Bilayer Blind Deconvolution with the Light Field Camera

Meiguang Jin
Institute of Informatics
University of Bern
Switzerland
jin@inf.unibe.ch

Paramanand Chandramouli
Institute of Informatics
University of Bern
Switzerland
chandra@inf.unibe.ch

Paolo Favaro
Institute of Informatics
University of Bern
Switzerland
paolo.favaro@inf.unibe.ch

Abstract

In this paper we propose a solution to blind deconvolution of a scene with two layers (foreground/background). We show that the reconstruction of the support of these two layers from a single image of a conventional camera is not possible. As a solution we propose to use a light field camera. We demonstrate that a single light field image captured with a Lytro camera can be successfully deblurred. More specifically, we consider the case of space-varying motion blur, where the blur magnitude depends on the depth changes in the scene. Our method employs a layered model that handles occlusions and partial transparencies due to both motion blur and out of focus blur of the plenoptic camera. We reconstruct each layer support, the corresponding sharp textures, and motion blurs via an optimization scheme. The performance of our algorithm is demonstrated on synthetic as well as real light field images.

1. Introduction

In the last decade, there has been a considerable effort towards solving blind deconvolution with conventional cameras [9, 28, 36, 11, 20]. Most solutions apply to scenes that can be well approximated with a plane, *i.e.*, when imaging objects at a distance or when the camera rotates about its center. However, when the depth between two objects in the scene becomes apparent, these methods produce visible artifacts. One approach is to formulate the task as an optimization problem with an explicit model for occlusions (*e.g.*, with an alpha matting model) and where depth and the object support are reconstructed together with their sharp texture and motion blur. Unfortunately, as discussed in section 4.3, a simple statistical analysis reveals that convergence to the optimal solution is difficult for this formulation. The evaluation instead reveals that when using a single image from a *light field camera* the depth layer support can converge to the optimal value. This motivates us to consider using this device for addressing blind deconvolution when depth variations are significant. Moreover, we are not aware of any method for solving blind motion deblurring in light field (LF) cameras.

Blind deconvolution techniques developed for conventional cameras cannot be directly applied to LF images, because the mechanism of image formation of a LF image differs from that of a conventional one. Due to the microlens array present between the camera main lens and the sensors, the captured image consists of repetitive and/or blurry patterns of the scene texture. Moreover these patterns depend on the camera settings and vary with depth. See Fig. 1 (a) for an example of a real motion-blurred LF image. The problem is further exacerbated by the fact that there could be variations in motion blur across the image due to depth changes. A possible approach could be to extract angular views from the LF image and apply space-varying deblurring on each view separately. However, this approach is hampered by aliasing (due to undersampling of the views) and would yield low-resolution images which cannot be easily merged in a single high resolution image.

In this paper, we consider a global optimization task where all the unknowns are simultaneously recovered by using all the information (the LF image) at once and by applying regularization directly to all the unknowns. Our objective is to recover a sharp high resolution scene texture from a motion blurred LF image. However, due to depth variations the textures on objects at different depths merge in the captured LF image. Thus, we consider a layered representation of the scene and explicitly model this blending effect via an LF *alpha matting*. We then reconstruct a sharp texture for each layer (alpha matte) and then compose them in a single sharp image via the recovered alpha mattes. We consider that the camera motion is translational on the X and Y axes. Thus, motion blurs on each layer will be related to each other via a scale factor. To speed up our algorithm and to avoid local minima in this complex optimization task, we initialize the layers by first estimating a depth map directly from the blurred light field and by discretizing the layers. Then, we recover an initial blurry texture by undoing

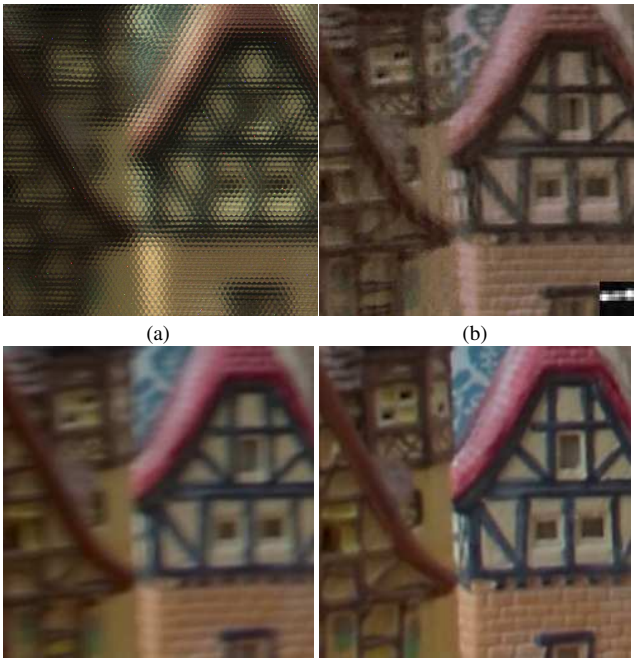


Figure 1. (a) Motion blurred LF image (zoom-in to see the microlenses). b) Recoverd texture (estimated motion blur in the insert at the bottom-right). c) Blurred images (from Lytro sharp image generation software). d) Image of the same scene without motion blur (from Lytro).

the LF image formation. Finally, we cast the optimization task with respect to all variables in a variational formulation which we minimize via alternating minimization. Although in this paper we consider only bilayer scenes, our model can be extended to more general cases.

2. Related work

We briefly discuss prior works related to motion deblurring and light field imaging.

Conventional motion deblurring Motion deblurring, the problem of jointly estimating a motion blur kernel and a sharp image is an ill-posed problem [38]. The case when blur is the same throughout the image, has been widely studied and impressive performance has been achieved by recent algorithms [9, 11, 28, 36, 21]. To handle the ill-posedness of the problem, these methods enforce priors on the image as well as the blur kernel [20, 6]. For more details on different approaches to the blind deconvolution problem, we refer the reader to recent papers such as [30, 26, 35] and the references therein.

When camera motion includes camera rotations, the blur kernel varies across an image. Approaches based on blind deconvolution have been adapted to handle such scenarios [31, 34, 12, 16, 18, 25] by including additional dimensions in the blur representation. These methods are typically more computationally demanding and the improvements over shift-invariant deblurring are limited [19]. In 3D scenes, motion blur at a pixel is also related to depth at a point. Techniques proposed in [37, 29] handle variation of motion blur due to depth changes when camera motion is restricted to in-plane translations. In [25] non-uniform motion blur is considered for bilayer scenes. However, the authors use two differently motion blurred observations, instead of one as in this work. The closest work to ours is [17] wherein the authors use a single motion blurred image as in our case. They model camera shake by in-plane translations and rotations and use a layered representation for the scene. The fundamental difference with our work is that they use a conventional camera. Thus, as shown in section 4.3, the support of depth layers cannot be reconstructed via optimization. Indeed, [17] relies on user interaction (via scribbles in the alpha matting step) while our method is fully automatic.

Plenoptic cameras, camera arrays and calibration. Light fields can be acquired by either microlens array-based plenoptic cameras, or through camera arrays. An important difference between camera arrays and plenoptic cameras is that while the spatial resolution is high in camera arrays, angular resolution is low. In the case of the plenoptic cameras, the opposite holds. For brevity, we concentrate our discussion on plenoptic cameras. Adelson and Wang developed the first plenoptic camera in computer vision by placing a lenticular array at the sensors [1]. Their objective was to estimate depth from a single image. The use of microlens arrays to capture LF image by Ng et al. [24] gave rise to portable camera designs. To overcome the limitation of spatial resolution, techniques for super-resolving the data up to the order of full sensor resolution have since been proposed [23, 4, 2]. While in [23], the spatial resolution is improved using information encoded in the angular domain, in [4] demosaicing is incorporated as a part of the reconstruction process. Bishop and Favaro [2] use an explicit image formation model to relate scene depth and high resolution texture. They follow a two-step approach to achieve super-resolution through a variational framework. Work of Broxton et al. [5] also shows a fast computational scheme for LF generation with an explicit point spread function.

Recently, techniques that demonstrate their applicability on Lytro and Raytrix cameras have been proposed. Cho et al. [7] develop a method for reconstructing a high resolution texture after rectification and decoding of raw data. Danserau et al. [10] and Bok et al. [3] propose calibration schemes to estimate camera parameters that relate a pixel in the image to rays in 3D space. Sabater et al. [27] propose a depth estimation method that uses angular view correspondences and also avoids cross talk due to demosaicking. Tao et al. [32] propose to combine correspondence and defocus cues in a light field image for depth estimation. Heber et

al. [15] propose a depth estimation scheme motivated by the idea of active wavefront sampling. Using a continuous framework, Wanner and Goldlücke [33] develop variational methods for estimating disparity as well as for spatial and angular super-resolution. Other recent works of interest in light field imaging include estimation of scene flow [14] and alpha matte [8]. An interesting theoretical analysis of the performance of light field cameras has been recently presented in [22] using the light transport framework.

Contributions. The contributions of our work in contrast to the above mentioned prior work can be summarized as:

1. This is the first attempt in blind deconvolution of light field images.

2. Our LF image formation model is the first to take into account the effect of camera optics on depth as well as variations of motion blur due to depth changes.

3. We handle occlusion boundaries at depth discontinuities in LF images.

4. We solve for the scene depth map, occlusion boundaries, super-resolved texture and motion blur within a variational framework. No user interaction is required.

3. Imaging model

In this section, we introduce the notation and describe our approach to model a motion blurred light field image. We consider that the 3D scene is composed of two depth layers. Initially, we consider a single depth layer scenario and subsequently extend our model to two layers.

3.1. Single layer model

Following the approach in [2, 5], we relate the light field image l formed at the image sensor plane to the scene texture f through a point spread function (PSF). For convenience, the texture f is defined at the microlens array plane. Let $\mathbf{u} = [u_1\ u_2]^T$ denote the discretized coordinates of a point on the microlens array plane and $\mathbf{x} = [x_1\ x_2]^T$ denote a pixel location. A space-varying PSF $P_{s(\mathbf{u})}(\mathbf{x}, \mathbf{u})$ relates the LF image and scene texture via

$$l(\mathbf{x}) = \sum_{\mathbf{u}} P_{s(\mathbf{u})}(\mathbf{x}, \mathbf{u}) f(\mathbf{u}). \tag{1}$$

where the PSF $P_s(\mathbf{u})$ depends on the scene depth $s(\mathbf{u})$ as well as the camera parameters. When the camera parameters and the scene depth are known, the entries of the matrix P_s can be explicitly evaluated [2, 5]. The evaluation of P_s involves finding the intersection between the blur circles generated by the main lens and the microlens array due to a point light source in space [2]. For convenience, we abuse the notation to denote vectorial representations of LF image and scene texture by l and f, respectively and a matrix version of PSF by P_s. The LF image generation is then expressed as a matrix vector product

$$l = P_s f. \tag{2}$$

Typically, the LF image will be of the order of megapixels and the texture resolution would be a fraction (say $\frac{1}{3}$) of that of the LF image. Consequently, the matrix P_s would turn out to be too large for practical computations. However, the intersection pattern between the main lens and microlens blur circles is repetitive resulting in the periodicity of PSF along the domain of microlens array plane. Consequently, by finding the PSF for texture elements corresponding to only one microlens, one has enough information about the whole PSF. This property enables one to express the LF image generation in Eqn. (1) as a summation of convolutions between a set of rearranged components of f with the components of light field PSF. These convolutions can be implemented in parallel [5]. We would like to point out that throughout the paper, the matrix vector products of the type in Eqn. (2) are implemented as such sums of convolutions.

Due to relative motion between the camera and scene, if the texture undergoes motion blur, one could express the light field image as

$$l = P_s M f \tag{3}$$

where M denotes a matrix representing the motion PSF.

3.2. Bilayer model

A naïve approach to model bilayer scenes would be to superpose components of motion blurred light field images from each layer separately. However, this model causes artifacts at depth boundaries even when synthesizing an LF image. We propose a more realistic, and still computationally simple, model of bilayer scenes by considering occlusion effects via an extension of the alpha matting model of Hasinoff and Kutulakos [13].

Initially, we discuss the model by neglecting the effect of motion blur. Let s denote the scene depth map defined on the same domain as the texture f. We assume that s takes 2 distinct values s_1, s_2. The region corresponding to the smallest depth s_1 is considered as the first support Ω_1, *i.e.*,

$$\Omega_1(\mathbf{u}) = \begin{cases} 1 & \text{if } s(\mathbf{u}) = s_1 \\ 0 & \text{if } s(\mathbf{u}) \neq s_1. \end{cases}$$

The second layer support is defined to be unity. Thus, in general, we define layer supports such that

$$\Omega_i(\mathbf{u}) = \begin{cases} 1 & \text{if } s(\mathbf{u}) \leq s_i \\ 0 & \text{if } s(\mathbf{u}) > s_i. \end{cases}$$

Since all depth values are less than or equal to s_2, the second layer Ω_2 is always defined to be 1 and is never estimated

or shown in the figures. For a depth layer i with support function Ω_i, we define a function α_i as

$$\alpha_i = P_i \Omega_i, \tag{4}$$

where P_i is the LF PSF for layer i (with depth s_i).

The LF image l can be expressed as a weighted sum of contributions from each depth layer having texture f_i, *i.e.*,

$$l = \beta_1 \odot P_1 f_1 + \beta_2 \odot P_2 f_2 \tag{5}$$

and the weights β_i are given by

$$\begin{aligned} \beta_1 &= \alpha_1 \\ \beta_2 &= \alpha_2 \odot (1 - \alpha_1) \end{aligned} \tag{6}$$

where $\odot$ denotes the Hadamard product (element by element product).

When there is relative motion between the camera and scene, the texture of a depth layer as well as its support undergo a translation. Let M_1 denote the motion blur for the first layer. We consider M_1 to be the reference blur as it is the largest (motion blur decreases with distance). Since we restrict the camera motion to translations alone, the blur at the second layer M_2 would be a scaled version of the reference blur where the scale factor is given by the depth ratio [29]. We can now express the blurred light field image l^b using the following equations:

$$\begin{aligned} &\beta_i^b = \alpha_i^b \odot \prod_{k=1}^{i-1} (1 - \alpha_k^b), \quad \alpha_i^b = P_i \Omega_i^b, \quad \Omega_i^b = M_i \Omega_i \\ &l^b = \beta_1^b \odot l_1^b + \beta_2^b \odot l_2^b, \quad l_i^b = P_i f_i^b, \quad f_i^b = M_i f_i. \end{aligned} \tag{7}$$

Notice that due to relative motion, the layers also move, and hence we need to introduce motion blurred layers Ω_i^b. In Fig. 4, we give an example of different components in the imaging model by assuming that the scene depth is as shown in Fig. 3 (d) (the chosen motion blur PSF is shown in the inset of Fig. 5 (d)).

4. Light field motion deblurring

Given a motion blurred LF image l_o, we initially estimate its depth map s by establishing correspondences across different views present in the LF image. We then quantize the depth map to 2 levels to arrive at a discrete depth map that takes values from the set $\{s_1, s_2\}$.

4.1. Depth estimation

Our depth estimation scheme is based on exploiting the correspondences across views within an LF image. We estimate the depth map s at the same resolution at which f is defined and assume that the scene is Lambertian (as in traditional stereo methods). Suppose that a texture element at a point $\mathbf{u}$ is imaged by microlenses with centers $\mathbf{c}_i$. Then, the corresponding angular index θ_i in the sub image corresponding to the microlens with center $\mathbf{c}_i$ is given by

$$\theta_i = \Lambda(\mathbf{u})(\mathbf{c}_i - \mathbf{u}). \tag{8}$$

The term $\Lambda(\mathbf{u})$ is also called the magnification factor and is related to scene depth $s(\mathbf{u})$ via

$$\Lambda(\mathbf{u}) = \frac{v}{v' - z'} \frac{z'}{v'} \quad \text{with} \quad \frac{1}{z'} = \frac{1}{F} - \frac{1}{s\,(\mathbf{u})} \tag{9}$$

where F denotes the camera focal length, v denotes the distance between the microlens array plane and image sensors, and v' is the distance between the main lens and microlens array plane [2].

In our depth estimation algorithm, the magnification Λ is analogous to a disparity map in stereo. We use a plane sweep approach and select a set of depth values which are then mapped to the magnification via Eqn. (9). For each magnification value we use Eqn. (8) to determine all possible correspondences $(\mathbf{c}_i, \theta_i)$ with $i \in \mathcal{N}(\mathbf{u})$, where $\mathcal{N}(\mathbf{u})$ is the set of immediate neighboring microlenses around the pixel $\mathbf{u}$. Firstly, we determine the closest microlens $\mathbf{c}_0$ to the coordinate $\mathbf{u}$ and find the corresponding θ_0 (the 2D coordinate local to a microlens). We then compute a matching cost associated to the values of the LF image at these pixels

$$E_0(\mathbf{u}, s_j) = \sum_{i \in \mathcal{N}(\mathbf{u})} \frac{|l(\mathbf{c}_i + \theta_i) - l(\mathbf{c}_0 + \theta_0)|}{l(\mathbf{c}_i + \theta_i) + l(\mathbf{c}_0 + \theta_0)} \tag{10}$$

where $\Lambda(\mathbf{u})$ has been computed with $s(\mathbf{u}) = s_j$. We then convexify the matching cost E_0 along the depth axis by taking its lower convex envelopes. Finally, we estimate an initial depth map by solving a regularized (convex) minimization algorithm of the form

$$\hat{s} = \arg\min_s \mu \sum_{\mathbf{u}} E_0(\mathbf{u}, s(\mathbf{u})) + |\nabla s|_2 \tag{11}$$

where $\mu > 0$ is a constant that defines the amount of regularization and the last term is the total variation of the depth s. Notice that the depth map maps to the real line, and hence it is necessary to interpolate the matching cost E_0 during the minimization. The cost is minimized by a simple gradient descent. Finally, we discretize the estimated depth map by selecting two modes from its histogram to arrive at the values s_1 and s_2.

4.2. Alternating minimization scheme

We follow an energy minimization approach to estimate the scene texture and the motion blur at each depth layer. From the discretized depth map, we initialize the supports Ω_1, Ω_2. We then refine the supports because our depth estimation process could have errors. Errors may be caused

by mismatches due to motion blur in the LF image. Based on the image formation model in Eqn. 7, the data term $E(f_i, M_i, \Omega_i)$ can be written as

$$\Big| (P_1(M_1\Omega_1)) \odot (P_1(M_1 f_1)) + (1 - P_1(M_1\Omega_1)) \odot (P_2(M_2)) \odot (P_2(M_2 f_2)) - l_0 \Big|^2 \quad (12)$$

where l_0 is the measured LF image. To handle the ill-posedness of the problem, we also incorporate isotropic total variation regularization for both texture and support. We also enforce that the blur kernel for the second layer is consistent with the reference blur kernel M_1. Thus, the energy functional to be minimized is given by

$$J(f_i, M_i, \Omega_i) = E(f_i, M_i, \Omega_i) + \sum_{i=1}^{2} \lambda_f |\nabla f_i|_2 + \lambda_\Omega |\nabla \Omega_1|_2 + \lambda_M |D_2 \text{vec}(M_1) - \text{vec}(M_2)|^2 \quad (13)$$

where $\lambda_f, \lambda_\Omega$, and λ_M are regularization parameters for texture, support and motion blur, respectively. The operator vec$(\cdot)$ denotes the mapping of the motion blur in matrix form to a vector with its entries in lexicographical order. The matrix D_2 down-scales the reference blur M_1 by a factor corresponding to the 2nd depth value. The cost function in Eqn. 13 is minimized using gradient descent. We follow an alternating minimization approach to update each layer of texture f_i, support Ω_i, and motion blur M_i. The gradients of the energy E with respect to the texture f_i, the support Ω_i and the motion blur M_i are given in Table 1.

4.3. Feasibility of support estimation

We perform a simple statistical analysis to check whether the data cost $E(f_i, M_i, \Omega_i)$ in Eqn. (12) can be minimized with respect to Ω_1. We synthetically generate l_0 by selecting realistic values for the variables f_1, f_2, P_1, P_2, M_1, M_2, and Ω_1. We add Gaussian noise to the true value of Ω_1 to arrive at Ω_1^n. When Ω_1^n is considered as the current estimate, the noise, and the gradient of energy with respect to Ω_1^n correspond to the terms Δ and δ, respectively in Fig. 2 (left). We evaluate the inner product between Δ and δ at 2,500 random samples around the exact solution. We also repeat this process for the scenario of conventional camera. i.e., by neglecting the effect of the LF PSFs P_1 and P_2. The plot in Fig. 2 (right) shows the unnormalized distribution (ordinate) of inner products (abscissa). The distribution of inner products of the conventional camera shows that the gradients are equally distributed between the negative and positive side of the abscissa. This means that a gradient descent would move randomly towards or away from the correct minimum, a behavior that denotes ambiguities in the solution and the lack of a valley structure. In contrast, the distribution of the inner products of the LF camera shows a clear preference for the positive side of the abscissa, thus moving towards the correct direction. Also, notice that the inner products tend to be very small. This means that the gradient descent would converge very slowly to the correct solution, a behavior that we also observe in our experiments.

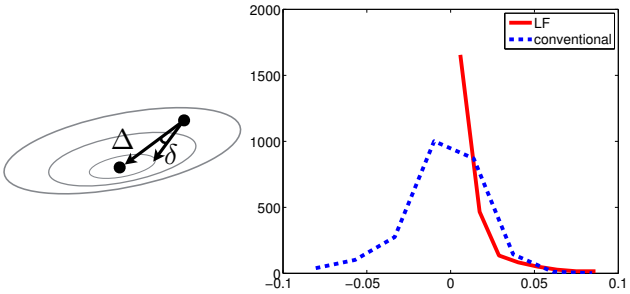


Figure 2. **Left**: illustration of the stochastic analysis. The three ellipses denote the level curves of a cost function. The dot in the middle of the smallest ellipse denotes a local minimum. In this scenario, the gradient vector δ at samples in the vicinity of the local minimum should form angles of less than 90 degrees with the ideal vector Δ connecting the sample to the local minimum. **Right**: stochastic evaluation of the cost functions in the case of a light field camera (red solid) and in the case of a conventional camera (blue dashed).

(a) (b) (c) (d)

Figure 3. Depth maps used in synthetic experiments.

5. Experimental results

We tested our method on synthetic as well as real experiments. In our synthetic experiments, we artificially simulate space-varying motion blurred LF images assuming rectangular as well as hexagonal arrangement of microlens arrays. We consider the depth maps to have different arrangements of the layer boundaries as shown in Figs. 3 (a)-(d). We performed nine experiments by randomly combining images, kernels and depth maps. For the simulation, we use the scene texture and motion blur kernels from the dataset in [21]. While we resize the scene texture to be of size 200×200, the motion blur kernels are resized to 7×7. Apart from the motion blur kernels in [21], we additionally include a PSF with three impulses that are spread to the corners of the support of the PSF. We generate an LF image, according to our model in Eqn. 7. We use a 40×40 microlens array, and assume realistic values for the camera settings, similar to those used in the real experiments (see Table 3), and scene depths between 45 to 110 cm.

A representative example of our synthetic experiment for

Table 1. Summary of all the gradients.

$$
\begin{aligned}
\Delta \doteq & (P_1(M_1\Omega_1)) \odot (P_1(M_1 f_1)) + (1 - P_1(M_1\Omega_1)) \odot (P_2(M_2)) \odot (P_2(M_2 f_2)) - \iota_0 \\
\frac{\partial E}{\partial f_1} = & M_1^T P_1^T \left((P_1(M_1\Omega_1)) \odot \Delta \right) \\
\frac{\partial E}{\partial f_2} = & M_2^T P_2^T \left((1 - P_1(M_1\Omega_1)) \odot (P_2(M_2\Omega_2)) \odot \Delta \right) \\
\frac{\partial E}{\partial \Omega_1} = & M_1^T P_1^T \left(\left(P_i(M_i f_i) - (P_2(M_2 f_2)) \odot (P_2(M_2\Omega_2)) \right) \odot \Delta \right) \\
\frac{\partial E}{\partial M_1} = & \Omega_1^T \left(P_1^T \left(\left(P_1(M_1 f_1) - (P_2(M_2 f_2)) \odot (P_2(M_2\Omega_2)) \right) \odot \Delta \right) \right) + f_1^T \left(P_1^T \left((P_1(M_1\Omega_i)) \odot \Delta \right) \right) \\
\frac{\partial E}{\partial M_2} = & \Omega_2^T \left(P_2^T \left(\left(1 - P_1(M_1\Omega_1) \right) \odot \left(P_2(M_2 f_2) \right) \odot \Delta \right) \right) + f_2^T \left(P_2^T \left(\left(1 - P_1(M_1\Omega_1) \right) \odot \left(P_2(M_2\Omega_2) \right) \odot \Delta \right) \right)
\end{aligned}
$$

bilayer rectangular arrays is shown in Fig. 5. The ground truth image and the reference motion blur kernel (insert at bottom-right) are shown together in Fig. 5 (d). For visual comparison, we show the image obtained by applying motion blur kernel on the latent image (according to the depth map of Fig. 3 (a)) in Fig. 5 (e). From the resultant LF image (shown in Fig. 5 (c)), we estimate the depth and solve for the layer support, latent image and motion blur kernel. While the ground truth supports is shown in Fig. 5 (a), the corresponding estimated support is shown in Fig. 5 (b). Despite regions in the image with significantly less texture, we see that the estimated support matches the true support. The recovered latent image and motion blur kernel are shown in Fig. 5 (f). It is to be noted that the restored image is quite close to the true image and there are no artifacts at the depth discontinuities thanks to our layered model.

In all our experiments, the same regularization parameters were used: $\lambda_f = 10^{-5}, \lambda_\Omega = 5 \times 10^{-4}$, and $\lambda_M = 10^{-3}$. For evaluation, we use the Peak Signal-to-Noise ratio (PSNR) metric. Between the blur kernel and the sharp image there is a translational ambiguity. Hence, for each image we take the maximum PSNR among all the possible shifts between the estimated and ground truth image. In Table 2 we show the mean and standard deviation PSNR values.

We perform real experiments using the Lytro Illum camera. We imaged a 3D scene with objects placed at different distances from the camera ranging from 50 cm to 100 cm. We placed the camera on a support to restrict its motion to in-plane translations. Due to the high dimensionality, we extract specific regions from the full light field image and perform reconstruction on these regions separately. Each region contains a pair of objects at different depths. The camera settings are summarized in Table 3. We extract, rectify and normalize the Lytro LF images by using the Light Field Toolbox V0.4 software.[1] Through our alternate minimization scheme, we solve for the support, sharp texture and motion blur. In contrast to the scenario of conventional camera images, our estimate of layer support improves as the iterations progress. For one of the examples, we show the evolution of support in Fig. 7. In Fig. 6 from left to right we show: input LF image region, reconstructed depth map (Lytro), reconstructed depth map (ours), final estimated supports, reconstructed blurred image from Lytro (it does not perform motion deblurring), reconstructed image from Lytro of the same static scene without motion blur, and reconstructed sharp image (composite) with estimated motion blur at the first layer (insert at the bottom-right). We only show the estimated motion blur on the first layer as the other layers are just scaled (down) versions of that blur. Notice how the proposed scheme can effectively remove motion blur from the LF images by comparing them with the images generated by the Lytro software of the same scene without motion blur.

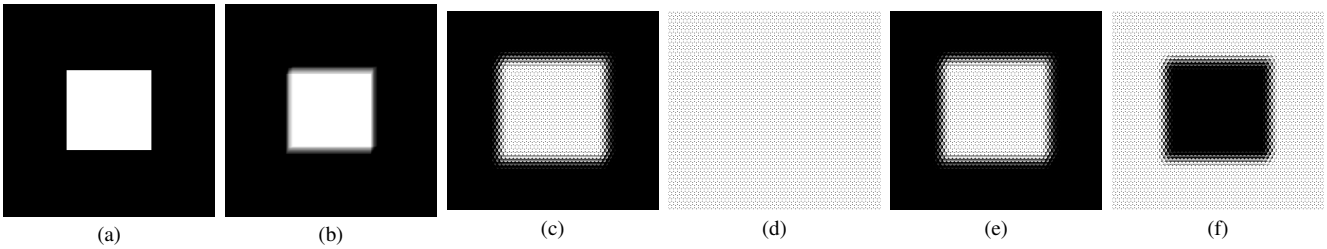
To demonstrate the consistency in our estimates, we simulate different sub-aperture views from the texture and layer support. We generated the left view shown in Fig. 8 (a) by applying a shift on each layer of the texture and its support. For a particular depth layer, the shift remains the same for the texture as well as for the support, and it changes as the depth changes. Similarly, we generated the right view in Fig. 8 (b). In both these images, we observe the effect of occlusion/disocclusion without any artifacts at the depth boundaries indicating that our estimates are accurate.

We also tested our algorithm on a scene with three depth layers as shown in the last row of Fig. 6. Although we see that the estimated texture is sharper than the Lytro rendering of the blurred LF image, when compared to the rendering of the sharp scene, the result shows artifacts. We believe that this is due to the increased complexity of the model and the need for higher depth estimation accuracy.

6. Conclusions

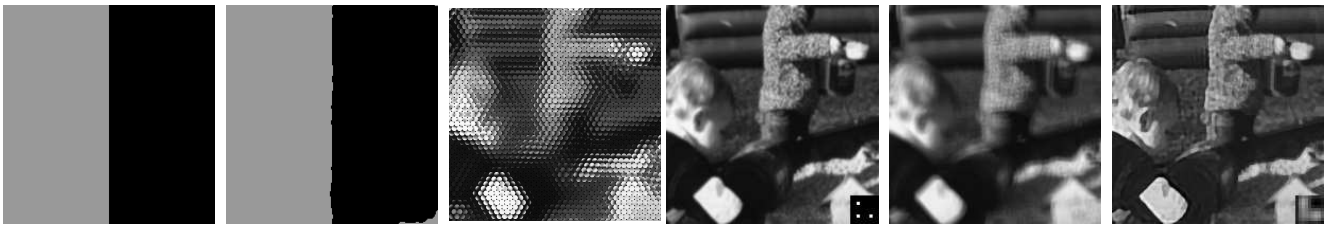
We introduced the novel problem of restoring a blurry light field image. We consider depth variations and model partial transparencies at occlusions. Through an energy minimization framework, we estimated the depth map as a set of discrete layers, sharp scene textures, and the mo-

[1] http://www.mathworks.com/matlabcentral/fileexchange/49683-light-field-toolbox-v0-4



(a) (b) (c) (d) (e) (f)

Figure 4. Components of the imaging model for the scene in Fig. 3 (d) : (a) Ω_1, (b) Ω_1^b, (c) α_1^b , (d) α_2^b, (e) β_1^b, (f) β_2^b.



(a) (b) (c) (d) (e) (f)

Figure 5. Two layer hexagonal scenario: (a) and (b) ground truth and recovered first layer supports; (c) simulated motion blurred LF image; (d) true image and blur kernel (e) blurry texture; f) recoverd texture and kernel.

Table 2. In this table we show the average (μ) and the standard deviation (σ) of the PSNR metrics for 9 synthetically generated motion blurred light fields.

	Rectangular	**Hexagonal**
μ	29.1229	24.2006
σ	1.4832	3.0105

Table 3. Summary of the Lytro Illum settings.

vertical rows	70
horizontal microlenses	62
pixels per microlens	16×16
vertical spacing between even rows	28 pixels
main lens focal length (F)	0.0095
pixel size	1.4 μm
main lens F-number	2.049
microlens spacing	20 μm
main lens to microlens array distance	9.8 mm
microlens array to sensor distance	47.8 μm
microlens focal length	48.0 μm
shutter	1/2 s
ISO	80
EV	+0.7

tion blur kernels by enforcing suitable priors. In contrast, for conventional images, estimation of layer support is not feasible as seen in our simulation. The proposed method is able to adapt to scaling of motion blur and return artifact-free boundaries at depth discontinuities. Our bilayer image formation model can be generalized to multiple depth layers. Since the LF image generation is parallelizable, an efficient implementation of our algorithm can be achieved by using GPUs. Further extensions of our work include handling camera rotations and dynamic scenes.

Acknowledgements

This work has been supported by the Swiss National Science Foundation (Project No. 153324).

References

[1] E. H. Adelson and J. Y. A. Wang. Single lens stereo with a plenoptic camera. *TPAMI*, 14:99–106, 1992. 2

[2] T. Bishop and P. Favaro. The light field camera: extended depth of field, aliasing and superresolution. *TPAMI*, 34(5):972–986, 2012. 2, 3, 4

[3] Y. Bok, H.-G. Jeon, and I. S. Kweon. Geometric calibration of micro-lens-based light-field cameras using line features. In *ECCV*, 2014. 2

[4] C. A. Bouman, I. Pollak, and P. J. Wolfe, editors. *Superresolution with the focused plenoptic camera*, volume 7873. SPIE, 2011. 2

[5] M. Broxton, L. Grosenick, S. Yang, N. Cohen, A. Andalman, K. Deisseroth, and M. Levoy. Wave optics theory and 3-d deconvolution for the light field microscope. *Opt. Express*, 21:25418–25439, 2013. 2, 3

[6] T. Chan and C.-K. Wong. Total variation blind deconvolution. *IEEE Transactions on Image Processing*, 7(3):370–375, 1998. 2

[7] D. Cho, M. Lee, S. Kim, and Y.-W. Tai. Modeling the calibration pipeline of the lytro camera for high quality light-field image reconstruction. In *ICCV*, 2013. 2

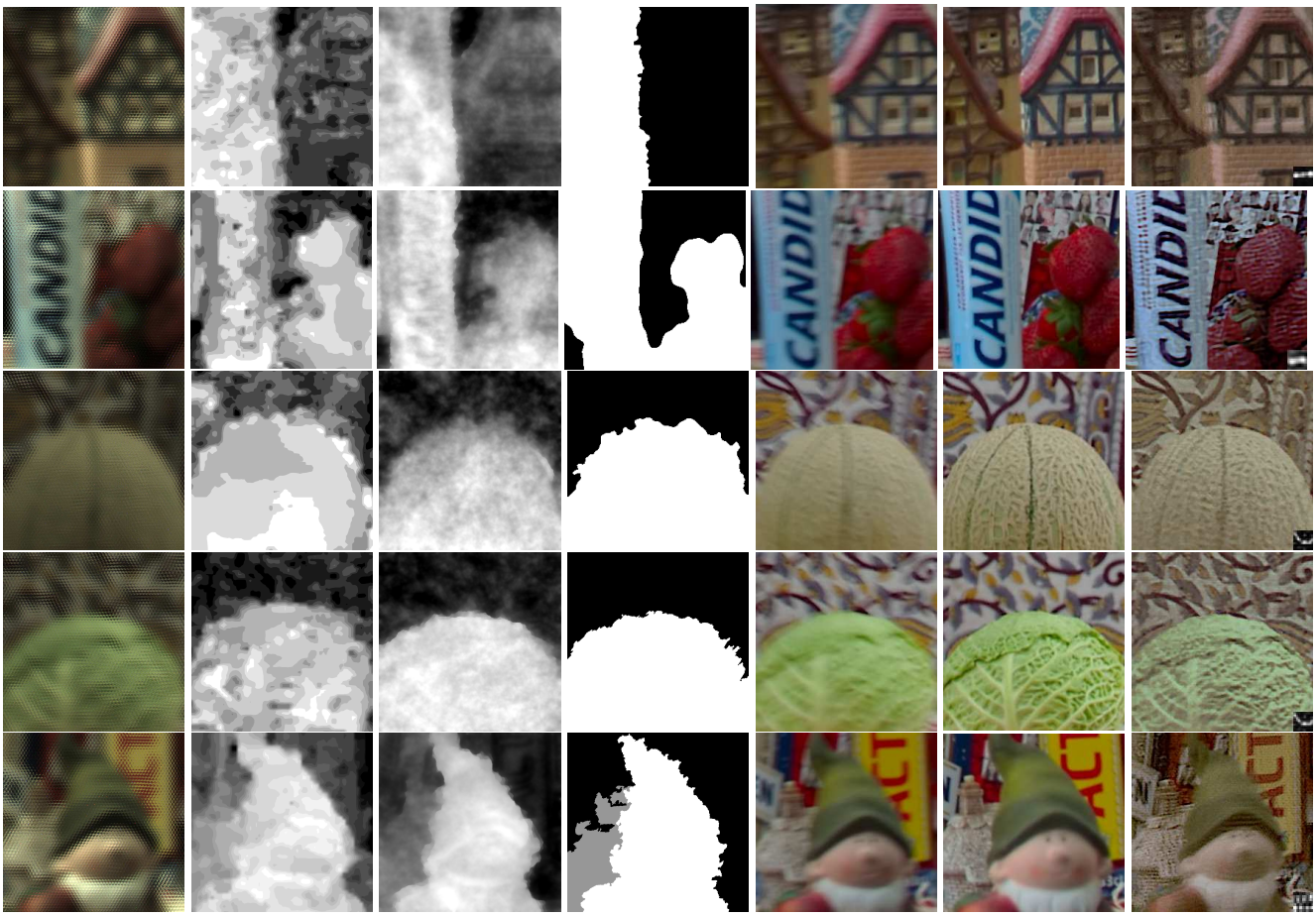


Figure 6. Experiments on real images (the first, second and fifth rows show results from the same scene and the third and fourth rows show results from another scene 2): First column: light field images (cropped region), second and third columns are depth maps from Lytro and our depth estimation. Fourth column: final supports. Fifth and sixth columns are blurred and no-blur texture generated from LYTRO software. The seventh column shows the estimated sharp image (merged with the estimated supports) with the estimated motion blur as an insert at the bottom-right corner.

Figure 7. Evolution of layer support of the scene shown in second row of in Fig. 6

[8] D. Cho, M. Lee, S. Kim, and Y.-W. Tai. Consistent matting for light field images. In *ECCV*, 2014. 3

[9] S. Cho and S. Lee. Fast motion deblurring. *ACM Trans. Graph.*, 28(5):1–8, 2009. 1, 2

[10] D. G. Dansereau, O. Pizarro, and S. B. Williams. Decoding, calibration and rectification for lenselet-based plenoptic cameras. In *CVPR*, 2013. 2

[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, 2006. 1, 2

[12] A. Gupta, N. Joshi, L. Zitnick, M. Cohen, and B. Curless. Single image deblurring using motion density functions. In *ECCV*, 2010. 2

[13] S. W. Hasinoff and K. N. Kutulakos. A layer-based restoration framework for variable-aperture photography. In *ICCV*, pages 1–8, 2007. 3

[14] S. Heber and T. Pock. Scene flow estimation from light fields via the preconditioned primal-dual algorithm. volume 8753 of *LNCS*, pages 3–14. 2014. 3

[15] S. Heber, R. Ranftl, and T. Pock. Variational shape from light field. In *EMMCVPR*, pages 66–79. 2013. 2

(a) (b)

Figure 8. View synthesis: (a) Synthesized left view. (b) Synthesized right view.

[16] M. Hirsch, C. J. Schuler, S. Harmeling, and B. Scholkopf. Fast removal of non-uniform camera shake. In *ICCV*, 2011. 2

[17] Z. Hu, L. Xu, and M.-H. Yang. Joint depth estimation and camera shake removal from single blurry image. In *CVPR*, 2014. 2

[18] H. Ji and K. Wang. A two-stage approach to blind spatially-varying motion deblurring. In *CVPR*, 2012. 2

[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. 2

[20] A. Levin, Y. Weiss, F. Durand, and W. Freeman. Efficient marginal likelihood optimization in blind deconvolution. In *CVPR*, pages 2657–2664, 2011. 1, 2

[21] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *CVPR*, 2009. 2, 5

[22] C.-K. Liang and R. Ramamoorthi. A light transport framework for lenslet light field cameras. *ACM Trans. Graph.*, 34(2):16:1–16:19, Mar. 2015. 3

[23] A. Lumsdaine and T. Georgiev. Full resolution lightfield rendering. Technical report, Adobe Systems, 2008. 2

[24] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan. Light field photography with a handheld plenoptic camera. *CSTR*, 2(11), 2005. 2

[25] C. Paramanand and A. N. Rajagopalan. Non-uniform motion deblurring for bilayer scenes. In *CVPR*, 2013. 2

[26] D. Perrone and P. Favaro. Total variation blind deconvolution: The devil is in the details. In *CVPR*, 2014. 2

[27] N. Sabater, M. Seifi, V. Drazic, G. Sandri, and P. Perez. Accurate disparity estimation for plenoptic images. In *ECCVW*, 2014. 2

[28] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. *ACM Transactions on Graphics*, 27(3). 1, 2

[29] M. Sorel and J. Flusser. Space-variant restoration of images degraded by camera motion blur. *IEEE Trans. Img. Proc.*, 17(2):105–116, 2008. 2, 4

[30] L. Sun, S. Cho, J. Wang, and J. Hays. Edge-based blur kernel estimation using patch priors. In *ICCP*, 2013. 2

[31] Y. Tai, P. Tan, and M. S. Brown. Richardson-lucy deblurring for scenes under projective motion path. *TPAMI*, 33(8):1603–1618, 2011. 2

[32] M. Tao, S. Hadap, J. Malik, and R. Ramamoorthi. Depth from combining defocus and correspondence using light-field cameras. In *ICCV*, 2013. 2

[33] S. Wanner and B. Goldluecke. Variational light field analysis for disparity estimation and super-resolution. *TPAMI*, 36(3):606–619, 2014. 3

[34] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. In *CVPR*, 2010. 2

[35] D. Wipf and H. Zhang. Revisiting bayesian blind deconvolution. *J. Mach. Learn. Res.*, 15(1):3595–3634, Jan. 2014. 2

[36] L. Xu and J. Jia. Two-phase kernel estimation for robust motion deblurring. In *ECCV*, 2010. 1, 2

[37] L. Xu and J. Jia. Depth-aware motion deblurring. In *ICCP*, pages 1–8, April 2012. 2

[38] Y.-L. You and M. Kaveh. Anisotropic blind image restoration. In *ICIP*, pages 461–464 vol.2, 1996. 2