

Platform motion blur image restoration system

Stephen J. Olivas,^{1,*} Michal Šorel,² and Joseph E. Ford¹

¹Photonic Systems Integration Laboratory, Electrical Engineering Department, University of California at San Diego, 9500 Gilman Dr., La Jolla, California 92093, USA

²Department of Image Processing, Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, Praha 8, CZ-182 08, Czech Republic

*Corresponding author: sjolivas@ucsd.edu

Received 11 July 2012; revised 23 October 2012; accepted 24 October 2012;
posted 24 October 2012 (Doc. ID 172207); published 30 November 2012

Platform motion blur is a common problem for airborne and space-based imagers. Photographs taken by hand or from moving vehicles in low-light conditions are also typically blurred. Correcting image motion blur poses a formidable problem since it requires a description of the blur in the form of the point spread function (PSF), which in general is dependent on spatial location within the image. Here we introduce a computational imaging system that incorporates optical position sensing detectors (PSDs), a conventional camera, and a method to reconstruct images degraded by spatially variant SVplatform motion blur. A PSD tracks the movement of light distributions on its surface. It leverages more energy collection than a single pixel since it has a larger area making it proportionally faster. This affords it high temporal resolution as it measures the PSF at a specific location in the image field. Using multiple PSDs, a spatially variant PSF is generated and used to reconstruct images. © 2012 Optical Society of America

OCIS codes: 110.0110, 100.0100, 110.1758.

1. Introduction

Image degradation due to platform motion blur is a common problem for airborne and space-based imagers, where object motion is often relatively negligible [1,2]. It is also a problem when using cameras in low-light conditions [3], especially when taking pictures from cell phones, moving vehicles, or point-and-shoot cameras. Motion blur is accentuated by long exposure times or more pronounced motion. Optical sensors fundamentally restrict light intensity and integration time to adhere to fundamental exposure requirements. In general, the motion that causes blur, and therefore the point spread function (PSF), is unknown, and varies across the image field.

Deconvolution methods to deblur images without being provided the PSF are called blind deconvolution methods. Deblurring images in software using these methods [4] can reduce both spatially invariant

[5–8] and spatially variant (SV) [9–12] motion blur. Multiple images with different exposures can be used for this purpose, again for both space-invariant [13,14] and space-variant blur [15]. In recent years, many approaches were proposed that facilitate blind deconvolution using various alternative optical designs, such as coded aperture [16,17] or wavefront coding [18]. Prototypes that linearly accelerate the image sensors mechanically were built to successfully leverage one- [19] and two-dimensional [20] parabolic coded exposures; however, they only apply to invariant blur in predetermined directions of motion. For an introduction to the restoration of blurred images, see [21]. For other methods in the fast expanding field of computational photography and light-field imaging, see survey papers [22,23].

Limitations of blind deconvolution methods can be eliminated by measuring the PSF during image acquisition using additional hardware. Camera movement has been measured using accelerometers to produce an SV PSF [24], with a disadvantage that the resolution of the accelerometer does not scale

with the focal length of the lens. Another solution is to estimate the PSF using an additional low-spatial-resolution, high-temporal-resolution video camera [25,26]; however, this increases processing costs arising from the need to compute a PSF estimate from the video data. The approach proposed in this paper uses alternative hardware to achieve the same goals.

Platform motion blur varies across the image field in a structured fashion, constrained by only six degrees of freedom corresponding to translation and rotation of the camera. This structured nature of platform motion blur makes it sufficient to detect motion from the image itself at a few locations in order to generate an SV PSF. A position sensing detector (PSD) placed in the image field for this purpose can maintain high motion-tracking accuracy provided a bright feature on a dark background remains incident on its surface. Simulation and testing of the PSD motivated the construction of a prototype, Fig. 1, which simultaneously acquires images while tracking motion at specific image locations using a PSD array. These image-motion measurements are used to form an SV PSF from which to significantly restore blurred images. By leveraging its larger photon-energy collection area, the PSD far exceeds the temporal resolution of image sensors, allowing for the fast collection of image-motion information. Since the PSD in the prototype acts on the image itself, its resolution is independent of the object distance.

2. Measuring Image Motion Using PSDs

A lateral-effect PSD is essentially a single photodiode capable of tracking intensity centroid motion using the lateral photoeffect [27–29]. Light incident on the PSD generates a photocurrent (shown in green) that flows across the uniformly resistive sensor surface (Fig. 2). The amount of current that is drawn from each bottom electrode (blue) or that flows to each top electrode (red) is inversely proportional to the distance between the incident light and the particular electrode. The (X, Y) position of a spot of light on the sensor surface is given by Eq. (1), where L is the distance between input or output electrodes and i_i are the currents at the respective electrodes. A PSD position output value of zero volts references to the sensor center and can be calibrated using manufacturer circuitry. For more complex light projections incident on the sensor surface, the lateral effect acts on the superposition of the points that make up the

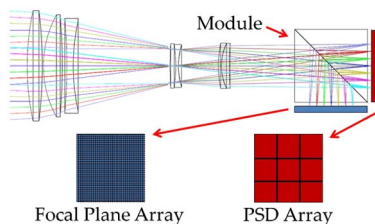


Fig. 1. (Color online) Platform motion deblur system schematic. A lens and beam splitter form identical image planes for the image sensor and PSD array to provide data for image restoration.

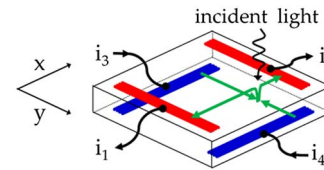


Fig. 2. (Color online) Lateral photoeffect of a PSD. Differential currents from input (blue) and output (red) electrodes produce sum (2) and position (1) data.

projection. This results in the device tracking the centroid of the illumination. The sensors also provide an S intensity (2) output, which is proportional to the net optical intensity at the sensor's surface. Tracking is not possible once the feature moves off a detection region. This event is revealed by the PSD outputs as a position reaching the sensor edge or as a change in intensity. A pathological case, which was simulated but does not appear in our experiments, occurs when one bright feature moves on the sensor while another moves off. This produces a position-tracking error value whose magnitude can be twice the size of the sensor; unfortunately, the PSD outputs would not indicate invalid data for this situation. We address this issue in Section 8.

$$X_{\text{PSD}} = \frac{i_1 - i_2}{i_1 + i_2} L \quad Y_{\text{PSD}} = \frac{i_3 - i_4}{i_3 + i_4} L, \quad (1)$$

$$S_{\text{PSD}} = i_1 + i_2 = i_3 + i_4. \quad (2)$$

The commercially available lateral-effect PSD modules used in our experiments were made by the On-Trak Photonics Corporation (module part number PSM2-4), which contains a Sitek Electro Optics PSD (part number 2L4SP). Each PSD has an active area of 16 mm², which corresponds to an area of more than 200,000 pixels in the image sensor. This affords the sensor more energy to make tracking proportionally faster. The lateral-effect PSD linearity, sensitivity [30], and transfer function [31] have been characterized. Its electronic and mechanical behavior and its application to spectroscopic optical systems [32] as well as tracking and displacement sensing [33] has been studied in depth. Specifications of the experimental lateral-effect PSDs are given in Table 1 and were verified by the manufacturer and as well as experimentally. It is possible for custom PSDs to be tailored to operate with high sensitivity or outside the visible band.

3. PSD Image Motion-Tracking Simulation

We modeled lateral-effect PSD behavior as images move across its surface in order to find how well it can measure the PSF. We used the definition of centroid position $(X_{\text{centroid}}, Y_{\text{centroid}})$ given by Eq. (3) to find the true centroid of images prior to simulated motion. Then this fixed centroid underwent the same SV movement as the images. This served as a benchmark for accurate centroid tracking. The

Table 1. On-Trak PSD (PSM2-4) Specifications

Parameter	Specification
Sampling rate	15 kHz
Position accuracy	100–200 nm
Position linearity	99.7%
Size	4 mm × 4 mm
Spectral response	400–110 nm
Required incident light	2.5 μW – 2.5 mW
Position output	–10 V–10 V
Sum output	0–6 V
Valid sum output	1–6 V

total intensity (S_{centroid}) is given by Eq. (4). Here $I(X, Y)$ is the optical intensity distribution at the sensor surface:

$$X_{\text{centroid}} = \frac{\int XI(X, Y)dXdY}{\int I(X, Y)dXdY},$$

$$Y_{\text{centroid}} = \frac{\int YI(X, Y)dXdY}{\int I(X, Y)dXdY}, \quad (3)$$

$$S_{\text{centroid}} = \int I(X, Y)dXdY. \quad (4)$$

In simulation, we then created frame-by-frame motion of the images using a known rotation and translation trajectory. A highly localized image [Fig. 3(a)] was blurred in simulation using SV movement. A video depicting the motion blur was used to calculate tracking error by comparing accurate motion [Eq. (3)] with a PSD-tracking model [Eq. (1)]. Gaussian noise with a positive mean value representing background illumination was included in the input images. Images used in simulation are shown in Figs. 3(b) and 3(c) with and without noise, respectively. In this way we quantified how well the PSD tracks images under varying levels of background illumination. The simulation showed that localized features on a dark background are easily tracked with subpixel error. The plot in Fig. 4 shows that tracking worsens for images as background illumination increases, since the optical energy distribution is less localized. As expected, this effect culminated in the PSDs' complete inability to track the position of scenes

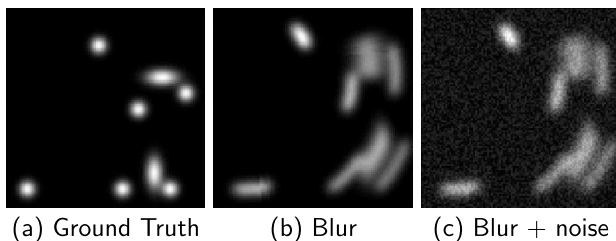


Fig. 3. Scenes input into PSD image motion-tracking simulation to characterize the effects of background illumination. (a) Ground-truth input image. (b) SV blurred image. (c) SV blurred image with Gaussian noise ($\sigma = 0.1$) with a positive mean value representing background illumination (5% of the maximum brightness).

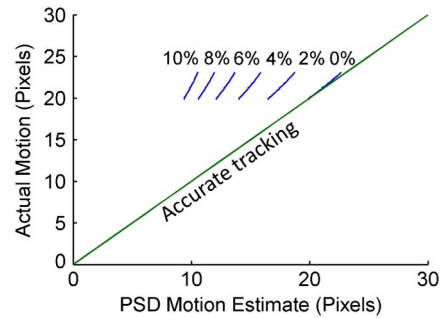


Fig. 4. (Color online) Simulation results: plot of PSD tracking error for various levels of background illumination showing the sensor is only capable of tracking image features much brighter than the background. Percentages represent the ratio between the brightness of the background and that of the tracked spots.

dominated by background. It is important to note that the PSD motion estimate of the centroid is skewed toward the sensor's center as ambient light increases (Fig. 5). This behavior follows from the centroid definition (3) since the PSD tracks the intensity centroid at the sensor's surface. These simulations were confirmed experimentally by projecting videos onto the PSD in a lab setting.

4. Generating a Spatially Variant PSF Using PSDs

The motion-blurred image z can be modeled by the linear operation

$$z(x, y) = [Hu](x, y) = \int u(s, t)h(x - s, y - t; s, t)dsdt, \quad (5)$$

where u is the unblurred image scene and h is the SV PSF. We can think of this operation as a convolution with a PSF that varies depending on pixel location (x, y) within the image. We denote the operation (5) in short form as Hu . Note that the PSF is a more general construct that can also represent other complex image degradations that depend on spatial or temporal coordinates, such as motion blur, lens distortions, out-of-focus blur, and atmospheric blur.

In the previous sections, we explained how the PSD can be used to estimate local image motion.

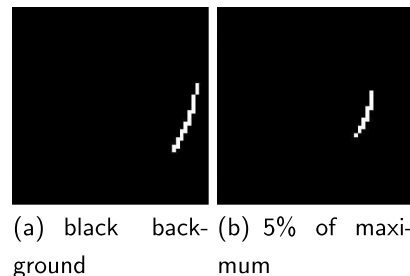


Fig. 5. PSD centroid tracking of identical image movement is skewed toward the sensor's center as background illumination increases. (a) Motion estimate from Fig. 3(b) with no background illumination and (b) from Fig. 3(c) with background illumination at 5% of the tracked spots.

The PSD output voltage signal represents the point location of the centroid of optical intensity at one instant in time. We can rapidly track these elementary shifts in time to get the PSF. This is valid for the pixel whose location in the image plane coincides with the PSD. In our experiments, we used three PSDs to give us the PSF for three different image coordinates. However, to be able to apply our model (5), we need to know the PSF h everywhere, for an arbitrary pixel coordinate.

Fortunately, there are sufficiently precise simplifications that allow PSD measurements to estimate an SV PSF for platform motion blur in most practical situations. One approach would be to consider only camera rotation, which was shown to be a good approximation for pictures taken by hand [10,12,24,34]. In this case, the PSF does not depend on the depth map and we can recover the change of the camera orientation from the motion of only two points [35]. Information from more than two PSDs can be used to make the estimate more robust.

In our experiments, we adopt an alternate approach, which assumes that the image motion within one elementary step can be modeled by the affine transform

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a(t) & b(t) \\ d(t) & e(t) \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} c(t) \\ f(t) \end{bmatrix}. \quad (6)$$

Motion blur depends on the spatial location (x,y) within the image as well as on time t . Rotation, scale, and shear manifest themselves in the coefficients a , b , d , and e , while translation is manifested in the c and f coefficients. The use of the affine transform can be justified by the fact that it includes as a subset in-plane translation (translation in x - and y -axes) and rotation about the optical axis, which is appropriate for many real scenes [11]. Computation can be made more robust by enforcing the transform to be rigid using singular value decomposition [36].

Coefficients of the transform, for each elementary time step t , are calculated using multiple regression solved by the method of least squares. This calculation is based on consecutive (in time) voltage information $(V_x(t), V_y(t))$ (Fig. 6) from PSDs placed in known locations within the image plane. At one instance, we collect two position voltage samples (X, Y) from each of the three PSDs. Each sample pair of consecutive PSD voltage data is mapped into pixel space to create its corresponding pixel information $(P_x(t), P_y(t))$; see Appendix A. Each sample pair of consecutive PSD voltage data is used to fit the affine model (6) forming a set of affine-model solutions, one for each sample pair of consecutive data. Each solution describes the motion that occurred from one time step to the next. These affine models are in turn used to generate the SV PSF (Fig. 7). The generation of the SV PSF is recursive in that as it is generated from the affine model, the current solution in time depends on the affine model of the previous time step. Three PSDs are sufficient for an unambiguous

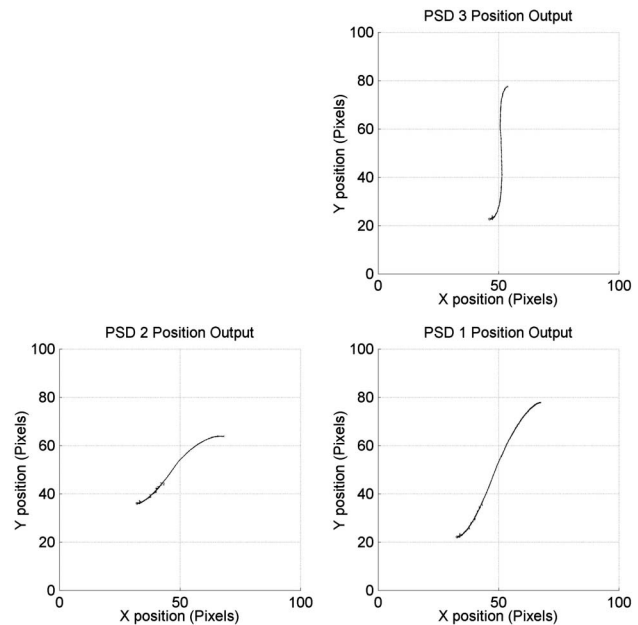


Fig. 6. PSD data at specific locations in the image field after calibration (Appendix A). These local PSF measurements are used to construct an SV PSF. Their relative location is given in Figs. 9 and 10.

estimate of transform coefficients. Our prototype system operated in the low-noise regime since the energy collection area of the PSD was large and illumination for tracking was ideal in our experimental setup; see Section 6. For this reason, we did not include a noise model in our coefficient estimation task.

5. System Prototype Design

The design space requires the imager and PSDs be in the same image plane, Fig. 1. It requires a long back focal distance lens in order for the beam splitter to occupy part of the optical path. Commercial cameras don't meet this requirement without modification and therefore we chose to build a system that uses two lenses that have long back focal distances. Similarly, two-dimensional closely tiled PSD arrays are not commercially available, so we met system requirements using three discrete PSDs.

We introduce a custom cube assembly fabricated to house the beam splitter, image sensor, and PSDs (Fig. 8). Two identical lenses are used to form

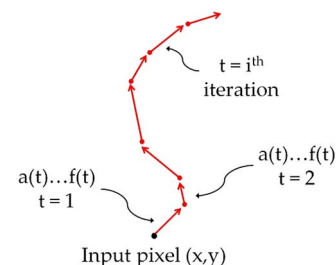


Fig. 7. (Color online) PSF specific to pixel (x,y) is recursively generated using affine model coefficients $a(t) - f(t)$ as the elementary time step t is integrated over the sampling time.

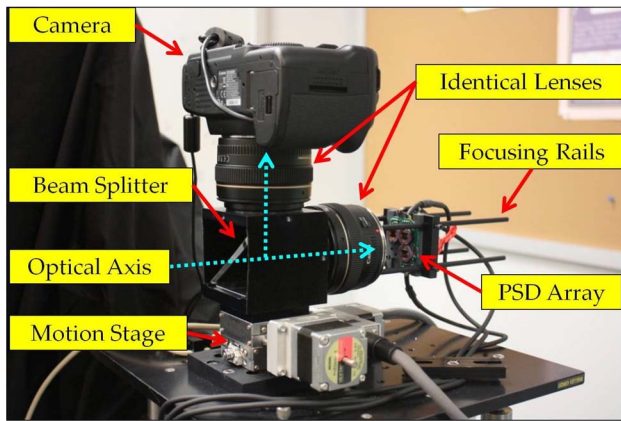


Fig. 8. (Color online) Photo of the imaging, position-sensing, motion-actuated system.

identical image planes. A motion stage creates reproducible motion. A custom plate mount holding the PSDs is supported by an adjustable sliding-rod configuration that allows for focusing of the PSDs. The computer-controlled system simultaneously triggers the motion stage, imaging camera, and PSD voltage acquisition, enabling the collection of blurry images and PSD data. Input parameters include aperture, ISO and exposure time for the camera, sampling rate, and number of samples for the PSD array and position, velocity, and acceleration for the motion stages. Misalignment due to the physical mounting of the PSDs to the plate is accounted for in the calibration method (Appendix A).

6. Experimental Setup

The laboratory experiments we conducted involve a starfield and a color image scene, Figs. 9 and 10. The images exhibit significant SV motion blur that exceeds 100 pixels at some image locations (Fig. 9). Both figures show the region of the scene that is to be restored; it is depicted enclosed in a red dashed line. The starfield scene is made up of dimly lit LEDs, randomly located within this region. Both scenes contain three bright LED sources focused onto the region of the image field where the PSDs are located. In this way we provide the PSDs' localized features with relatively little background illumination for tracking

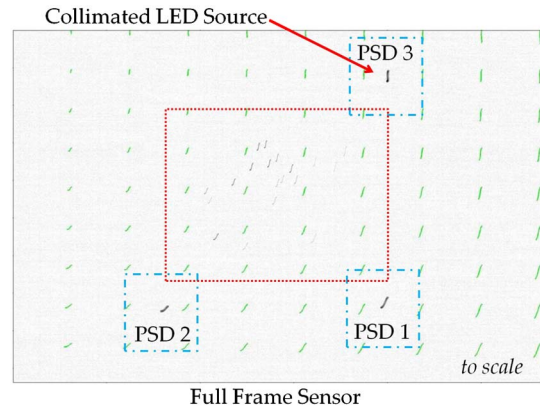


Fig. 10. (Color online) Experimentally blurred starfield image (contrast reversed). The calculated SV PSF is shown superimposed in green and is consistent with the SV image motion blur. The PSDs' relative locations are shown in cyan. The portion of the image that is to be deblurred is enclosed in red.

purposes. The physical scale and location of the PSDs in relation to a commercial full frame image sensor in our experiments are shown superimposed (in cyan) in Figs. 9 and 10. For all experiments performed, 30,000 sample voltage pairs (X, Y) are taken from each PSD during the 2 s acquisition of a single image. Overlaying the blurred image in Fig. 9 is the SV PSF (yellow lines) describing the calculated motion blur. Similarly, Fig. 10 has superimposed green lines that describe the calculated SV PSF. Both figures show that the SV PSF is consistent with the image blur.

7. Image Deblurring: Prototype Results

As soon as we have a reliable estimate of the SV PSF, we can deblur the image using one of many known image-restoration algorithms. The modern theory of image restoration is based on Bayesian statistics and algorithms usually seek the maximally probable solution, which is equivalently formulated as a minimization of certain functionals. We used a de facto standard solution, which can be expressed as the minimum

$$\min_u \left[\frac{1}{2} \|z - Hu\|^2 + \lambda \int |\nabla u| dx dy \right], \quad (7)$$

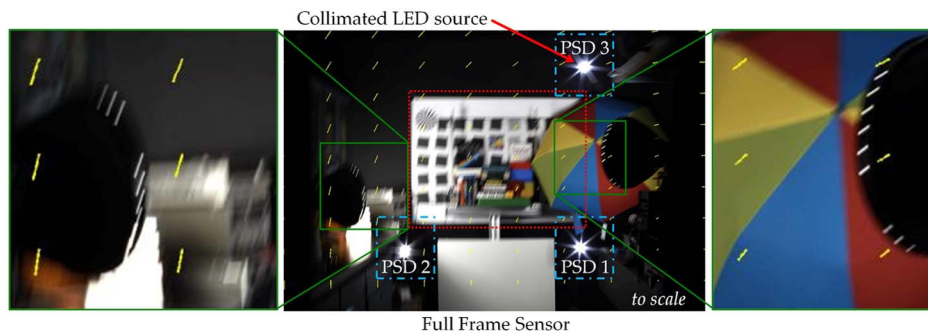


Fig. 9. (Color online) Experimentally blurred color image. Magnified sections of the image confirm SV blur. The calculated SV PSF is shown superimposed (yellow lines) and is consistent with the image motion blur. The PSDs' relative locations are shown in cyan. The portion of the image that is to be deblurred is enclosed in red.

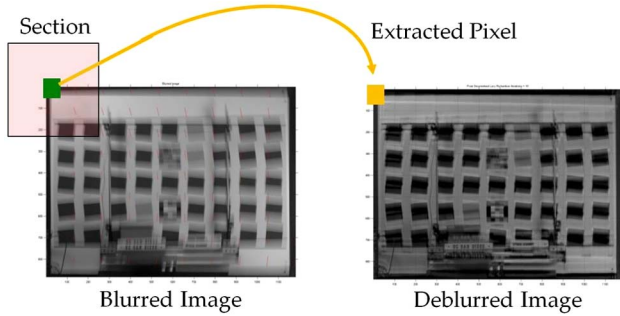


Fig. 11. (Color online) Pixel-by-pixel deconvolution. A PSF is generated for the blurred pixel of interest (green). A section in the neighborhood of the pixel is deblurred using the PSF. Only the center pixel (yellow) of the deblurred section is used in the final reconstruction.

where z is the blurred observed image, H is the SV PSF, u is the unknown sharp image, and λ is a positive regularization constant, usually set empirically. The left term penalizes discrepancy between model and measurements, and the right term is a so-called regularization term, which serves in the Bayesian framework as a statistical prior. In our case we utilize total variation, a regularization technique that exploits the sparsity of image gradients in natural images. Minimizing the convex functional (7) is

now considered a standard way to achieve close to state-of-the-art quality of restoration without excessive time requirements [4]. There are a number of methods to minimize the functional, many of them mathematically quite involved. We refer interested readers to a report that summarizes most of the latest developments [37]. We used an efficient method [38] solving Eq. (7) iteratively as a sequence of quadratic functionals,

$$u_{i+1} = \arg \min_u \left[\frac{1}{2} \|z - Hu\|^2 + \lambda \int \frac{|\nabla u|^2}{2|\nabla u_i|} + \frac{|\nabla u_i|}{2} dx dy \right]. \quad (8)$$

Functional (8) bounds the original function (7) from above and has the same value and gradient in the current estimate u_i , which leads to provable convergence to the global minimum of Eq. (7). To solve Eq. (8), we used the conjugate gradient method [39]. For details, see Section 3 in [38]. The description is relatively accessible as this method does not use more complex results from convex analysis, such as Fenchel's duality and Moreau's theorem, which are necessary in primal dual methods [37]. One example result of the procedure (8) can be seen in Fig. 12(c).

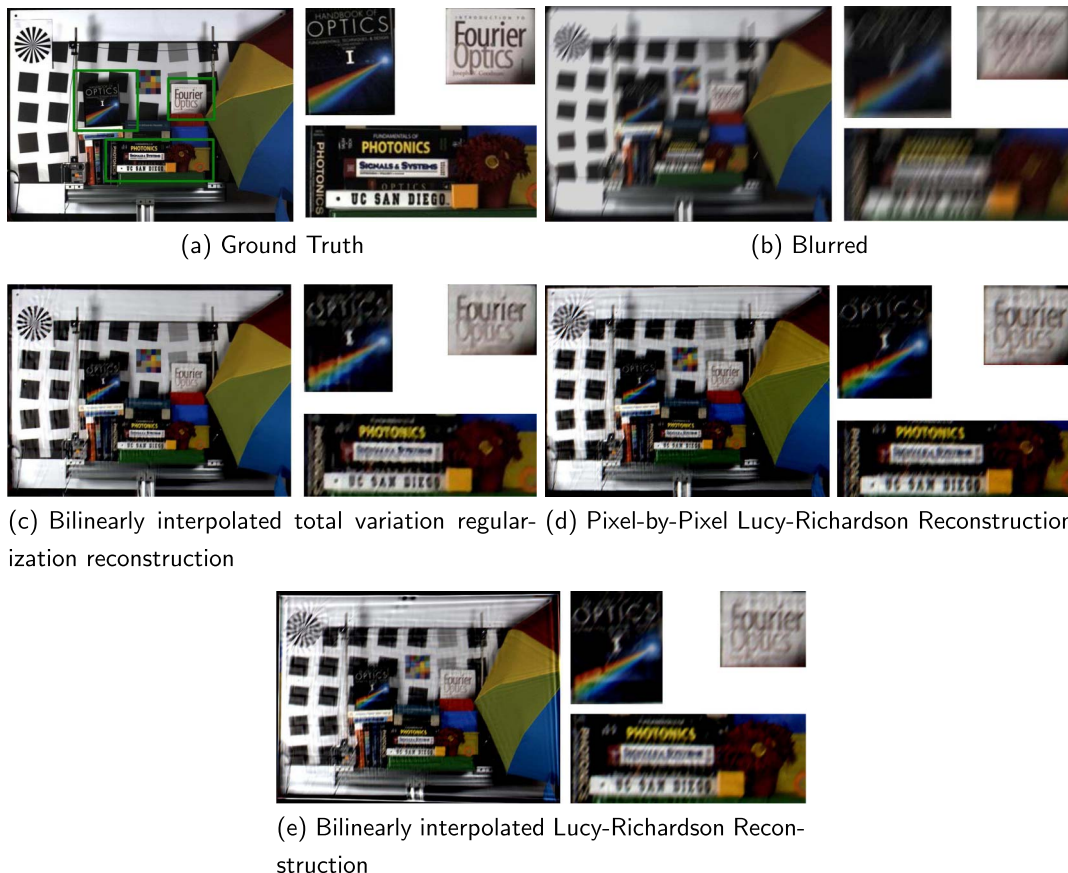


Fig. 12. (Color online) Region enclosed in red of the blurred image in Fig. 9 is deblurred using (c) bilinearly interpolated total variation regularization, (d) pixel-by-pixel Lucy–Richardson techniques, and (e) bilinearly interpolated Lucy–Richardson. Magnified sections of the images are also shown. (a) A ground-truth image is shown for comparison.

The main problem of any algorithm working in our scenario with the SV PSF (5) directly is that its computation is very time consuming and takes a lot of memory. To get an idea, consider our example; a 0.25 megapixel image in our example, 2 s exposure time with the PSD sampling at 15 kHz. Generation of the PSF at each and every pixel requires us to fit an affine or rigid transform 7.5×10^9 times. Each model fit is a nontrivial operation in itself that involves inversion of a small matrix or an singular value decomposition. If the size of the PSF is about 25×25 pixels and it is stored in 4-byte floating-point format, the PSF requires $25 \times 25 \times 4 \times 250000 = 625$ MB. Even if it is stored in a sparse form, it takes at least 60 MB of memory. Common present-day cameras produce 10–20 MP images, which gives us 40–80 times larger figures.

To make the problem tractable, we used bilinear interpolation to both save memory and speedup computation of the PSF [40]. This idea was used in practical deblurring algorithms [10,15,41] and recently for superresolution in [42]. The interpolation relies on the fact that blur is caused by camera motion where the PSF changes very slowly with position in the field of view. The PSFs are computed only on a regular grid of positions and the values of the PSF at intermediate positions are estimated with reasonable precision by bilinear interpolation of four adjacent known PSFs [40]. In our example in Fig. 12(c), we considered $15 \times 15 = 225$ positions and so we computed this number of PSFs. This number was chosen so that using more positions did not improve the quality of restoration significantly. Indexing any four adjacent grid points as $i = 1 \dots 4$ (starting in the top-left corner and going clockwise), the SV PSF in the rectangle among them is defined as

$$h(s, t; x, y) = \sum_{i=1}^4 \alpha_i(x, y) h_i(s, t), \quad (9)$$

where α_i are the coefficients of bilinear interpolation. Let us denote x_1 and x_2 minimum and maximum x -coordinates of the rectangle, respectively, and analogously y_1 and y_2 in the y -coordinates. Using auxiliary quantities

$$t_x = \frac{x - x_1}{x_2 - x_1} \quad t_y = \frac{y - y_1}{y_2 - y_1}, \quad (10)$$

the bilinear coefficients are

$$\begin{aligned} \alpha_1 &= (1 - t_y)(1 - t_x) \\ \alpha_2 &= (1 - t_y)t_x \\ \alpha_3 &= t_y(1 - t_x) \\ \alpha_4 &= t_y t_x. \end{aligned} \quad (11)$$

Space-variant convolution can be then computed as a sum of four convolutions of the image weighted by coefficients $\alpha_i(x, y)$

$$[Hu](x, y) = \int u(s, t) h(x - s, y - t; s, t) ds dt \quad (12)$$

$$= \int u(s, t) \sum_{i=1}^4 \alpha_i(s, t) h_i(x - s, y - t) ds dt \quad (13)$$

$$= \sum_{i=1}^4 \int (\alpha_i(s, t) u(s, t)) h_i(x - s, y - t) ds dt \quad (14)$$

$$= \left[\sum_{i=1}^4 [\alpha_i u] * h_i \right] (x, y). \quad (15)$$

All first-order minimization algorithms also need the operator adjoint to H (space-variant counterpart of correlation)

$$[H^*u](x, y) = \int u(s, t) h(s - x, t - y; x, y) ds dt \quad (16)$$

$$= \int u(s, t) \sum_{i=1}^4 \alpha_i(x, y) h_i(s - x, t - y) ds dt \quad (17)$$

$$= \sum_{i=1}^4 \alpha_i(x, y) \int u(s, t) h_i(s - x, t - y) ds dt \quad (18)$$

$$= \sum_{i=1}^4 \alpha_i(x, y) [u \otimes h_i](x, y). \quad (19)$$

In the method described above, the adjoint operator is used in the conjugate gradient method in the gradient of data term $(\partial / \partial u)(1/2) \|z - Hu\|^2 = H^*(Hu - z)$. Using the operators H and H^* , in forms (15) and (19), for large PSFs can be sped up significantly by computing convolutions and correlations using the fast Fourier transform.

Consider the huge savings we achieved in both time and memory. Memory consumption is now just $4 \times 25 \times 25 \times 225 \approx 560$ kB. While the PSF computation for all pixels took around 12 h, now it is slightly more than 2 min, more than 250 times less time (all algorithms implemented in MATLAB running on an ordinary personal computer).

We finish this section with a few remarks on the Lucy–Richardson algorithm [43–45], which is a well known iterative procedure that can be described as

$$u_{i+1} = u_i H^* \left(\frac{z}{Hu} \right). \quad (20)$$

Because of its simplicity of implementation and speed, it is still occasionally in use for rapid prototyping or for extremely large data in general. Even in this project, we used it initially to prove the concept of the proposed system. It can be applied in several ways. First, pixel by pixel in its space-invariant version where each pixel value of the restored image is acquired by deconvolving in its square neighborhood

Table 2. Time and Memory Requirements of the Described Algorithms Used in Fig. 12

PSF Estimate	Deblur Algorithm	Speed	Memory	RMSE
Blurred image				0.109
Pixel by pixel	Lucy–Richardson	12 h	625 MB	0.055
Bilinear interpolation	Lucy–Richardson	15 s	560 kB	0.082
Bilinear interpolation	Total variation	140 s	560 kB	0.044

and taking the central value (Fig. 11). Second, in the space-variant version, as proposed initially in [45] with PSF computed in each pixel, and finally using the bilinear interpolation as we did in the total-variation approach above. In the first two cases, the computation of the PSF totally dominates (above-mentioned 10 h); the third case is fast, even faster than the total variation approach with bilinear interpolation.

Figures 12(d) and 12(e) show the Lucy–Richardson approach in its two forms, in comparison with the total variation reconstruction in Fig. 12(c). We can see that the pixel-by-pixel Lucy–Richardson algorithm gives slightly better results than with the bilinear interpolation at the expense of computation time. The main reason is not interpolation but that it better suppresses boundary artifacts. The result of the total variation deblurring contains much fewer artifacts than both versions of the Lucy–Richardson algorithm. To better see the differences, readers are encouraged to zoom the results in the electronic version of this article. One important property of regularization is that it partially suppresses artifacts caused by an imprecise PSF. The Lucy–Richardson algorithm, on the other hand, produces many artifacts even for an ideal PSF and ideal photon noise for which it was derived. One reason is that it converges to the maximum-likelihood estimate from only one observation, which is known make the algorithm highly unstable. Table 2 summarizes the algorithms in terms of speed, memory consumption, and reconstruction quality measured by the root-mean-square error with respect to a ground truth taken from a

motionless camera. Notice that the difference between the total variation deblurring and pixel-by-pixel Lucy–Richardson does not seem so large in terms of mean-square error, but the visual difference is well noticeable. The reason is that visual perception cannot be expressed in such simple form. Alternatives for measuring image quality can be found in [46] and the references therein. An additional example of pixel-by-pixel Lucy–Richardson image reconstruction is provided in Fig. 13(c). This shows that in the simpler case of starfield images the algorithm works very well.

8. Future Directions

Although detecting compromised data using PSDs is not guaranteed, a PSD array could increase the probability of capturing valid data. A PSD array is not a new idea [47,48], but here we propose forming PSD aggregate groups to improve tracking capabilities. We did not explore this experimentally and would also require a custom fabricated device, which, to the best of our knowledge, has yet to made. A PSD array schematic is shown in Fig. 14(a), where valid and invalid tracking of bright features is denoted by green and red arrows, respectively. Motion tracking of image features whose trajectory produces any invalid data is undesirable. Invalid data could be monitored and used to trigger a camera shutter in order to prevent excessive blur. Invalid data from individual sensors may be recovered by joining adjacent PSDs in the array using Eq. (21) as shown in Fig. 14(b). Here $(X_{\text{aggregate}}, Y_{\text{aggregate}})$ denotes the centroid location

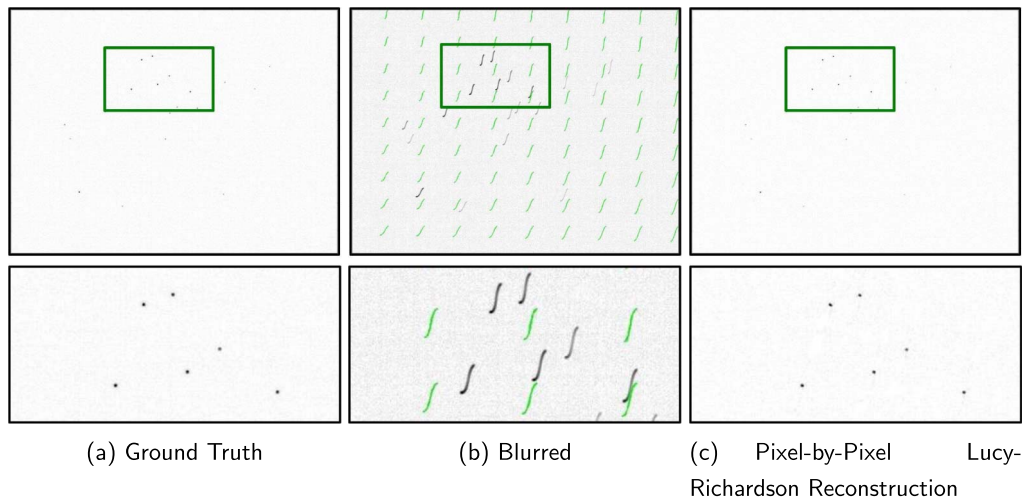


Fig. 13. (Color online) Region of Fig. 10 enclosed in red is deblurred using a pixel-by-pixel method with the iterative spatially invariant Lucy–Richardson algorithm. The (a) ground-truth, (b) blurred, and (c) five-iteration reconstructed images are shown for comparison along with magnified sections of the images. Superimposed green lines in (b) show calculated SV PSF.

over the closely tiled PSD aggregate array. $(X_{\text{PSD}}, Y_{\text{PSD}})$ is defined in Eq. (1) and is the centroid position output of a single PSD in the array. S_{PSD} is defined in Eq. (2). L is the distance between electrodes in a single PSD, (u, v) denote the index of PSDs in the array. The pairs $\{l, m\}$ and $\{o, p\}$ index the beginning and ending of the sensor group in the x - and y - directions, respectively; see Fig. 14(a). Piecewise concatenation of the position signal can be used when $l = m$ or $o = p$. As opposed to the discrete PSDs available for the experimental demonstration, an aggregate group requires a two-dimensional array of closely tiled PSDs to create a contiguous measurement space.

$$\begin{aligned} X_{\text{aggregate}} &= \frac{\sum_{u=l}^m \sum_{v=o}^p [L(u-1) + X_{\text{PSD}}(u, v)] S_{\text{PSD}}(u, v)}{\sum_{u=l}^m \sum_{v=o}^p S_{\text{PSD}}(u, v)}, \\ Y_{\text{aggregate}} &= \frac{\sum_{u=l}^m \sum_{v=o}^p [L(v-1) + Y_{\text{PSD}}(u, v)] S_{\text{PSD}}(u, v)}{\sum_{u=l}^m \sum_{v=o}^p S_{\text{PSD}}(u, v)} \end{aligned} \quad (21)$$

Compensation of background illumination to recover accurate PSF estimates could be made by using image data in the regions where the PSDs are located. This could be computationally expensive because it requires adaptive signal thresholding in order to estimate background-intensity levels. The reason arises from the fact that it is difficult to discern the moving signal from background illumination. Overcoming the need for negligible background illumination may be achieved through the use of gradient centroid sensors, which would operate independent of ambient background light.

Finally, a radial distortion model to compensate for the radial pincushion or barrel distortions can be

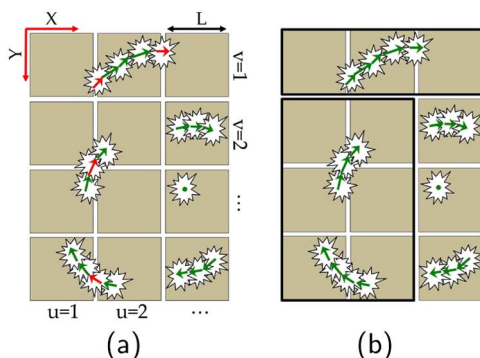


Fig. 14. (Color online) PSD can track the moving centroid of features that remains on the sensor under low background illumination. They provide *position* (1) and *intensity* (2) outputs, which can reveal invalid data. (a) A PSD array provides SV information and increases probability of valid data (green) capture. (b) Invalid data (red) can be recovered by aggregating closely tiled PSDs using Eq. (21) to form a sensor group.

included along with the affine model. While it is in principle possible to fit a distortion model directly to PSD data, it would require an excessive number of PSDs in the measurement space to sufficiently constrain this higher-order model. A better option is to measure distortions in advance (pp. 189–193 in [35]).

9. Conclusion

We introduced a proof-of-concept imaging system prototype and method to significantly reconstruct images degraded by SV motion blur. The system exploits the large energy collection area of the PSDs to make fast analog centroid position measurements. Data from a few PSDs at known locations in the

image field is mapped into pixel space and then used to fit an affine model valid for platform motion. Using the information from the model, we recursively generate an SV PSF for any pixel in the image enabling us to deblur images using a pixel-by-pixel deconvolution. We also show an alternate approach that uses a bilinearly interpolated PSF in conjunction with total variation to improve restoration quality and speed up computation.

The image sensor and PSD array both operate directly on the image itself, making the system resolution independent of object distance. The system hardware and algorithm functionality can be incorporated into the form factor of modern cameras and serves as a new direction in computational imaging. The PSDs limit the system to certain low background-illumination applications, such as for star imagers, where bright features of specific size and distribution appear on a dark background. Although not explored experimentally, we suggest as a possible direction of future research to overcome the low background-illumination requirement through the use of closely tiled PSD arrays along with grouping algorithms. As an alternative way to resolve this limitation, we propose the use of a not-commercially-available gradient centroid sensor. Our discussion of possible avenues to mitigate limitations of the system also merited the inclusion of spherical lens distortion models along with the affine model to extend system functionality.

Appendix A: Calibration—Mapping of Voltage to Pixels

To calibrate the PSD module, data is taken from the system in the absence of motion with an LED focused

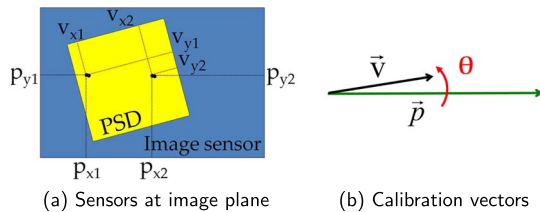


Fig. A1. (Color online) (a) Depiction of a PSD and imager's physical location in the image plane. PSD voltage and imager pixel information are paired at two separate image locations and used to form calibration vectors. (b) Depiction showing vectors whose angle and magnitude are used to map voltage data into pixel data.

on the PSD and imager. The PSD voltage output and the acquired image pixel information correspond to the centroid of the LED in the scene. In this way PSD voltages are paired up with image pixel data [Fig. A1]. This is done at two image locations. The voltages and pixel values are used to create the vectors, shown schematically in Fig. 15(b). These vectors are not necessarily collinear due to misalignment during the mounting of the sensors. The magnitudes of the two vectors and the angle between them are used to serve as calibration parameters, which are valid across the image plane:

$$\begin{bmatrix} P_x \\ P_y \end{bmatrix} = \frac{|P|}{|V|} \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} V_x \\ V_y \end{bmatrix}. \quad (\text{A1})$$

The authors thank Robin Dawson for technical discussion. This work was supported by the Charles Stark Draper Laboratory, contract SC001-000000628A. The work of the second author on this project was partially supported by the Czech Ministry of the Interior, Grant No. VG20102013064.

References

1. A. F. Boden, D. C. Redding, R. J. Hanisch, and J. Mo, "Massively parallel spatially variant maximum likelihood restoration of Hubble space telescope imagery," *J. Opt. Soc. Am. A* **13**, 1537–1545 (1996).
2. D. E. Meisner, "Fundamentals of airborne video remote sensing," *Remote Sens. Environ.* **19**, 63–79 (1986).
3. B. Hulgren and D. Hertel, "Low light performance of digital cameras," *Proc. SPIE* **7242**, 724214 (2009).
4. P. Campisi and K. Egiazarian, *Blind Image Deconvolution: Theory and Applications* (CRC, 2007).
5. R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, "Removing camera shake from a single photograph," *ACM Trans. Graph.* **25**, 787–794 (2006).
6. Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," *ACM Trans. Graph.* **27**, 73:1–73:10 (2008).
7. J.-F. Cai, H. Ji, C. Liu, and Z. Shen, "High-quality curvelet-based motion deblurring from an image pair," in *Proceedings of Conference on Computer Vision and Pattern Recognition* (IEEE, 2009), pp. 1566–1573.
8. F. Sroubek and J. Flusser, "Multichannel blind deconvolution of spatially misaligned images," *IEEE Trans. Image Process.* **14**, 874–883 (2005).
9. Q. Shan, W. Xiong, and J. Jia, "Rotational motion deblurring of a rigid object from a single image," in *Proceedings of 11th International Conference on Computer Vision* (IEEE, 2007), pp. 1–8.
10. S. Harmeling, M. Hirsch, and B. Schölkopf, "Space-variant single-image blind deconvolution for removing camera shake," *Adv. Neural Inf. Process. Syst.* **23**, 829–837 (2010).
11. A. Gupta, N. Joshi, C. Lawrence Zitnick, M. Cohen, and B. Curless, "Single image deblurring using motion density functions," in *Proceedings of the 11th European Conference on Computer Vision: Part I* (Springer-Verlag, 2010), pp. 171–184.
12. O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, "Non-uniform deblurring for shaken images," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (IEEE, 2010), pp. 491–498.
13. M. Tico, M. Trimeche, and M. Vehvilainen, "Motion blur identification based on differently exposed images," in *Proceedings of International Conference on Image Processing* (IEEE, 2006), pp. 2021–2024.
14. L. Yuan, J. Sun, L. Quan, and H.-Y. Shum, "Image deblurring with blurred/noisy image pairs," in *SIGGRAPH '07: ACM SIGGRAPH 2007* (ACM, 2007), p. 1.
15. M. Sorel and F. Sroubek, "Space-variant deblurring using one blurred and one underexposed image," in *Proceedings of 16th International Conference on Image Processing (ICIP)* (IEEE, 2009), pp. 157–160.
16. R. Raskar, A. Agrawal, and J. Tumblin, "Coded exposure photography: motion deblurring using fluttered shutter," *ACM Trans. Graph.* **25**, 795–804 (2006).
17. A. Agrawal and R. Raskar, "Resolving objects at higher resolution from a single motion-blurred image," in *Proceedings of IEEE Conference Computer Vision and Pattern Recognition* (IEEE, 2007), pp. 1–8.
18. E. R. Dowski, Jr. and W. T. Cathey, "Extended depth of field through wave-front coding," *Appl. Opt.* **34**, 1859–1866 (1995).
19. A. Levin, P. Sand, T. Sang Cho, F. Durand, and W. T. Freeman, "Motion-invariant photography," *ACM Trans. Graph.* **27**, 71:1–71:9 (2008).
20. T. Sang Cho, A. Levin, F. Durand, and W. T. Freeman, "Motion blur removal with orthogonal parabolic exposures," in *Proceedings of International Conference on Computational Photography (ICCP)* (IEEE, 2010), pp. 1–8.
21. P. Christian Hansen, J. G. Nagy, and D. P. Leary, *Deblurring Images: Matrices, Spectra and Filtering* (SIAM, 2006).
22. C. Zhou and S. Nayar, "Computational cameras: convergence of optics and processing," *IEEE Trans. Image Process.* **20**, 3322–3340 (2011).
23. G. Wetzstein, I. Ihrke, D. Lanman, and W. Heidrich, "Computational plenoptic imaging," *Computer Graphics Forum*, 2397–2426 (2011).
24. N. Joshi, S. B. Kang, C. L. Zitnick, and R. Szeliski, "Image deblurring using inertial measurement sensors," *ACM Trans. Graph.* **29**, 30:1–30:9 (2010).
25. S. K. Nayar and M. Ben-Ezra, "Motion-based motion deblurring," *IEEE Trans. Pattern Anal. Mach. Intell.* **26**, 689–698 (2004).
26. Y.-W. Tai, H. Du, M. S. Brown, and S. Lin, "Correction of spatially varying image and video motion blur using a hybrid camera," *IEEE Trans. Pattern Anal. Mach. Intell.* **32**, 1012–1028 (2010).
27. W. Schottky, "Ueber den entstehungsort der photoelektronen in kupfer-kupferoxydul photozellen," *Phys. Z.* **31**, 913–925 (1930).
28. J. T. Wallmark, "A new semiconductor photocell using lateral photoeffect," *Proc. IRE* **45**, 474–483 (1957).
29. H. Niu, C. Aoki, T. Matsuda, M. Takai, and M. Maeda, "A position-sensitive MOS device using lateral photovoltaic effect," *Jpn. J. Appl. Phys.* **26**, L35–L37 (1987).
30. W. Wang and I. J. Busch-Vishniac, "The linearity and sensitivity of lateral effect position sensitive devices—an improved geometry," *IEEE Trans. Electron Devices* **36**, 2475–2480 (1957).
31. M. de Bakker, P. W. Verbeek, G. K. Steenvoorden, and I. T. Young, "The PSD transfer function," *IEEE Trans. Electron Devices* **49**, 202–206 (2002).
32. H. Andersson, "Position sensitive detectors: device technology and applications in spectroscopy," Ph.D. thesis (Mid Sweden University, Department of Information Technology and Media, 2008).

33. A. Makynen, "Position-sensitive devices for optical tracking and displacement sensing applications," Ph.D. thesis (University of Oulu, 2000).
34. M. Ben-Ezra and S. K. Nayar, "Motion deblurring using hybrid imaging," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (IEEE, 2003), pp. I-657–I-664.
35. R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. (Cambridge University, 2003).
36. B. K. P. Horn, H. M. Hilden, and S. Negahdaripour, "Closed-form solution of absolute orientation using orthonormal matrices," *J. Opt. Soc. Am. A* **5**, 1127–1135 (1988).
37. A. Chambolle, V. Caselles, D. Cremers, M. Novaga, and T. Pock, "An introduction to total variation for image analysis," in *Theoretical Foundations and Numerical Methods for Sparse Recovery* (De Gruyter, 2010).
38. A. Chambolle and P. Lions, "Image recovery via total variation minimization and related problems," *Numer. Math.* **76**, 167–188 (1997).
39. G. H. Golub and C. F. Van Loan, *Matrix Computations* (Johns Hopkins University, 1996).
40. J. G. Nagy and D. P. O'Leary, "Restoring images degraded by spatially variant blur," *SIAM J. Sci. Comput.* **19**, 1063–1082 (1998).
41. M. Hirsch, C. J. Schuler, S. Harmeling, and B. Schölkopf, "Fast removal of non-uniform camera shake," in *Proceedings of International Conference on Computer Vision (ICCV)* (IEEE, 2011), pp. 463–470.
42. M. Sorel and F. Sroubek, *Restoration in the Presence of Unknown Spatially Varying Blur in Image Restoration: Fundamentals and Advances* (CRC, 2012).
43. R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. (Prentice-Hall, 2006).
44. W. H. Richardson, "Bayesian-based iterative method of image restoration," *J. Opt. Soc. Am. A* **62**, 55–59 (1972).
45. L. B. Lucy, "An iterative technique for the rectification of observed distributions," *Astron. J.* **79**, 745–754 (1974).
46. A. Hore and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *Proceedings of 20th International Conference on Pattern Recognition (ICPR)*, 2010, pp. 2366–2369.
47. N. Massari, L. Gonzo, M. Gottardi, and A. Simoni, "High speed digital CMOS 2D optical position sensitive detector," in *Proceedings of Solid-State Circuits Conference (ESSCIRC)*, 2002, pp. 723–726.
48. M. A. Clapp and R. Etienne-Cummings, "A dual pixel-type array for imaging and motion centroid localization," *IEEE Sens. J.* **2**, 529–548 (2002).