Conclusions and Limitations

We have demonstrated that image stabilization hardware commonly found in digital single lens reflex (DSLR) cameras can be used to emulate optical filters and custom blur without the use of lens accessories. We have also implemented motion invariant image capture using the same control software, and demonstrated good quality deblurring re-

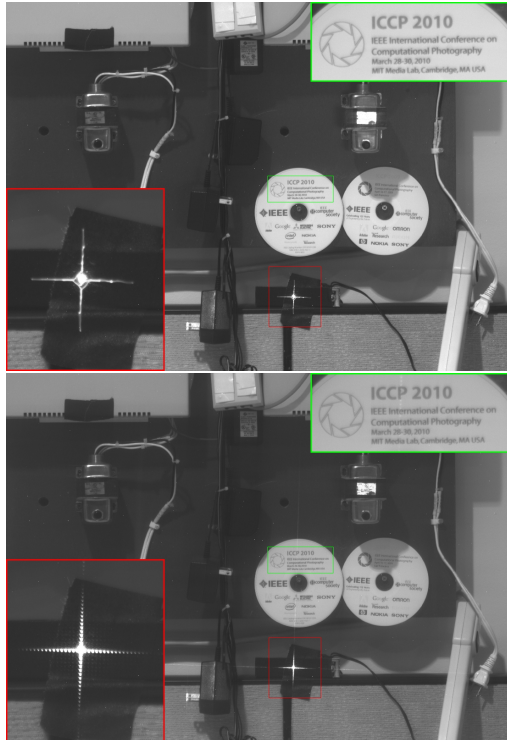


Figure 9. Implementing a cross screen filter with lens motion. (Top) Image with +-shaped lens motion for 10% of the exposure time. (Bottom) Image acquired with a stationary lens, through a PROMASTER[®] Cross Screen 4 Filter 67mm. In both images, insets show a region around a highlight (red box) and text (green box) which has better contrast in the lens motion image.

sults in certain cases. We have pointed out that, since the lens element travels over a fixed range during the exposure time, the breadth of the resulting motion invariance decreases with increased exposure time. We have also illustrated several trade-offs in the use of motion invariant image capture.

By far, the greatest limitation of our current system is its size and resulting lack of portability. The camera is controlled by a host computer, and the associated serial connection and electronics are relatively large. Though power usage is modest, the lens is currently connected to a power supply providing 5V DC. The camera is mounted on a tripod and can't easily be moved so, as a result, all of our experiments are limited to the same laboratory environment. Our lens modifications have also disabled the lens's aperture, which is fixed at F/4 and does not provide much depth of field. We plan to address these shortcomings in an upcoming portable revision which will run on a battery and can trigger a pre-defined sequence using the hot shoe trigger provided by existing camera hardware.

Acknowledgements

PROMASTER[®] is a trademark of Photographic Research Organization. All brand names and trademarks used herein are for descriptive purposes only and are the property of their respective owners.

References

- [1] S. Bae and F. Durand. Defocus magnification. In *Eurographics*, 2007.
- [2] M. Ben-Ezra and S. K. Nayar. Motion deblurring using hybrid imaging. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 657–664, 2003.
- [3] T. S. Cho, A. Levin, F. Durand, and W. T. Freeman. Motion blur removal with orthogonal parabolic exposures. In *International Conf. on Computational Photography*, 2010.
- [4] S. Haykin. *Blind Deconvolution*. Prentice-Hall, 1994.
- [5] J. Jia. Single image motion deblurring using transparency. In *IEEE Conf. on Computer Vision and Pattern Recognition*, pages 1–8, 2007.
- [6] N. Joshi, S. B. Kang, C. L. Zitnick, and R. Szeliski. Image deblurring using inertial measurement sensors. In *ACM SIGGRAPH*, pages 1–9, 2010.
- [7] A. Levin. Blind motion deblurring using image statistics. In *NIPS*, pages 841–848, 2006.
- [8] A. Levin, R. Fergus, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. In *ACM SIGGRAPH*, 2007.
- [9] A. Levin, P. Sand, T. S. Cho, F. Durand, and W. T. Freeman. Motion-invariant photography. In *ACM SIGGRAPH*, 2008.
- [10] L. B. Lucy. An iterative technique for the rectification of observed distributions. *Astron. J.*, 79:745+, June 1974.
- [11] A. Mohan, D. Lanman, S. Hiura, and R. Raskar. Image destablization: Programmable defocus using lens and sensor motion. In *International Conf. on Computational Photography*, 2009.
- [12] H. Nagahara, S. Kuthirummal, C. Zhou, and S. Nayar. Flexible Depth of Field Photography. In *European Conf. on Computer Vision*, Oct 2008.
- [13] R. Raskar, A. Agrawal, and J. Tumblin. Coded exposure photography: motion deblurring using fluttered shutter. In *ACM SIGGRAPH*, 2006.
- [14] H. W. Richardson. Bayesian-based iterative method of image restoration. *Journal of the Optical Society of America*, 62(1):55–59, January 1972.
- [15] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. In *ACM SIGGRAPH*, 2008.
- [16] A. Veeraraghavan, R. Raskar, A. Agrawal, A. Mohan, and J. Tumblin. Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing. In *ACM SIGGRAPH*, 2007.