# Motion-Invariant Photography

Anat Levin Peter Sand Taeg Sang Cho Frédo Durand William T. Freeman

Massachusetts Institute of Technology, Computer Science and Artificial Intelligence Laboratory

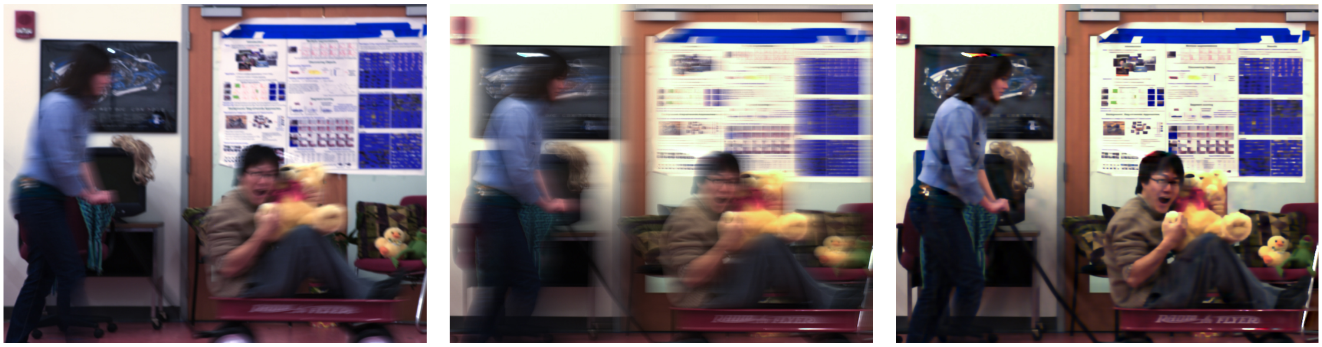


**Figure 1: Left:** *Blurred motion captured by a static camera.* **Center:** *The same scene captured by a camera with a specially designed motion that causes both the static and dynamic regions to blur identically.* **Right:** *The blur from the center image can be removed independently of motion via deconvolution of the entire image with a single known point spread function.*

## Abstract

Object motion during camera exposure often leads to noticeable blurring artifacts. Proper elimination of this blur is challenging because the blur kernel is unknown, varies over the image as a function of object velocity, and destroys high frequencies. In the case of motions along a 1D direction (e.g. horizontal) we show that these challenges can be addressed using a camera that moves during the exposure. Through the analysis of motion blur as space-time integration, we show that a parabolic integration (corresponding to constant sensor acceleration) leads to motion blur that is invariant to object velocity. Thus, a single deconvolution kernel can be used to remove blur and create sharp images of scenes with objects moving at different speeds, without requiring any segmentation and without knowledge of the object speeds. Apart from motion invariance, we prove that the derived parabolic motion preserves image frequency content nearly optimally. That is, while static objects are degraded relative to their image from a static camera, a reliable reconstruction of all moving objects within a given velocities range is made possible. We have built a prototype camera and present successful deblurring results over a wide variety of human motions.

**Keywords:** Computational Photography, Motion Deblurring, Coded Imaging, Space-time

## 1 Introduction

Motion blur often limits the quality of photographs and can be caused by either the camera shake or the movement of objects in the scene. Modern cameras address the former case with image stabilization, where motion sensors control mechanical actuators that shift the sensor or lens element in real time during the exposure to compensate for motion of the camera, e.g. [Canon 2003]. The use of image stabilization enables sharp hand-held photographs of still subjects at much slower shutter speeds, thereby reducing image degradation. Unfortunately, image stabilization only addresses camera motion and cannot help with moving objects.

One option is to remove the blur after the shot was taken using deconvolution. However, this raises several challenges. First, the typical motion-blur kernel is a line segment in the direction of motion, which corresponds to a box filter. This kernel severely attenuates high spatial frequencies and deconvolution quickly becomes ill-conditioned as kernel size increases. Second, the length and direction of the blur kernel both depend on object motion and are therefore unknown and must be estimated. Finally, motion blur usually varies over the image since different objects or regions can have different motions, and segmentation must be used to separate image regions with different motions. These two later challenges lead most existing motion deblurring strategies to rely on multiple input images [Bascle et al. 1996; Rav-Acha and Peleg 2005; Shi and Zheng 2005; Bar et al. 2007; Ben-Ezra and Nayar 2004; Yuan et al. 2007]. Some recent methods attempt to remove blur from a single input image using natural image statistics [Fergus et al. 2006; Levin 2006]. While these techniques enable impressive results, they have their own limitations. Raskar et al. proposed a hardware approach that addresses the first challenge [2006]. A fluttered shutter modifies the line segment kernel to achieve a more broad-band frequency response, which allows for dramatically improved deconvolution results. While their solution blocks half of the light, the improved kernel is well worth the tradeoff. However, their method still requires the precise knowledge of motion segmentation boundaries and object velocities, an unsolved problem.

In this paper, we show that if the motion is restricted to a 1D set of velocities, such as in a horizontal motion (as is the case with many real world objects like cars or walking people), we can address all three challenges. Using camera hardware similar to that used for image stabilization, we can make the point-spread function (PSF) invariant to motion and easy to invert. For this, we introduce

**ACM Reference Format**
Levin, A., Sand, P., Cho, T., Durand, F., Freeman, W. 2008. Motion–Invariant Photography. *ACM Trans. Graph. 27*, 3, Article 71 (August 2008), 9 pages. DOI = 10.1145/1360612.1360670 http://doi.acm.org/10.1145/1360612.1360670.

a specific camera movement during exposure. This movement is designed so that the compound motion of the camera and objects at any depth results in nearly the same easy-to-invert PSF. Since the entire scene is blurred with a nearly identical PSF, the blur can be removed via deconvolution, without segmenting moving objects and without estimating their velocities. In practice, we find that motions even somewhat away from the selected 1D orientation are deblurred as well.

Our approach is inspired by wavefront coding [Cathey and Dowski 1995] where depth of field is improved by modifying a lens to make the defocus blur invariant to depth and easy to invert. While their work deals with wave optics and depth of field, we consider geometric ray optics and remove 1D motion blur.

By analyzing motion blur as integration in a space time volume over curves resulting from camera and object motion, we prove that the only integration curve that results in a motion-invariant PSF is a parabola. This corresponds to constant 1D acceleration of the senor, first going fast in one direction, progressively slowing down until it stops, and then picking up speed in the other direction. As a result for any object velocity within a range, there is always one moment during exposure when the camera is perfectly tracking the object. While the camera motion is along a straight line, we will call it "parabolic motion" because of the parabolic relationship between position and time.

In addition to its invariance to object speed, our PSF preserves more high frequencies for moving objects than a normal exposure. This however comes at the cost of slightly degraded performance for static objects. In fact, we show that even if object motions could be estimated perfectly, the type of PSFs resulting from the parabolic camera motion is near-optimal for a stable deconvolution over a range of possible object speeds. This optimality is in the sense of minimizing the degradation of the reconstructed image for a range of potential object velocities. In a nutshell, we show that there is a fixed bandwidth budget for imaging objects at different velocities. A static camera spends most of this budget to achieve high-quality images of static objects, at the cost of severe blur for moving objects. Our design distributes this budget more uniformly, improving the reconstruction of all motion velocities, at the price of a slightly worse reconstruction of the static parts.

There are three basic options for implementing this camera movement: a translation of the full camera, a rotation of the camera, or a translation of the sensor or lens element. For a commercial product, the latter may be the best solution, and could be achieved with the existing hardware used for stabilization. However, if the field-of-view is not too wide, camera rotation is a good approximation to sensor translation and is easier to implement as a prototype. We demonstrate a prototype using camera rotation and show 1D speed-invariant deconvolution results for a range of 1D and even for some 2D motions.

## 2 Motion invariant integration

In order to derive a motion–invariant photography scheme, we characterize motion blur as an integration in space–time. We show that the effect of object motion can be characterized by a shear. We also show that a camera parabolic movement is invariant to shear in space–time and permits the removal of motion blur.

**Space–time analysis of motion:** The set of 2D images falling on a detector over time forms a 3D *xyt*–space–time volume of image intensities. We consider a 2D *xt*-slice through that 3D space-time volume. Each row in this slice represents a horizontal 1D image, as captured by a static pinhole camera with an infinitesimal exposure time.

For sufficiently small exposure, a first order approximation to object motion is sufficient, and motion paths are assumed to be linear. In this case scene points trace straight lines in the *xt*–slice and the slopes of these lines are a function of the object velocity and depth. Figure 2(a–d) demonstrate an *xt*–slice - the green object is static, which means that it is invariant to time and results in vertical lines in space time. The blue and red objects are moving in opposite directions, resulting in oblique lines. The slope of these lines correspond to image-space object velocity.

Formally, the space-time function of an object moving at constant image–space velocity $s$ is related to that of a static object by a shear, since kinematics gives:

$$x(t) = x(0) + st \tag{1}$$

**Camera motion and integration:** The *xt*–plane represents the scene relative to a static camera. If the sensor is translating, the image recorded at time instance $t$ is a shifted version of row $t$ in the *xt*–plane. Thus, when the scene is captured by a translating sensor over a finite exposure time, the recorded intensities (the blurred image) are the average of all shifted images seen during the exposure length, at all infinitesimal time instances. That is, the sensor elements integrate light over curves in the *xt*–plane. The simplest case is a static camera, which integrate light over straight vertical lines (Figure 2a). A sensor translating with constant velocity leads to slanted integration lines (in Figure 2b the camera tracks the red object motion). The integration curve of a uniformly-translating sensor is a sheared version of that of the static sensor, following the same principle as the object-motion shear (but in the opposite direction). More complex sensor motion leads to more general curves. For example Figure 2c presents a parabola, obtained with translating sensor undergoing a parabolic displacement. If a shutter is fluttered during exposure as in [Raskar et al. 2006], the integration curve is discontinuous (Figure 2d).

Since we only translate the sensor, the integration curves we consider are spatially shift invariant. We denote by $L(x,t)$ the intensity of light rays in the *xt*–slice, $I(x)$ the intensity of the captured image, $f(t)$ the integration curve, and $[-T,T]$ an integration interval of length $2T$. The captured image can be modeled as:

$$I(x) = \int_{-T}^{T} L(f(t)+x,t)dt \tag{2}$$

**The Point Spread Function:** Let us denote by $I^0(x)$ an ideal instantaneous pinhole image $I^0(x) = L(x,0)$. The movement of the camera creates motion blur. For a static object, we can model the camera blur as a convolution of $I^0$ with a Point Spread Function (PSF) $\phi_0$: $I = \phi_0 \otimes I^0$. In this case, $\phi_0$ is simply the projection of $f$ along the time direction onto the spatial line:

$$\phi_0(x) = \int_t \delta_{f(t)=x} dt \tag{3}$$

where $\delta$ is a Dirac.

We now consider objects moving at speed $s$ and seek to derive an equivalent Point Spread Function $\phi_s$. We can reduce this to the static case by applying a change of frame that "stabilizes" this motion, that is, that makes the space-time volume of this object vertical. We simply apply the inverse of the shear in eq. 1, which is the shear in the opposite direction, and the sheared curve can be expressed as:

$$f_s(t) = f(t) - st \tag{4}$$

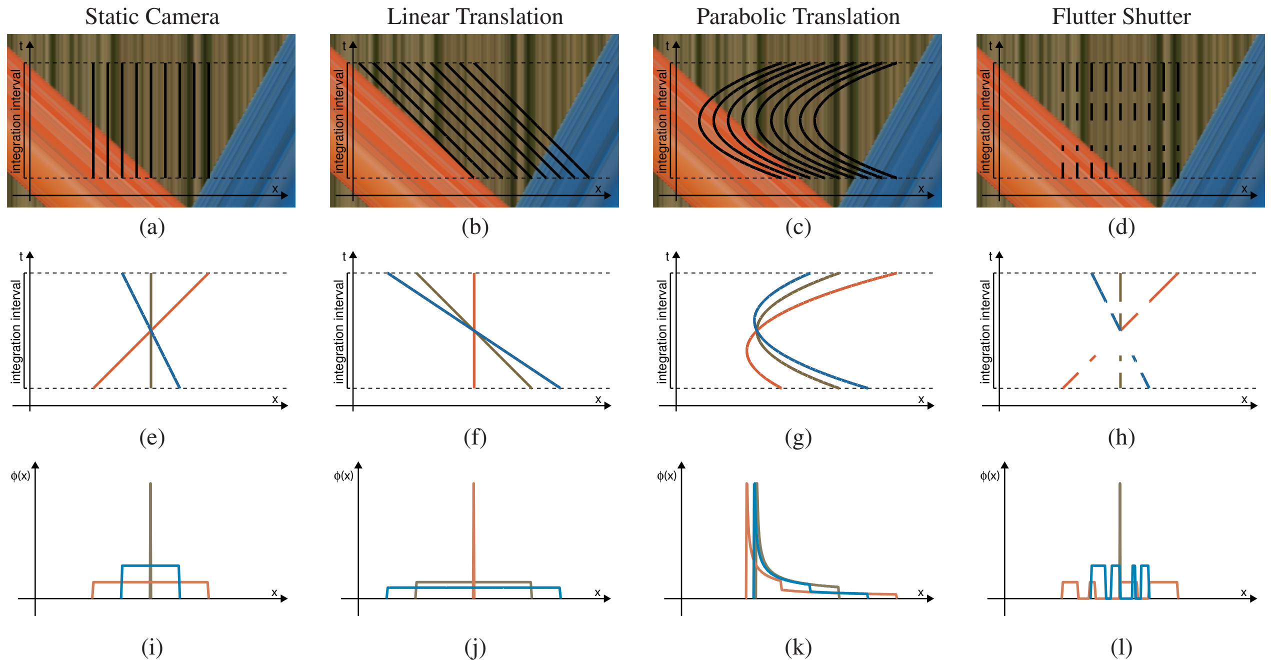


**Figure 2:** *From integration curves to Point Spread Functions.* **Top row***: xt-slice and integration curves resulting from different camera motions.* **Middle row***: integration curves sheared to account for object slope (curve colors match corresponding object).* **Bottom row***: projected PSFs corresponding to different object velocities.*

Sheared curves are illustrated in Figure 2(e-h). As a result, the PSF $\phi_s$ for a moving object is the vertical projection[1] of the sheared integration curve $f_s$. Equivalently, it is the oblique projection of $f$ along the direction of motion.

Figs 2(i-l) presents the PSF of the 3 different objects, for each of the curves in figs 2(a-d). For example, if the integration curve is a straight line the PSF is a delta function for objects whose slope matches the integration slope, and a box filter for other slopes. The box width is a function of the deviation between the object slope and integration slope. The analytic way to derive this projection is to note that the vertical projection is the “amount of time” the curve $f$ spent at the spatial point $x$ - the slope of the inverse curve. That is, if $g_s = f_s^{-1}$ is the inverse curve, the PSF (the vertical projection) satisfies: $\phi_s(x) = g_s'(x)$.

**Shear invariant curves:** Our goal is to derive a camera motion rule that leads to a velocity invariant PSF. Intuitively, we can achieve such effect if we devote a portion of the exposure time tracing each possible velocity, and we spend an equal amount of time tracing each velocity. The derivative of an integration curve representing such motion should be linear and therefore, the curve itself should be a *parabola*. This intuition provides a hint that parabolas are good candidates, but we still need to prove that they yield motion-invariant PSFs.

We have seen that PSFs corresponding to different velocities are obtained from sheared version of the sensor integration curve. Consider a parabola curve of the form: $f(t) = a_0 t^2$ (Figure 2(c)). The resulting PSF behaves like $1/\sqrt{a_0 x}$ (Figure 2(k)). We note that a sheared parabola is also a parabola with the same scale, only the center is shifting

$$f_s(t) = f(t) - st = a_0(t - \frac{s}{2a_0})^2 - \frac{s^2}{4a_0} \tag{5}$$

Thus, the projections $\phi_s$ are also *identical up to a spatial shift*. The important practical application of this property is that if the camera is moved during integration along a parabola curve we can deconvolve all captured images $I$ with the same PSF, *without segmenting the moving objects in the image, and without estimating their velocity or depth*. The small spatial shift of the PSF leads to a small spatial shift of the deconvolved image, but such a shift is uncritical as it does not translate to visual artifacts. This simply means that the position of moving objects corresponds to different time instants within the exposure. The time shift for a given velocity corresponds to the time where the sensor is perfectly tracking this velocity.

We note that the above invariance is almost true, but involves two source of approximation. The first approximation has to do with the fact that the invariant convolution model is wrong at the motion layer boundaries. However, this has not been a major practical issue in our experiments and is visible only when both foreground and background have high-contrast textures. The second approximation results from the fact that a parabola is perfectly shear invariant only if an infinite integration time is used. For any finite time interval, the accurate projection is equal to:

$$\phi_s(x) = \begin{cases} \frac{1}{\sqrt{a_0(x+\frac{s^2}{4a_0})}} & 0 \le x + \frac{s^2}{4a_0} \le a_0(T - \frac{s}{2a_0})^2 \\ \frac{1}{2\sqrt{a_0(x+\frac{s^2}{4a_0})}} & a_0(T - \frac{s}{2a_0})^2 \le x + \frac{s^2}{4a_0} \le a_0(T + \frac{s}{2a_0})^2 \\ 0 & otherwise \end{cases} \tag{6}$$

Thus, for a finite integration interval, the tails of the PSF do depend on the slope $s$. This change in the tail clipping can also be observed in the projected PSFs in Figure 2(k). For a sufficiently bounded range of slopes the tail clipping happens far enough from the center and its effect could be neglected. However, equation 6 also highlights the tradeoffs in the exact parabola scaling $a_0$. Obviously, smaller $a_0$ values lead to sharper PSFs. On the other hand, for a given integration interval $[-T, T]$, the tail clipping starts at $\sqrt{a}(T - \frac{s}{2a_0})$; thus reducing $a_0$ also reduces the range of $s$ values for which the tail clipping is actually negligible.

[1] Note that the shear does not affect the projection integral measure because the determinant of its Jacobian is 1.

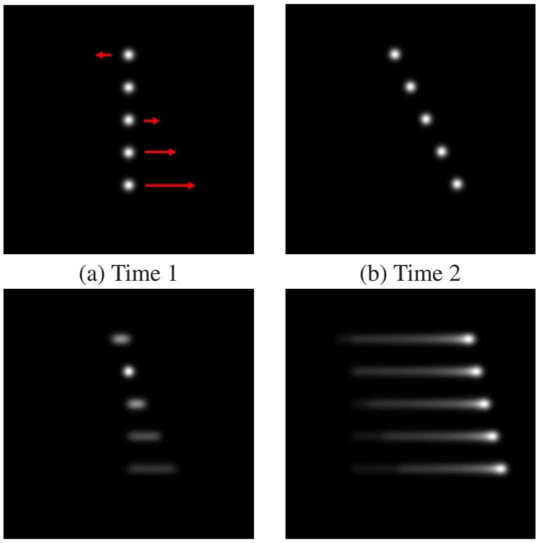


**Figure 3:** *Simulation of photographs of 5 dots moving over a range of speeds and directions. (a) Dot positions at beginning of photograph exposure, showing dot velocities. (b) Dot positions at end of exposure. (c) Photograph from stationary camera. Each differently moving dot produces a different blur. (d) Photograph from a parabolic camera. Note that the blur kernel is nearly identical for each of the differently moving dots, allowing for deblurring by a spatially invariant deconvolution.*

**Simulation:** To simulate the blur from a camera moving in a parabolic displacement in space-time (constant 1D acceleration), we projected synthetic scenes and summed displaced images over the camera integration time. Figure 3(a) shows 5 dots at the initial time, with their motion vectors, and Figure 3b shows their final configuration at the end of the camera integration period. Figure 3c is the photograph obtained with a static camera, revealing the different impulse response for each of the 5 different dot speeds. Figure 3(d) shows the image that would be recorded from the camera undergoing a parabolic displacement. Note that now each of the dots creates virtually the same impulse response, regardless of its speed of translation (there is a small speed-dependent spatial offset to each kernel). This allows an unblurred image to be recovered from spatially invariant deconvolution.

## 3 Uniqueness and optimality

In this section, we derive uniqueness and optimality criteria. The reader interested in practical implementation can directly read section 4.

### 3.1 Uniqueness

We have seen that a parabola integration curve leads to a shear invariant PSF. Moreover, we can prove that the parabola is the *only* shear invariant curve.

**Theorem 1** *If for every s there exist an a and b such that*

$$f(t) - st = f(t+a) + b \tag{7}$$

*then the curve $f(t)$ must be a parabola.*

*Proof:* Differentiating equation 7 and rearranging, we find

$$\frac{\partial f(t+a)}{\partial t} - \frac{\partial f(t)}{\partial t} = s$$

which means that $\frac{\partial f(t)}{\partial t}$ is a linear function of $t$, and thus, $f(t)$ is a parabola. □

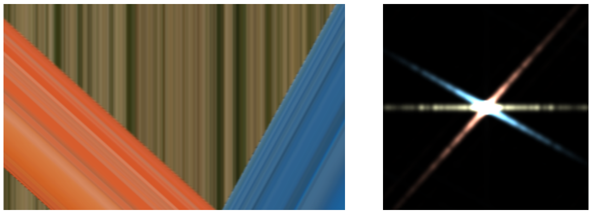


**Figure 4:** *The velocity range of an xt–slice and the corresponding frequency content.*

### 3.2 Upper Bound

We have seen that the parabola is the only shear invariant curve, and therefore parabolic displacement is the only camera movement that yields a PSF invariant to motion. Here we show that, in the case of 1D motions, this curve approaches optimality even if we drop the motion-invariant requirement. That is, suppose that we could perfectly segment the image into different motions, and accurately know the PSF of each segment. For good image restoration, we want the PSFs corresponding to different velocities or slopes to be as easy to invert as possible. In the Fourier domain, this means that low Fourier coefficients must be avoided. We show that, for a given range of velocities, the ability to maximize the Fourier spectrum is bounded and that our parabolic integration approaches the optimal spectrum bound.

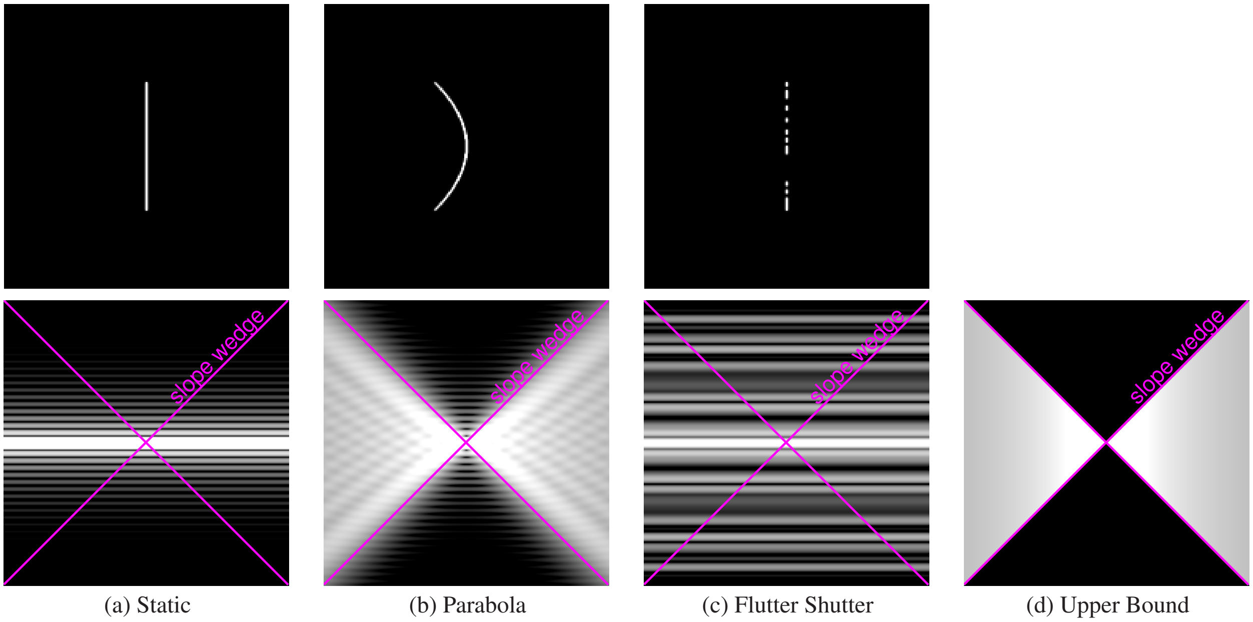
At a high level, our proof is a bandwidth budget argument. We show that, for a given spatial frequency $\omega_x$, we have a fixed budget which must be shared by all motion slopes. A static camera spends most of this budget on static objects and therefore does poorly for other speeds. In contrast, our approach attempts to distribute this budget uniformly across the range of velocities and makes sure that no coefficient is low.

**Space time integration in the frequency domain:** We consider the Fourier domain $\omega_x, \omega_t$ of a scanline of space time. Fourier transforms will be denoted with a hat and Fourier pairs will be denoted with $k \Leftrightarrow \hat{k}$.

First consider the space-time function of a static object. It is constant over time, which means that its Fourier transform is non-zero only on the pure spatial frequency line $\omega_t$=0 (green object in Figure 4). This line is the 1D Fourier transform of the ideal instantaneous image $I^0$.

We have seen that image-space object velocity corresponds to the slope of a shear in space time. In the frequency domain, a given slope corresponds to a line orthogonal to the primal slope (Figure 4). Or equivalently, the shear in the primal corresponds to a shear in the opposite direction in the Fourier domain. The frequency content of an object at velocity $s$ is on the line of slope $s$ going through the origin. A range of velocities $-S \leq s \leq S$ corresponds to a double-wedge in the Fourier domain. This is similar to the link between depth and light field spectra [Chai et al. 2000; Isaksen et al. 2000]. This double-wedged is the frequency content that we strive to record. Areas of the Fourier domain outside it correspond to faster motion, and can be sacrificed.

Consider a light integration curve $f$ and its 2D trace $k(x,t)$ in space time, where $k(x,t)$ is non zero only at $x = f(t)$ (Figure 5). The 1D image scanline can be seen as the combination of a 2D convolution in space time by $k$, and a 2D to 1D slicing. That is, we



(a) Static (b) Parabola (c) Flutter Shutter (d) Upper Bound

**Figure 5: Top row**: *integration curve traces in space time.* **Bottom row**: *corresponding log spectrums*

lookup the result of the convolution only at time $t = 0$. The convolution step is key to analyzing the loss of frequency content. In the Fourier domain, the convolution by $k$ is a multiplication by its Fourier transform $\hat{k}$.

For example, a static camera has a kernel $k$ that is a box in time, times a Dirac in space. Its Fourier transform is a sinc in time, times a constant in space (Figure 5(a)). The convolution by $k$ results in a reduction of the high temporal frequencies according to the sinc. Since faster motion corresponds to larger slopes, their frequency content is more severely affected, while a static object is perfectly imaged.

In summary, we have reduced the problem to designing an integration curve whose spectrum $\hat{k}$ has the highest possible Fourier coefficients in the double-wedge defined by a desired velocity range.

**Slicing:** We now show that for each vertical slice of the Fourier doubled wedge, we have a fixed bandwidth budget because of conservation of energy in the spatial domain. That is, the sum of the squared Fourier coefficients for a given spatial frequency $\omega_x$ is bounded from above.

When studying *slices* in the Fourier domain, we can use the slicing theorem. First consider the vertical Fourier slice $\hat{k}_0$ going through $(0,0)$. In the primal space time, this Fourier slice corresponds to the *projection* along the horizontal $x$ direction.

$$\hat{k}_0(\omega_t) \Leftrightarrow k_p(t) = \int_x k(x,t)dx$$

Using the shifting property, we obtain an arbitrary slice for a given $\omega_x$ using

$$\hat{k}_{\omega_x}(\omega_t) \Leftrightarrow \int_x k(x,t)e^{-2\pi i\omega_x x}dx$$

which only introduces phases shifts in the integral.

**Conservation of energy:** We have related slices in the Fourier domain to space-only integrals of our camera's light integration curve in space-time. In particular, the central slice is the Fourier transform of $k_p(t)$, the total amount of light recorded by the sensor at a given moment during the exposure. Conservation of energy imposes

$$k_p(t) \leq 1 \tag{8}$$

Since $k$ is non-zero only during the $2T$ exposure time, we get a bound on the square integral

$$\int_t k_p(t)^2 dt \leq 2T \tag{9}$$

This bound is not affected by the phase shift used to extract slices at different $\omega_x$.

Furthermore, by Parseval's theorem, the square integral is the same in the dual and the primal domains. This means that for each slice at a spatial frequency $\omega_x$,

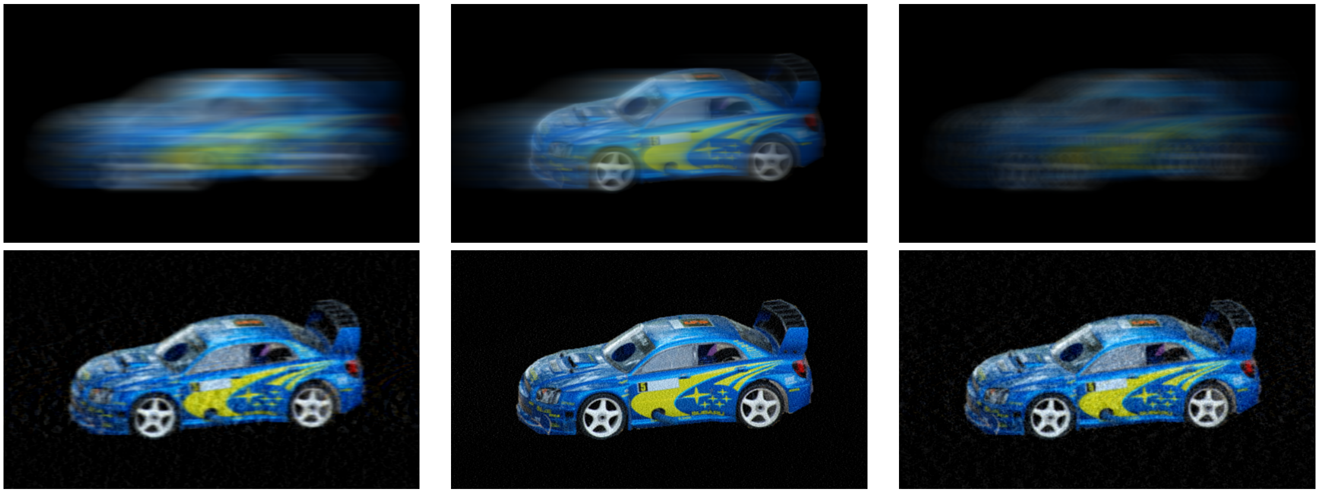
$$\begin{aligned} \int_{\omega_t} \hat{k}_{\omega_x}(\omega_t)^2 d\omega_t &= \int_t \left[k_p(t)e^{-2\pi i\omega_x x}\right]^2 dt & (10)\\ &\leq 2T & (11) \end{aligned}$$

The squared integral for a slice is bounded by a fixed budget of $2T$. In order to maximize the minimal frequency response, one should use a constant magnitude. Given the wedged shape of our velocity range in the Fourier domain, we get

$$\min_{\omega_t} |\hat{k}_{\omega_x}(\omega_t)|^2 \leq \frac{T}{S|\omega_x|} \tag{12}$$

where $S$ is the absolute maximal slope (speed). This upper bound is visualized in Figure 5(d). In other words, if we wish to maximize the spectrum of the PSFs over a finite slope range $-S \leq s \leq S$, eq 12 provides an upper bound on how much we can hope to achieve.

There is no free lunch: when one wants to cover a broader range of velocities, the budget must be split between a larger area of the Fourier domain and overall signal-noise ratio is reduced according to a square root law.



(a) Static (b) Parabola (c) Flutter Shutter

**Figure 6:** *Synthetic visualization of PSF information loss.* **Top***: Blurred input.* **Bottom***: Deblurring results.*

### 3.3 Discussion of different cameras

We have seen that in a traditional static camera, the light integration curve in space-time $k(x,t)$ is a vertical box function. Performances are perfect for the $\omega_t = 0$ line corresponding to static objects, but degrade according to a sinc for lines of increasing slope, corresponding to higher velocities.

The flutter-shutter approach adds a broad-band amplitude pattern to a static camera. The integration kernel $k$ is a vertical 1D function over $[-T,T]$ and the amount of recorded light is halved. Because of the loss of light, the vertical budget is reduced from $2T$ to $T$ for each $\omega_x$. Furthermore, since $k$ is vertical, its Fourier transform is constant along $\omega_x$. This means that the optimal flutter code must have constant spectrum magnitude over the full domain of interest (Figure 5(c)). This is why the spatial resolution of the camera must be taken into account. The intersection of the spatial bandwidth $\Omega_{max}$ of the camera and the range of velocities defines a finite double-wedge in the Fourier domain. The minimum magnitude of the slice at $\Omega_x$ is bounded by $\frac{T}{2S\Omega_x}$. Since $\hat{k}$ is constant along $\omega_x$, this bound applies to all $\omega_x$. As a result, for all band frequencies $|\omega_x| < \Omega_{max}$, $\hat{k}$ spends energy outside the slope wedge and thus does not make a full usage of the vertical $\hat{k}_{\omega_x}$ budget.

The parabolic integration curve attempts to distribute the bandwidth budget equally for each $\omega_x$ slice (Figure 5(b)), resulting in almost the same performance for each motion slope in the range, but a falloff proportional to $1/\sqrt{\omega_x}$ along the spatial dimension. To see why, we note that using some calculus manipulation the Fourier transform of an infinite parabola can be compute explicitly. If the integration kernel $k$ is defined via the (infinite) parabola curve $f(t) = a_0 t^2$, then

$$\hat{k}(\omega_x,\omega_t) = \frac{1}{\sqrt{2a_0\omega_x}} e^{i\frac{\omega_t^2}{4a_0\omega_x}} \tag{13}$$

On the other hand, achieving a good PSF for all $-S \leq s \leq S$ implies that $a_0 \geq \frac{S}{2T}$ (otherwise, from eq 6 the PSF won't include an infinite spike). Using eq 13 we can conclude that if the exposure was infinitely long $|\hat{k}(\omega_x,\omega_t)|^2 = \frac{T}{S|\omega_x|}$ and the infinite parabola would have the same falloff as the upper bound. Of course, the upper bound is valid for a finite exposure, and we can relate the infinite parabola to our finite kernel by a multiplication by a box in the spatial domain, which is a convolution by a sinc in the Fourier domain. Thus, the parabola curve of finite exposures approaches the upper bound, in the limit of long exposure times. The intuitive reason why the parabolic camera can better adapt to the wedged shape of the Fourier region of interest is that its kernel is not purely vertical, that is, the sensor is moving. A parabola in 2D contains edge pieces of different slopes, which corresponds to Fourier components of orthogonal orientation.

In summary, in the special case of 1D motions, if one seeks the ability to reconstruct a given range of image-space velocities, with minimal degradation to the reconstructed image for any velocity, the parabolic light integration curve is near optimal. On the other hand, a fluttered shutter can handle all motion directions, albeit at the cost of motion identification and image segmentation.

**Simulation:** To visualize these tradeoffs, in Figure 6 we synthetically rendered a moving car. We simulate a static camera, a parabolic displacement, and a static camera with a flutter-shutter, all with an identical exposure length and an equal noise level. To better demonstrate information loss, a simplified wiener filter deconvolution was applied. The box-deblurred car in Figure 6(a) lost high frequencies. The flutter-shutter reconstruction (Figure 6(c)) is much better, but the best results are obtained by the parabolic blur (Figure 6(b)).

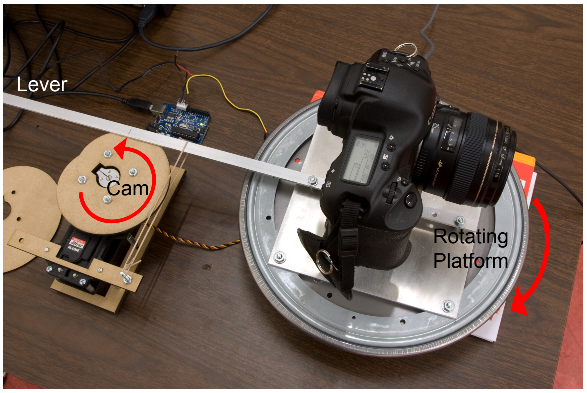


**Figure 7:** *Prototype camera setup.*

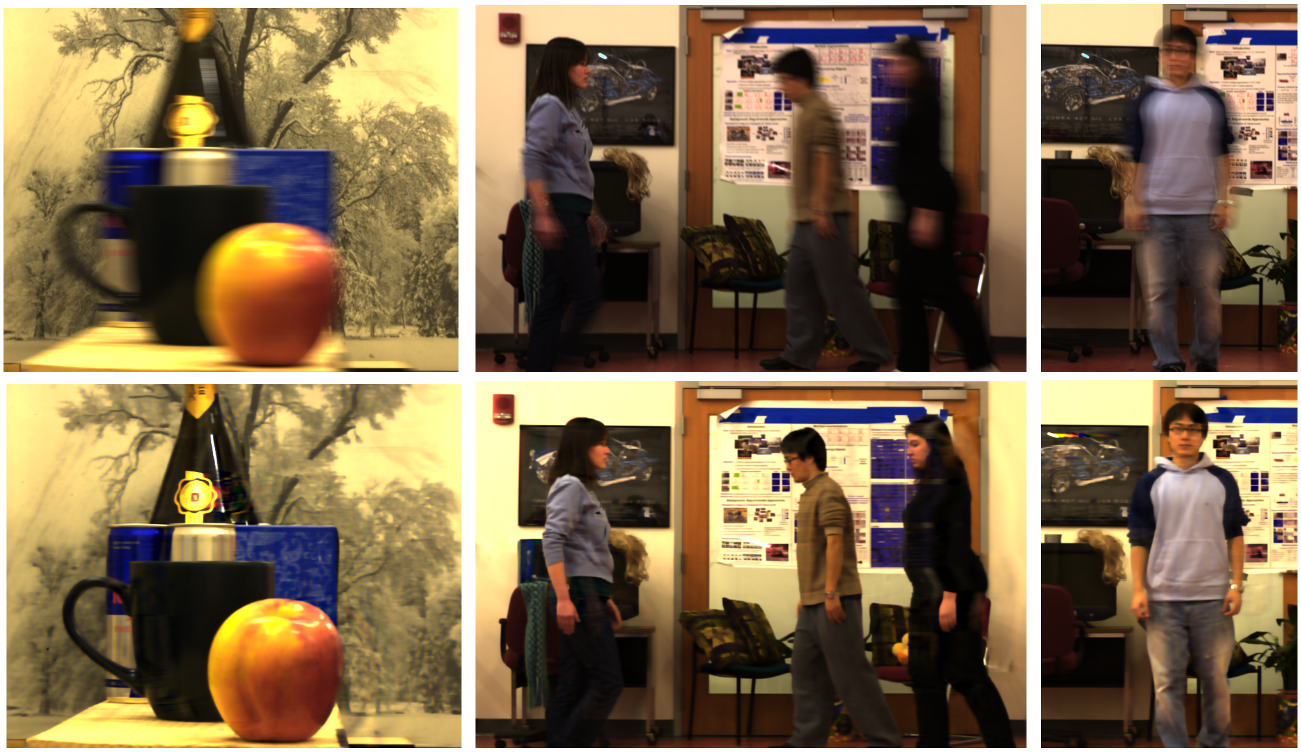


**Figure 8:** *Deblurring results.* **Top row***: image from a static camera.* **Bottom row***: deblurring results using the image from our parabolic camera.*

## 4 Experiments

While camera stabilization hardware should be capable of moving a detector with the desired constant acceleration (parabolic displacement) inside a hand-held camera, we chose to use larger–scale structures for our initial prototype, and approximate sensor translation using a rotation of the entire camera. The hardware shown in figure 7 rotates the camera in a controlled fashion to evaluate the potential of motion-invariant photography. The camera (Canon EOS 1D Mark II with an 85mm lens) sits on a platform that rotates about a vertical axis through the camera's optical center. We use a cam approach to precisely control the rotation angle over time (Figure 7.) A rotating cam moves a lever that is rigidly attached to the camera to generate the desired acceleration. For $\theta \in [-\pi, \pi]$, the cam edge is designed such that the polar coordinate radius is a parabola:

$$\begin{aligned} x(\theta) &= \cos(\theta)(c - b\theta^2) \quad (14) \\ y(\theta) &= \sin(\theta)(c - b\theta^2) \end{aligned}$$

In our experiments we used $c = 8$cm, $b = 0.33$cm. The cam rotates at a constant velocity, pushing the lever arm to rotate the camera with approximately constant angular acceleration, yielding horizontal motion with the desired parabolic integration path in space-time. For a fixed cam size, we can increase the magnitude of the parabola by moving the cam closer to the camera. We place a static camera next to the rotating camera to obtain a reference image for each moving-camera image (this camera is not part of our system and was used only for validation). A microcontroller synchronizes the cam rotation and the camera shutters. In order to reduce mechanical noise, the camera exposure length was set to 1 second. This relatively long exposure time limits the linear motion approximation for some real-world motions. To calibrate the exact PSF produced by our rotating camera we captured a blurred image $I$ of a static calibration pattern. We also captured a static image $I^0$ of the same pattern and solved for the PSF $\phi$ minimizing the squared convolution error: $\phi = argmin\|I^0 - \phi * I\|^2$.

### 4.1 Results

All deconvolution results presented in this paper were achieved with the sparse deconvolution algorithm of [Levin et al. 2007]. Comparable, but slightly worse, results can be obtained using the Richardson-Lucy deconvolution algorithm [Lucy 1974].

Figure 8 presents deblurring results on images captured by our camera. For all examples we present the scene captured by a synchronized static camera, and the deconvolution results obtained from our moving camera input (this input is not shown here). In the first image, the moving objects were mounted on a linear rail to create multiple velocities from the multiple depth layers. The other images involved natural human motions. Our approach deblurs the images reasonably well despite the fact that the human motions were neither perfectly linear nor horizontal. The middle image shows multiple independent motions in opposite directions, all deblurred well. The far-right image had many non-horizontal motions, resulting from the man walking toward the camera. While the face contains some residual blur, the deconvolved image has few objectionable artifacts. Results for more images under different test conditions are available in the supplementary file. We were encouraged by the deconvolution results on some images even with substantial non-horizontal motions. One possible explanation for this result is the aperture effect [Horn 1986], the ambiguity of the 2D motion of locally 1-dimensional image structures, such as edges and contours. The velocity component normal to the edge or contour is determined from the image data, but the parallel component is ambiguous. Local image motion that could be explained by horizontal motions (within the allowed range of motions) should deconvolve correctly, even though the true motions were not horizontal.

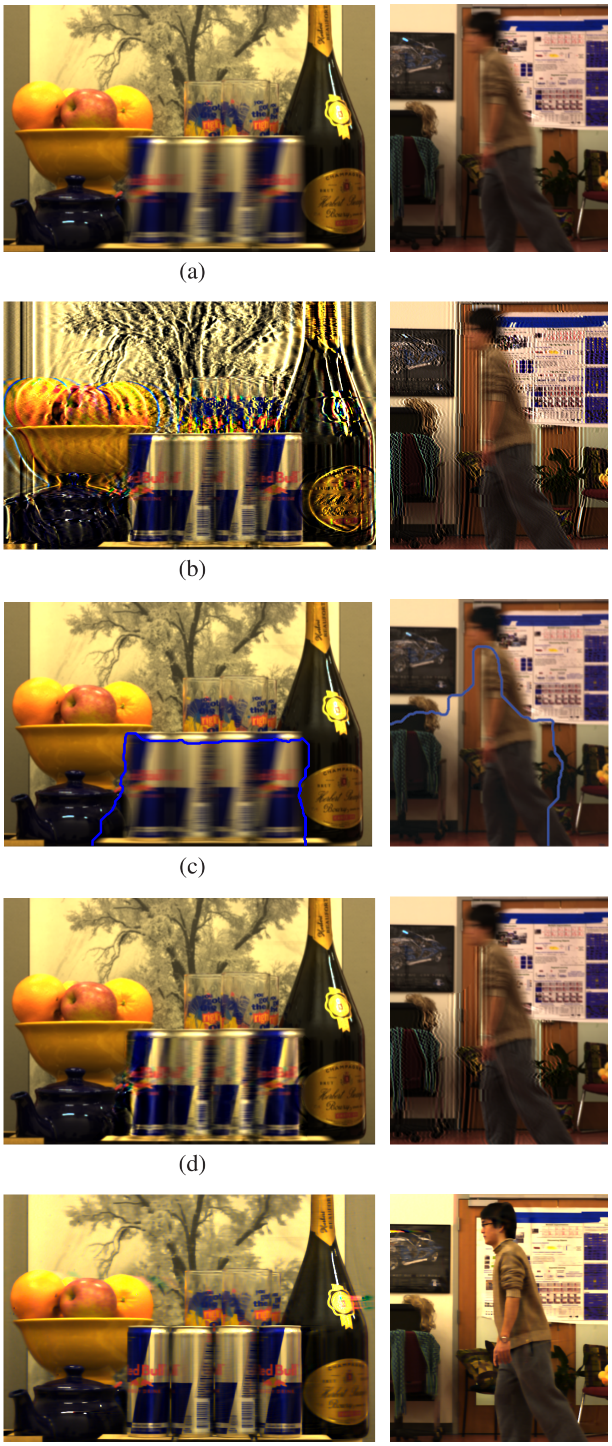
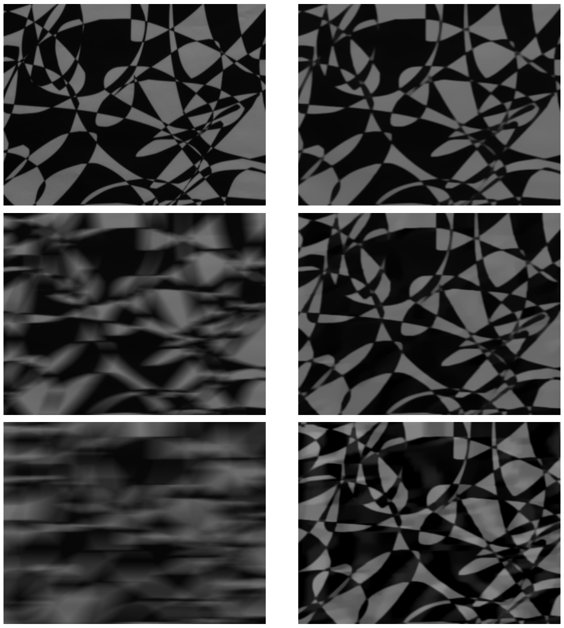


**Figure 9:** *Parabolic camera deblurring vs. state-of-the-art static camera deblurring. (a) Static camera input images, (b) Box filter fitted manually to moving layer and applied to deblur the entire image. (c) Layer segmentation by [Levin 2006]. (d) Debluring results by [Levin 2006]. (e) Our result, obtained by spatially uniform deconvolution of image from parabolic integration.*

Note that the static object fidelity in our results decreases with respect to the static camera input, as we uniformly distribute our bandwidth budget over a range of velocities.



(a) Static camera input (b) Parabolic deblurring output

**Figure 10:** *PSF clipping and handled velocity range.* **Top row***: Static scene.* **Middle row***: scene with slow motion.* **Bottom row***: scene with fast motion. While in the 1st two cases the static PSF approximation is reasonable, the tail clipping translates into artifacts in the faster motion case.*

Motion deblurring from a stationary camera is very challenging due to the need to segment the image by motion and estimate accurate PSFs within each segment. Figure 9 demonstrate these challenges. For example, we can try to deconvolve the image with a range of box widths, and manually pick the one producing the most visually plausible foreground results (Figure 9(b)). In the left image, deconvolving with the correct blur can sharpen the moving layer, but creates significant artifacts in the static parts, and an additional segmentation stage is required. In the right image, most blurred edges are occlusion boundaries and even manually identifying the correct blur kernel is challenging. In Figure 9(c-d) we compare our results with a recent automatic algorithm [Levin 2006]. While this algorithm did a reasonable job for the first image (which was captured using a linear rail) it did a much worse job on the human motion in the second one.

In Figure10 we illustrate the effect of PSF tail clipping (discussed toward the end of sec 2) on the valid velocity range. We used our parabolic camera to capture a binary pattern. The pattern was static in the first shot and for the other two, linearly moving during the exposure. The static motion was easily deblurred, and the PSF approximation is reasonable for the slow motion as well. For the faster motion, deconvolution artifacts start to be observed, as the effect of the clipping of the PSF tails becomes important and the static motion PSF is not an accurate approximation for the moving object smear.

Finally, Figure 11 illustrates the limitations of our technique for motions involving significant non-horizontal components, such as the man standing up or the board rotation. Note that despite the significant non–horizontal components, the reconstruction of many contours is actually artifacts free (such as the man's body and most of the rotating areas). This can possibly be explained by the aperture effect discussed above. Another source of artifacts are intensity non-linearities caused by highlights. Finally, both for the static and dynamic parts of the scene, the deconvolution sometimes leads to

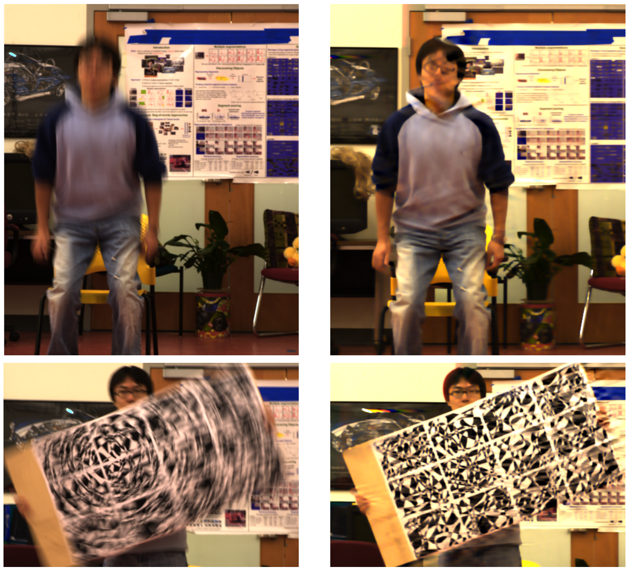

(a) Static image (b) Parabolic deblurring output

**Figure 11:** *Results for motions involving a significant non-horizontal component – man standing up and board rotation. Note artifacts in face, although also note that many contours reconstruct well. While some artifacts around the board boundary appear, many of the rotating areas are recovered surprisingly well.*

halo artifacts around high contrast edges. We suspect these artifacts result from imperfect PSF calibration or mechanical noise, and thus might be eliminated with a more careful mechanical implementation.

## 5 Summary and discussion

This paper suggests a simple solution that handles motion blur along a 1D direction. The camera translates within its exposure following a parabolic displacement rule. The blur resulting from this special camera motion is shown to be invariant to object depth and velocity. Hence, blur can be removed by deconvolving the entire image with an identical, known PSF. This solution eliminates the major traditional challenges involved with motion deblurring: the need to segment motion layers and estimate a precise PSF in each of them. Furthermore, we analyze the amount of information that can be maintained by different camera paths, and show that the parabolic path approaches the optimal PSF whose inversion is stable at all velocities.

A static camera, the flutter shutter camera, and a parabolic motion camera each offer different performance tradeoffs. A static camera is optimal for photographing static objects, but suffers significantly in its ability to reconstruct spatial details of moving objects. A flutter shutter camera is also excellent for photographing static objects (although records a factor of two less light than a full exposure). It provides good spatial frequency bandwidth for recording moving objects and can handle 2D motion. However, for reconstruction, one needs to identify the image velocities and segment regions of uniform motion. Motion-invariant photography requires no motion estimation or object segmentation and provides nearly optimal reconstruction for the worst-case speed within a given range. However, relative to the static camera and the flutter shutter camera, it gives a degraded reconstruction of static objects. While the method is only designed for 1D motions, we found it gave reasonable reconstructions of some 2D motions as well.

## Acknowledgments

This work was supported by an NSF CAREER award 0447561, a grant from Royal Dutch/Shell Group, NGA NEGI-1582-04-0004 and the Office of Naval Research MURI Grant N00014-06-1-0734. Fredo Durand acknowledges a Microsoft Research New Faculty Fellowship and a Sloan Fellowship.

## References

BAR, L., BERKELS, B., SAPIRO, G., AND RUMPF:, M. 2007. A variational framework for simultaneous motion estimation and restoration of motion-blurred video. In *ICCV*.

BASCLE, B., BLAKE, A., AND ZISSERMAN, A. 1996. Motion deblurring and superresolution from an image sequence. In *ECCV*.

BEN-EZRA, M., AND NAYAR, S. K. 2004. Motion-based motion deblurring. *PAMI*.

CANON. 2003. *EF Lens Work III, The Eyes of EOS*. Canon Inc. Lens Product Group.

CATHEY, W., AND DOWSKI, R. 1995. A new paradigm for imaging systems. *Applied Optics 41*, 1859–1866.

CHAI, J.-X., TONG, X., CHAN, S.-C., AND SHUM, H.-Y. 2000. Plenoptic sampling. In *Proceedings of ACM SIGGRAPH 2000*, Computer Graphics Proceedings, Annual Conference Series, 307–318.

FERGUS, R., SINGH, B., HERTZMANN, A., ROWEIS, S. T., AND FREEMAN, W. T. 2006. Removing camera shake from a single photograph. *ACM Transactions on Graphics 25*, 3 (July), 787–794.

HORN, B. K. 1986. *Robot Vision*. McGraw-Hill Higher Education.

ISAKSEN, A., MCMILLAN, L., AND GORTLER, S. J. 2000. Dynamically reparameterized light fields. In *Proceedings of ACM SIGGRAPH 2000*, Computer Graphics Proceedings, Annual Conference Series, 297–306.

LEVIN, A., FERGUS, R., DURAND, F., AND FREEMAN, W. T. 2007. Image and depth from a conventional camera with a coded aperture. *ACM Transactions on Graphics 26*, 3 (July), 70:1–70:9.

LEVIN, A. 2006. Blind motion deblurring using image statistics. In *Advances in Neural Information Processing Systems (NIPS)*.

LUCY, L. 1974. Bayesian-based iterative method of image restoration. *Journal of Astronomy*.

RASKAR, R., AGRAWAL, A., AND TUMBLIN, J. 2006. Coded exposure photography: motion deblurring using fluttered shutter. *ACM Transactions on Graphics 25*, 3 (July), 795–804.

RAV-ACHA, A., AND PELEG, S. 2005. Two motion-blurred images are better than one. *Pattern Recognition Letters*.

SHI, M., AND ZHENG, J. 2005. A slit scanning depth of route panorama from stationary blur. In *Proc. IEEE Conf. Comput. Vision Pattern Recog.*

YUAN, L., SUN, J., QUAN, L., AND SHUM, H.-Y. 2007. Image deblurring with blurred/noisy image pairs. *ACM Transactions on Graphics 26*, 3 (July), 1:1–1:10.