Light Field Blind Deconvolution

Paramanand Chandramouli, Daniele Perrone and Paolo Favaro

Abstract

We address for the first time the issue of motion blur in light field images captured from plenoptic cameras (instead of camera arrays), where the spatial sampling in each view is decimated. We propose a solution to the estimation of a sharp light field given a blurry one, when the motion blur point spread function is unknown, i.e., the so-called blind deconvolution problem. Unfortunately, the (decimated) spatial sampling of each light field view does not allow the use of current blind deconvolution approaches for traditional cameras. Also, due to the complexity of the imaging model, we investigate first the case of uniform (shift-invariant) blur of Lambertian objects, i.e., when objects are sufficiently far away from the camera to be approximately invariant to depth changes and their reflectance does not vary with the viewing direction. We introduce a highly parallelizable model for light field motion blur that is computationally and memory efficient. We then adapt a regularized blind deconvolution approach to our model and demonstrate its performance on both synthetic and real light field data.

Fig. 1. Motion blur in a real light field image: a) Raw light field image (zoom in to see microlenses); b) super resolved central view (motion blur is still present); c) super resolved and deblurred central view; d) image from a (static) conventional camera at the same resolution.

I. INTRODUCTION

With Lytro and Raytrix [1], [2], plenoptic cameras have entered into the realm of consumer photography by demonstrating capabilities, such as digital refocusing, not possible in traditional devices. This has led to an increased interest in the scientific community in high-quality light field reconstruction. As these commercial cameras are portable, camera shake is sometimes unavoidable and may result in blurry light fields. Similarly, due to the finite exposure of the sensor, pictures of moving objects might also appear blurry.

Until now, software for these imaging systems has not been designed to handle motion blur (see Fig. 1 for an example of a real motion blurred light field). In contrast, motion blur in conventional cameras has been widely studied and current methods achieve remarkable results (see, for instance, [9], [26], [36], [11], [22]). Unfortunately, conventional images and light field (LF) images undergo a quite different process when motion blurred and therefore current methods cannot be adapted in a straightforward manner. For instance, motion blur is never shift-invariant in a blurred light field. This behavior is due to the additional microlens array between the main lens and the sensors in the LF camera [25].

When a LF camera moves relative to an object the measured light field will be a combination of out of focus blur (due to the microlenses) and motion-blur. All these effects can be captured in a model, but its complexity leads to a very computationally intensive and memory inefficient mapping. Because

The authors are with Institut für Informatik und angewandte Mathematik, University of Bern, Switzerland. E-mail: chandra@iam.unibe.ch, perrone@iam.unibe.ch, paolo.favaro@iam.unibe.ch.
of these challenges, most of recent work ignores blur when dealing with light fields. However, we show that, when imaging scenes at infinity or fronto parallel, it is possible to describe a motion blurred light field as a linear combination of parallel convolutions (see sec. II-A). As a result, the model is extremely computationally and memory efficient. This model is then used to solve the problem of recovering the motion blur and a sharp image of the scene via an alternating minimization technique. As a regularizer for the sharp image, we use total variation.

II. RELATED WORK

There is no prior work that deals with motion blurred light field reconstruction. However, since our work relates to both light field reconstruction and motion deblurring, we provide a brief overview of some methods developed in both areas.

**Single image motion deblurring.** Motion deblurring involves the joint estimation of a sharp image and a blur kernel [38]. Because of its ill-posedness, motion deblurring is typically addressed by enforcing prior information on the sharp image and on the blur kernel [9], [11], [26], [36], [20]. Commonly used image priors enforce a heavy-tail distribution on the image gradients, as suggested by recent findings in natural image statistics [28]. A popular choice is the Laplace distribution [22], total variation (TV) regularization [39], [7] or $L_0$ regularization [37]. Some other methods encourage sharp edges by using a shock filter [9], [36] or a dictionary of sharp edges [29]. For the blur function some methods use a Gaussian distribution [36], [9], others a sparsity-inducing distribution [11], [27] or a uniform distribution [22]. The aforementioned recent methods assume uniform blur across the image plane. With the introduction of a novel blur formation model many algorithms can be adapted to include also rotational camera shake blur [30], [35], [13], [15], [16]. Nonetheless, the increase in the dimension of the blurring function could result in more computational load which may not lead to significant improvement over the space-invariant deblurring schemes [19]. Other methods attempt to solve general space-varying motion deblurring by estimating locally uniform blur and carefully interpolate them to cope with regions with poor texture [17], or by iteratively employing a uniform deblurring and segmentation algorithms [18].

**Plenoptic cameras, camera arrays and calibration.** The first plenoptic camera was developed by Adelson and Wang for inferring scene depth from a single snapshot [3]. The portable design by Ng et al. with the use of microlens arrays triggered the development of handheld light field cameras [25]. The main drawback of the first image reconstruction methods for plenoptic cameras was their limited spatial resolution. To overcome this problem, techniques for super-resolving the data up to sensor resolution have since then been proposed [4], [24]. Danserau et al. propose a decoding, calibration and rectifying procedure for lenselet-based plenoptic cameras [10]. They present a 15 parameter plenoptic camera model that relates pixels to rays in 3D space. Cho et al. [8] develop a method for camera calibration as well as for rendering the light field at a higher resolution. Both [10] and [8] demonstrate their method on the Lytro camera. Light field data can also be obtained by using camera arrays or a gantry-based moving camera [23], [31]. A camera array-based design for practical mobile devices which can synthesize high resolution images and depth maps is proposed in [32]. An important difference between camera arrays and plenoptic cameras is that the spatial and angular sampling resolutions are different. In the case of the plenoptic cameras the spatial sampling is low while the angular sampling is high. The opposite is true in the case of camera arrays. We also found out that recent work of Broxton et al. [6] might suggest a fast computational scheme similar to the one proposed here (a very short explanation is given in a paragraph in sec.3.4 of [6]). However, our and this scheme were developed simultaneously. Moreover, our scheme includes the case of motion blur.

**Light field reconstruction methods.** Wanner and Goldlücke [34] propose a technique to super-resolve 4D light fields in both spatial and angular direction. In related work, Goldlücke et al. [12] propose also a super-resolution framework to increase the texture quality obtained from a camera array system. Other recent work is that of Heber et al. [14] where the optimization is solved via a Primal-Dual method. Georgiev et al. [5] incorporate demosaicing as part of the reconstruction process, claiming the theoretical possibility for rendering natural images at full sensor resolution.
Contributions. Our focus is on lenselet-based plenoptic cameras and we consider only uniform motion blur and Lambertian scenes. The contributions of our work can be summarized as: i) This work is the first solution to motion deblurring of light field images from a plenoptic camera. ii) We propose a computationally and memory efficient imaging model for motion blurred LF images. iii) We solve a joint blind deconvolution and super resolution problem as the light field is recovered at a resolution much higher than that of a single view.

III. IMAGING MODEL

In this section we introduce notation and the complete model for a light field image. The model relates the light field image captured by a plenoptic camera with the scene radiance via an explicit PSF. We choose the microlens array plane to be the domain of the scene texture \( g \). The light field image \( f \) is defined on the image sensor plane. Let \( p \) denote the coordinates of a point on the microlens array and \( x \) denote a pixel location on the sensor plane. We have \( p = [p_1 \ p_2]^T \), and \( x = [x_1 \ x_2]^T \), where \( p_1, p_2, x_1, x_2 \) are discretized. A pixel of the LF image \( f(x) \) is related to texture elements \( g(p) \) through a space-varying point spread function \( h(x, p) \) as

\[
f(x) = \sum_p h(x, p)g(p).
\] (1)

In general, the PSF \( h \) depends on the scene depth map and the parameters of the plenoptic camera. An explicit formula is available in [6], [4]. As discussed in the Introduction, we consider the case where objects are at infinity or when the depth map is constant and known. A relative motion between the camera and the scene during the exposure interval causes the light field image to be motion blurred. The observed motion blurred LF image can be written as a weighted sum of multiple light field images corresponding to shifted scene texture. The weights define the motion blur PSF \( h_m \). We then have the motion blurred light field \( f_m \) as

\[
f_m(x) = \sum_p h(x, p)\sum_q h_m(q)g(p - q) = \sum_p h(x, p)g_m(p)
\] (2)

where \( g_m(p) \equiv \sum_q h_m(q)g(p - q) \). The model (2) is made very inefficient because of the highly dimensional PSF \( h \). In the next section we explain how to drastically reduce its complexity by exploiting its highly parallelizable structure.

A. Convolution-based LF image generation

A direct implementation of eq. (1) is practically not feasible due to the memory and computation requirements. For instance, if the scene radiance and the light field image were 1 mega pixels, the number of elements necessary to store \( h \) would be of the order of \( 10^{12} \). Although \( h \) is sparse, performing the sum and product calculation in eq. (1) would still be computationally intensive. We address these shortcomings by exploiting the structure of LF PSFs. As a result we obtain an exact and efficient implementation of LF image generation that is highly parallelizable.

We first provide an intuitive explanation of the key idea that we exploit. Then, we present it formally. Consider two LF images that are generated with the same texture \( g \), but differ by a small shift of \( g \) along the \( X \) axis. If the shift in the texture \( g \) matches the distance between two microlenses, the two LF images will turn out to be exact copies of each another up to a shift equal to the number of pixels under a microlens. To illustrate this fact, we captured a few real light field images by shifting a poster from left to right while keeping our LF camera still. In the first row of Fig. 2 we show the same region cropped from three light field images. We can observe the shift of the edge pattern in these images as the poster is shifted. Now compare the first with the third image. They are (almost) identical up to a shift of 1 microlens. To better see such match, we then show in the second row of Fig. 2 zoomed-in patches. While the red and green patches are almost similar, the yellow patch has a shift corresponding to one
Fig. 2. Illustration of the periodicity of a LF image. Row 1: three different LF images captured by horizontally shifting texture. Zoomed-in patches whose colors indicate the portion that was cropped from the images in the first row.

microlens (with respect to the red patch). The blue patch corresponds to a texture with an intermediate shift. Thus, we can see that there is a periodic structure in the light field. The artifact at the edge of a microlens visible in the zoomed-in patches is due to dust on CCD sensor.

To formalize this periodicity, we need to introduce some basic notation. Suppose there are $J \times J$ pixels under each microlens in the sensor plane. A pixel location $x$ can be written as $x = kJ + j$, where $j = [j_1 j_2], j_1, j_2 \in [0, \ldots, J - 1]$, and $k = [k_1 k_2]$ with $k_1, k_2 \in \mathbb{Z}$. It should be noted that while all those pixels with the same $j$ correspond to a view (like one image from a camera array), those with the same $k$ correspond to the image within a single microlens. We decompose the coordinates $p$ on the microlens plane in a similar fashion. If two microlenses are $D$ units apart, then we write $p = bD + t$, where $t = [t_1 t_2], t_1, t_2 \in [0, \ldots, D - 1]$, and $b = [b_1 b_2], b_1, b_2 \in \mathbb{Z}$.

Now consider two points $p$ and $q$ located in different microlenses in such a way that their relative position with respect to the microlens center is the same (i.e., both have the same value of $t$). Then, the PSFs $h(x, p)$ and $h(x, q)$ will be shifted versions of each another. Technically, this is because the PSF of the light field image is given by the intersection of the main lens blur disc and the microlens array. A shift of a point light source will cause the main lens blur disc onto the microlens array to shift in the opposite direction. If this shift corresponds exactly to the microlens diameter, then the intersection will be identical up to a shift to the intersection with the initial point light source. This exploits the regularity of the microlens array. Thus, we can write

$$h(x, p) = h(x - Jy, p - Dy)$$

for any $y$, where $y = [y_1 y_2], y_1, y_2 \in \mathbb{Z}$. The first observation is that we only need to store $J^2 \times D^2$ blur kernels, as the others can be obtained by using eq. (3). Secondly, the extent of the blur will be limited (for example, see Fig. 3) so that in practice it is sufficient to use fewer than $J \times J$ microlenses. Thus, we have obtained a memory efficient representation of the PSF $h$. 
We now replace $p$ by $bD + t$ in eq. (1), and use eq. (3) with $y = b$ to get

$$f_m(x) = \sum_t \sum_b h(x - bJ, t)g_m(bD + t).$$

(4)

Then, by expressing $x$ as $x = kJ + j$, we get

$$f_m(kJ + j) = \sum_t \sum_b h(kJ + j - bJ, t)g_m(bD + t).$$

(5)

Let $\hat{g}_m(b, t) \doteq g_m(bD + t)$, $\hat{f}_m(k, j) \doteq f_m(kJ + j)$ and $\hat{h}_j(k, t) \doteq h(kJ + j, t)$ (i.e., just a rearrangement
of the same pixels. Then, for every value of \( j \), we have
\[
\hat{f}_m(k, j) = \sum_t \sum_b h((k - b)J + j, t) \hat{g}_m(b, t)
\] (6)
and finally
\[
\hat{f}_m(k, j) = \sum_t \sum_b \hat{h}_j(k - b, t) \hat{g}_m(b, t) = \sum_t \left( \hat{h}_j(\cdot, t) * \hat{g}_m(\cdot, t) \right)(k)
\] (7)
where \(^*\) denotes the convolution operation. Eq. (6) indicates that we can arrive at the LF image by performing convolutions for every possible value of \( t \) and \( j \). Note that these convolutions are completely independent and therefore can be executed in parallel. In Fig. 3 we illustrate a light field PSF. As can be observed, for each \( t \) the extent of the blur is limited. These small kernels are then used to calculate the convolutions in eq. (7).

IV. LIGHT FIELD MOTION DEBLURRING

Given a motion blurred LF image, our objective is to estimate the sharp scene texture. By using the camera calibration parameters we compute an explicit PSF \( \hat{h} \). Then, to solve for both the motion blur PSF and the latent sharp texture from a single blurred LF image, we jointly estimate the motion blur PSF and latent texture by using an alternating minimization procedure.

Based on the model (7) for \( f_m \) discussed in sec. III the objective function can be written as
\[
\min_{g,h_m} \left\| \sum_p h(x,p) \sum_q h_m(q)g(p - q) - f_m \right\|_2^2 + \lambda \|g\|_{BV}
\] subject to \( h_m \gg 0, \|h_m\|_1 = 1 \) (8)
where \( \|g\|_{BV} = \int \|\nabla g(p)\|_2 dp \), with \( \nabla g = [g_x g_y]^T \), is the total variation of \( g \), and \( \lambda > 0 \) regulates the amount of total variation. The norm \( \| \cdot \|_1 \) corresponds to the \( L^1 \) norm.

To solve the above problem, we use the Total Variation blind deconvolution algorithm of [7]. The optimization is solved via an iterative procedure, where at each iteration \( n \) the current estimate of the sharp texture is given by
\[
\begin{align*}
\bar{g}_m &= h_m^{n-1} * g^{n-1} \\
\bar{f}_m(x) &= \sum_p h(x,p)\bar{g}_m(p) \\
\bar{g}_m^*(q) &= \sum_x h(x,q)(\bar{f}_m(x) - f_m(x)) \\
g^n &= g^{n-1} - \epsilon \left[ \overline{h}_m^{n-1} * \bar{g}_m^* - \lambda \nabla \cdot \frac{\nabla g^n}{\|g^n\|_{BV}} \right]
\end{align*}
\] (9)
for some step \( \epsilon > 0 \) and where \( \overline{h}_m(q) = h_m(-q) \). For the blur kernel \( h_m \) the iteration is
\[
\begin{align*}
\bar{g}_m &= h_m^{n-1} * g^n \\
\bar{f}_m(x) &= \sum_p h(x,p)\bar{g}_m(p) \\
\bar{g}_m^*(q) &= \sum_x h(x,q)(\bar{f}_m(x) - f_m(x)) \\
\overline{h}_m^{n-3/2} &= h_m^{n-1} - \epsilon [\bar{g}_m^* * \bar{g}_m^*]
\end{align*}
\] (10)
The last updated \( \bar{h}_m^{n-3/2} \) is used to set \( h_m^n \) by using the following sequential projections
\[
\begin{align*}
h_m^{n-1/3} &\leftarrow \max\{\bar{h}_m^{n-2/3}, 0\} \\
h_m^n &\leftarrow \frac{h_m^{n-1/3}}{\|h_m^{n-1/3}\|_1}
\end{align*}
\] (11)
Notice that all steps can be computed very efficiently via convolutions as presented in sec. III-A. Furthermore, in our implementation we also use a pyramid-based scheme which facilitates faster convergence.
### TABLE I

In this table we show a comparison between the methods for two different levels of noise. We show the average ($\mu$) and the standard deviation ($\sigma$) of the PSNR and SSIM metrics for 8 synthetically generated motion blurred light fields.

|                | 0%                      | 2.5% noise              |
|----------------|-------------------------|-------------------------|
|                | PSNR one-step | PSNR two-step | PSNR Initial | SSIM one-step | SSIM two-step | SSIM Initial |
| $\mu$          | 33.39        | 31.53        | 29.13       | 0.960        | 0.898        | 0.883        |
| $\sigma$       | 2.39         | 5.7912       | 4.2665      | 0.028        | 0.191        | 0.103        |
|                | 29.07        | 22.98        | 28.67       | 0.911        | 0.666        | 0.872        |

**V. EXPERIMENTAL RESULTS**

We evaluate our light field blind deconvolution algorithm on several synthetic and real experiments. For synthetic experiments, we artificially generate motion blurred light fields by assuming scene texture, depth, camera parameters, and motion blur kernel. We perform real experiments using images of motion-blurred scenes captured with a plenoptic camera prototype developed in [4].

We generate 8 motion blurred light fields (from the dataset of Levin et al. [21]) with 0% and 2.5% additive zero mean Gaussian noise, and we compare our proposed method (we call it one-step in the figures and tables) to two additional cases: 1) Recover the texture from the light field by ignoring motion blur (we call it Initial); 2) Apply a standard blind deconvolution method to the texture recovered in Initial (we call it two-step). For the noise-free experiments we set the parameters $\lambda = 4 \cdot 10^{-4}$ for both methods, while for the experiment with noisy images we set $\lambda = 10^{-3}$ (all input light fields are rescaled between 0 and 1). In our comparison we used the Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity Index (SSIM) [33] metrics. Since between the blur function and the sharp image there is a translational ambiguity, for each image we take the maximum PSNR and SSIM among all the possible shifts between the estimate and ground truth image. In Table I we show the mean and standard deviation PSNR and SSIM for the one-step, two-step, and for the Initial method results. Both one-step and two-step methods show an improvement over the Initial image, with the one-step method outperforming the two-step method for both metrics. The two-step method has a larger standard deviation due to some failures for particularly difficult blurry images. On the contrary, the one-step has a small standard deviation and therefore is more consistent and robust. In Fig. 4 we show a more detailed statistical analysis of the results shown in Table I. The use of the median and of the Median Absolute Deviation (MAD) allows a comparison between the two different methods without excessively considering outliers. Nonetheless, the one-step method still results in a better overall performance. The benefits of our one-step deblurring approach become clearer when the input images are noisy. In Table I and in Fig. 4 we show how the one-step method is able to cope with noisy images without a significant loss of accuracy. Conversely, the two-step approach has an unacceptable drop in both average PSNR and SSIM. Fig. 5 reveals the main reason for the poor performance of the two-step method: the motion deblurring step enhances both the sharpness of the noise and the sharpness of the true image texture, and therefore results in a worse overall performance. By taking advantage of the redundancy given by the light field image the one-step method reduces both the noise and the motion blur effects. In Fig. 6 to 9 we show additional synthetic results. Fig. 6 and 8 use noise-free LFs. Fig. 7 and 9 use noisy LFs with 2.5% additive Gaussian noise. Notice how the one-step approach can achieve better robustness to outliers and noise than the two-step approach.
Fig. 4. Statistics on synthetic deblurring experiments (it is recommended that images are viewed in color). We show with a blue star the mean PSNR and SSIM and with blue brackets the standard deviation from the mean. A red dot denotes the median and the red brackets the median absolute deviation (MAD) from the median. We denote outliers with a black dot.

Fig. 5. Example of results with noisy input images.

Fig. 6. Texture reconstruction from synthetic light field image. a) LF image. b) Initial image. c) two-step algorithm deblurred image. d) one-step algorithm deblurred image. e) Ground truth sharp image.
Fig. 7. Texture reconstruction from synthetic light field image with 2.5% additive noise. a) LF image. b) Initial image. c) two-step algorithm deblurred image. d) one-step algorithm deblurred image.

Fig. 8. Texture reconstruction from synthetic light field image. a) LF image. b) Initial image. c) two-step algorithm deblurred image. d) one-step algorithm deblurred image. e) Ground truth sharp image.
Fig. 9. Texture reconstruction from synthetic light field image with 2.5% additive noise. a) LF image. b) Initial image. c) two-step algorithm deblurred image. d) one-step algorithm deblurred image.

A. Real experiments

In our real experiments, we use a plenoptic camera built from a Hasselblad H2 medium format camera by placing an array of microlenses near the image sensors. The CCD array of the digital back is of size 4096×4096 pixels with each pixel of size 9µm. The microlens array consists of circular lenses with a rectangular grid structure (in contrast to Lytro or Raytrix, which have a hexagonal grid). The spacing between consecutive microlens centers is 135µm. We use an 80 mm f/2.8 lens. All our images were captured with f-number 11 so as to match the f-number of the microlens array lenslets. Due to imperfections in our imaging setup, there is both geometric and photometric distortion of data. We rectify these through a simple calibration procedure.

Fig. 10 (a) shows an example of an image captured by our camera of a computer screen onto which the Matlab peppers.png image was being displayed. The misalignment can be easily noticed in Fig. 10 (c) (which shows a zoomed-in region) wherein lines formed due to the gap between two rows of microlenses appear to have a non-zero slope instead of being strictly horizontal. For calibration, we manually select four points that should form a rectangle on the image plane. Using the coordinates of these points, we solve for the affine transformation. Note that this manual selection of points needs to be done only once. We found that the raw LF image consisted of 15.225 pixels per microlens. For convenience, we resize the image so that every microlens has 15×15 pixels under it (i.e., Q = 15). The resized calibrated image is shown in Fig. 10 (b). Fig. 10 (d), shows a zoomed region from the calibrated image. From Figs. 10 (c) and (d), we see that the misalignments are rectified by our calibration. For photometric correction, we capture an image of a scene with only a white background. This white light field is used to normalize all captured LF images. A LF image is normalized by dividing the captured LF by the white LF. This procedure tends to compensate for the darkening of pixels at the boundary of the micro lenses and causes the pixels at the gaps between the micro lenses to become whiter.

In Figs. 11, 112 and 13 we show deblurring results on images captured by our LF camera. We compare the reconstruction obtained with the one-step approach to that of the two-step approach. In all the figures we show the calibrated light field (a portion of the full LF), the reconstructed image of the Initial approach, the reconstructed image of the two-step approach, the reconstructed image from the one-step approach and one real in-focus image of the object when motionless (for Figs. 11 and 1 only). Note that small distortions are observed in images due to sensor dust.

VI. Conclusions

We have presented the first motion deblurring method for plenoptic cameras. Our method extends classical blind deconvolution methods to light field cameras by modeling the interaction of motion blur and defocus point spread functions. Moreover, our model is highly parallelizable and memory efficient. To solve blind deconvolution we propose an alternating minimization procedure that estimates a sharp and
Fig. 10. Illustration of calibration. a) Raw image. b) Calibrated image. c) Cropped region of the raw image. d) A region of the calibrated image.

Fig. 11. Texture reconstruction from a real light field image. a) LF image. b) Image estimated with Initial method. c) Two-step algorithm deblurred image. d) One-step algorithm deblurred image. e) Sharp image captured with a (static) conventional camera.

In this paper, we have studied the case of uniform blur. In future work, we also plan to extend this model to the case of space-varying blur, so that depth variations can be handled. This case, however, will introduce formidable computational challenges as the fast model in sec. III-A will not longer hold.
Fig. 12. Texture reconstruction from a real light field image. a) LF image. b) Image estimated with Initial method. c) Two-step algorithm deblurred image. d) One-step algorithm deblurred image.

Fig. 13. Texture reconstruction from a real light field image. a) LF image. b) Image estimated with Initial method. c) Two-step algorithm deblurred image. d) One-step algorithm deblurred image.

REFERENCES

[1] Lytro, inc. http://www.lytro.com/
[2] Raytrix. http://www.raytrix.de/
[3] E. H. Adelson and J. Y. A. Wang. Single lens stereo with a plenoptic camera. IEEE Trans. Patt. Anal. Mach. Intell., 14(2):99–106, 1992.
[4] T. Bishop and P. Favaro. The light field camera: extended depth of field, aliasing and superresolution. IEEE Trans. Patt. Anal. Mach. Intell., 34(5):972–986, 2012.
[5] Charles A. Bouman, Ilya Pollak, and Patrick J. Wolfe, editors. Superresolution with the focused plenoptic camera, volume 7873, San Francisco, CA, 02/07/2011 2011. SPIE. doi: 10.1117/12.872666.
[6] Michael Brixton, Logan Grosenick, Samuel Yang, Noy Cohen, Aaron Andalman, Karl Deisseroth, and Marc Levoy. Wave optics theory and 3-d deconvolution for the light field microscope. Opt. Express, 21(21):25418–25439, Oct 2013. doi: 10.1364/OE.21.025418. URL http://www.opticsexpress.org/abstract.cfm?URI=oe-21-21-25418
[7] T.F. Chan and C.-K. Wong. Total variation blind deconvolution. IEEE Transactions on Image Processing, 7(3):370–375, 1998.
[8] Donghyeon Cho, Minhaeng Kim, Lee Sunyeong, and Yu-Wing Tai. Decoding, calibration and rectification for lenselet-based plenoptic cameras. In Proc. ICCV, 2013.
[9] S. Cho and S. Lee. Fast motion deblurring. ACM Trans. Graph., 28(5):1–8, 2009a. ISSN 0730-0301.
[10] Donald G Dansereau, Oscar Pizarro, and Stefan B Williams. Decoding, calibration and rectification for lenselet-based plenoptic cameras. In Proc. CVPR, 2013.
[11] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. ACM Trans. Graph., 25(3):787–794, 2006a. ISSN 0730-0301.
[12] B. Goldluecke, M. Aubry, K. Kolev, and D. Cremers. A super-resolution framework for high-accuracy multiview reconstruction. IJCV, 2013.
[13] A. Gupta, N. Joshi, L. Zitnick, M. Cohen, and B. Curless. Single image deblurring using motion density functions. In Proc. ECCV, 2010.
[14] Stefan Heber, Rene Ranfyl, and Thomas Pock. Variational shape from light field. In Energy Minimization Methods in Computer Vision and Pattern Recognition, volume 8081, pages 66–79. 2013. ISBN 978-3-642-40394-1.
[15] M. Hirsch, C. J. Schuler, S. Harmeling, and B. Scholkopf. Fast removal of non-uniform camera shake. In Proc. ICCV, 2011.
[16] Z. Hu and M. Yang. Fast non-uniform deblurring using constrained camera pose subspace. In Proc. BMVC, 2012.
[17] H. Ji and K. Wang. A two-stage approach to blind spatially-varying motion deblurring. In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, pages 73–80, june 2012. doi: 10.1109/CVPR.2012.6247660.
[18] Tae Hyun Kim, Byeongjoo Ahn, and Kyoung Mu Lee. Dynamic scene deblurring. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2013.

[19] R. Köhler, M. Hirsch, B. J. Mohler, B. Schölkopf, and S. Harmeling. Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database. In *ECCV* (7), pages 27–40, 2012.

[20] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *Proc. CVPR*, 2009.

[21] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding blind deconvolution algorithms. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(12):2354–2367, 2011a.

[22] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Efficient marginal likelihood optimization in blind deconvolution. In *CVPR*, pages 2657–2664, 2011b.

[23] Marc Levoy and Pat Hanrahan. Light field rendering. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 31–42. ACM, 1996.

[24] A. Lumsdaine and T. Georgiev. Full resolution lightfield rendering. Technical report, Adobe Systems, 2008.

[25] Ren Ng, Marc Levoy, Mathieu Brédif, Gene Duval, Mark Horowitz, and Pat Hanrahan. Light field photography with a hand-held plenoptic camera. *Computer Science Technical Report CSTR*, 2(11), 2005.

[26] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. *ACM Transactions on Graphics*, 27(3).

[27] Q. Shan, W. Xiong, and J. Jia. Rotational motion deblurring of a rigid object from a single image. In *ICCV*, pages 1–8, 2007. URL http://dx.doi.org/10.1109/ICCV.2007.4408922.

[28] A. Srivastava, A.B. Lee, E. P. Simoncelli, and S.-C. Zhu. On advances in statistical modeling of natural images. *Journal of Mathematical Imaging and Vision*, 18:17–33, January 2003. ISSN 0924-9907.

[29] Libin Sun, Sunghyun Cho, Hue Wang, and James Hays. Edge-based blur kernel estimation using patch priors. In *Proc. IEEE International Conference on Computational Photography*, 2013.

[30] Y. Tai, P. Tan, and M. S. Brown. Richardson-lucy deblurring for scenes under projective motion path. *IEEE Trans. Patt. Anal. Mach. Intell.*, 33(8):1603–1618, 2011.

[31] Vaibhav Vaish, Bennett Wilburn, Neel Joshi, and Marc Levoy. Using plane+ parallax for calibrating dense camera arrays. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–2. IEEE, 2004.

[32] Kartik Venkataraman, Dan Lelescu, Jacques Duparre, Andrew McMahon, Gabriel Molina, Priyam Chatterjee, Robert Mullis, and Shree Nayar. Picam: an ultra-thin high performance monolithic camera array. *ACM Trans. Graph.*, 32(5):1–8, 2013. ISSN 0730-0301.

[33] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4):600–612, April 2004. ISSN 1057-7149. doi: 10.1109/TIP.2003.819861.

[34] S. Wanner and B. Goldluecke. Richardson-lucy deblurring for scenes under projective motion path. *IEEE Trans. Patt. Anal. Mach. Intell.*, 33(8):1603–1618, 2011.

[35] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. In *Proc. CVPR*, 2010.

[36] L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. In *ECCV*, 2010a.

[37] L. Xu, S. Zheng, and J. Jia. Efficient filter flow for space-variant multiframe blind deconvolution. In *CVPR*, 2010.

[38] Y.-L. You and M. Kaveh. A regularization approach to joint blur identification and image restoration. *Image Processing, IEEE Transactions on*, 5(3):416–428, 1996a. ISSN 1057-7149. doi: 10.1109/83.491316.

[39] Y.-L. You and M. Kaveh. Anisotropic blind image restoration. In *Image Processing, 1996. Proceedings., International Conference on*, volume 1, pages 461–464 vol.2, 1996b. doi: 10.1109/ICIP.1996.560885.