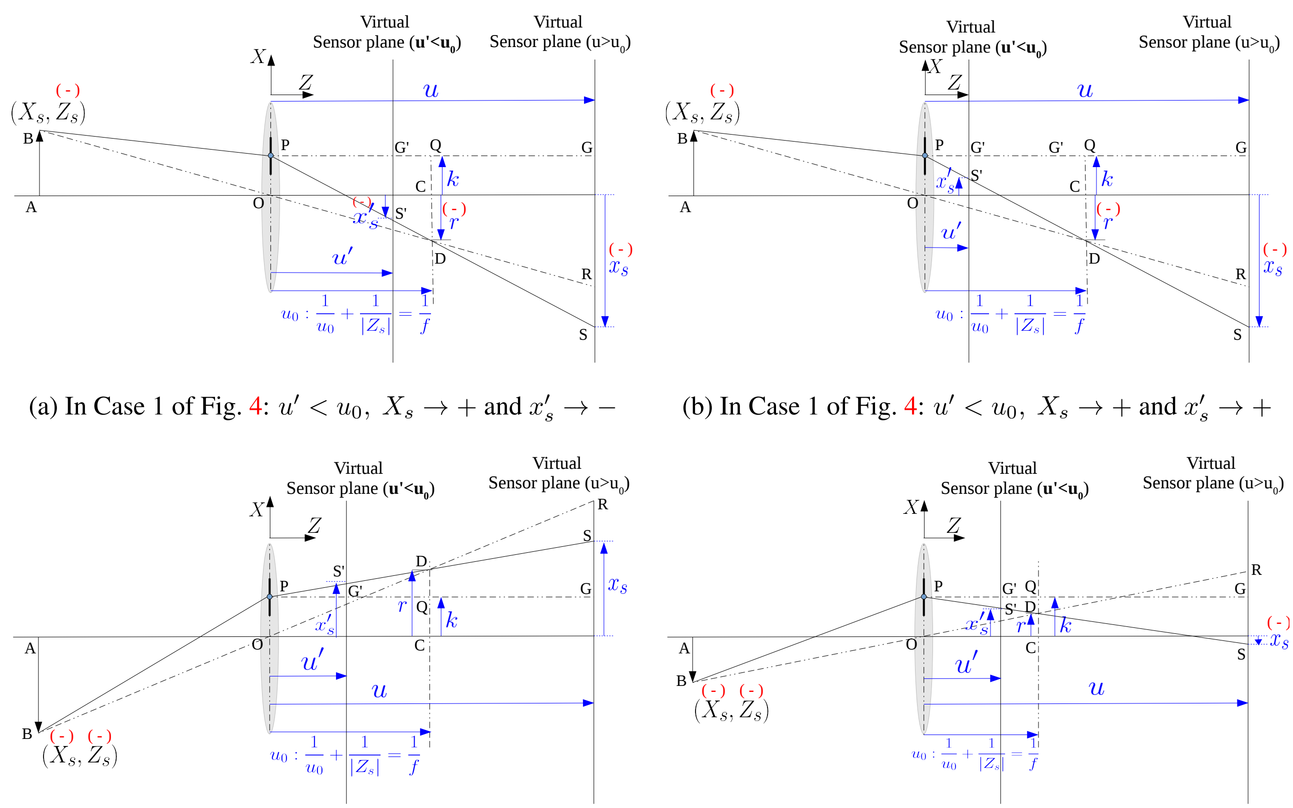
Divide and Conquer for Full-Resolution Light Field Deblurring (Supplementary Material)

M. R. Mahesh Mohan, A. N. Rajagopalan
Indian Institute of Technology Madras
{ee14d023,raju}@ee.iitm.ac.in

We begin by proving that the world-to-sensor mapping derived in Sec. 3.1, and thus our LF-MDF model, holds good in general. This is followed by a discussion on the choice of our deconvolution method in EFF (Sec. S2), and analysis of various aspects of our LF-BMD (Sec. S3). We then provide our implementation details in Sec. S4, and additional evaluations in Sec. S5. Note that the sections, equations, and figures in the supplementary are numbered with a prefix ‘S’.



(a) In Case 1 of Fig. 4: $u' < u_0,\ X_s \to +$ and $x'_s \to -$

(b) In Case 1 of Fig. 4: $u' < u_0,\ X_s \to +$ and $x'_s \to +$

(c) In Case 2 of Fig. 4: $u' < u_0,\ X_s \to -$ and $x'_s \to +$

(d) In Case 3 of Fig. 4: $u' < u_0,\ X_s \to -$ and $x'_s \to +$

Figure S1. Different cases of world-to-sensor mapping for $u \leq u_0$ for a subaperture positioned at positive X axis. Note that these cases are superimposed on the $u > u_0$ cases shown in Fig. 4. (A symbol $'$ is added to those variables representing $u \leq u_0$.)

(a) Input (b) [9] (2X bicubic interpolated) (c) Direct approach (Gaussian prior)

(d) [5] (Hyper-Laplacian prior) (e) [6] (0.8 norm on gradient) (f) RL deconvolution [8]

Figure S2. Qualitative evaluation of different LF-EFF deconvolutions using a full-resolution LF. (a) Input, (b) LF-BMD result of [9] for reference (2X bicubic-interpolated). (c) Direct approach using Gaussian prior, (d) Fast MAP estimation with hyper-Laplacian prior using lookup table [5], (e) MAP estimation with heavy-tailed prior ($\alpha = 0.8$) [6], and (f) Richarson Lucy deconvolution [8]. Note the ringing artifacts in c in the saturated regions (*e.g.*, in lights and door exit). RL deconvolution in f produces the best result with negligible artifacts.

Deconvolution method	Direct (Gaussian)	[5] (Fast hyper-Laplacian)	[6] (0.8 norm on gradients)	[8] RL deconv.
Time/SA image (**Full-res. LF**)	1.1 second (closed-form)	6.2 seconds (lookup table)	55 seconds (50 iters.)	80 seconds (50 iters.)

Table S1. Time per subaperture (SA) image for different LF-EFF deconvolution methods for full-resolution LFs.

S1. World-to-Sensor Mapping for the case $\mathbf{u} \leq \mathbf{u_0}$

We showed in Sec. 3.1 that the three fundamental equations (Eqs. (7)-(9)) that we employed to derive world-to-sensor mapping for a subaperture (and further for the MDF formulation) hold good for different cases of $u > u_0$ (*i.e.*, sensor plane in front of the focal-plane of the lens). Here we show that those relations are also valid for $u \leq u_0$. In Fig. S1, we depict various cases of $u \leq u_0$, superimposed on the $u > u_0$ cases in Fig. 4. For distinguishing both cases, we have used a symbol '$'$' to indicate variables of $u \leq u_0$ case (*e.g.*, u' indicates u). Eqs. (7) and (8) can be verified using the lens equation and similarity of triangles ΔABO and ΔODC, respectively (as in Sec. 3.1). Similarity of $\Delta PG'S'$ and ΔPQD gives

$$\frac{k-r}{k-x_s'} = \frac{u_0}{u'} \implies x_s' = r \cdot \frac{u'}{u_0} - k \cdot \left(\frac{u'}{u_0} - 1\right), \tag{S1}$$

which is same as Eq. (9). This shows that Eqs. (7)-(9) hold true *in general* for a subaperture positioned at positive X axis as well; *i.e.*, valid irrespective of the scene-point location and the sensor-plane placement ($u > u_0$ or $u \leq u_0$). Due to symmetry about the optical axis of ray diagrams, these relations are *equally valid for subapertures positioned at negative X axis*.

S2. Choice of LF-Deconvolution

In this section, we discuss our choice of deconvolution method employed to perform LF-EFF patch-wise deblurring in Eq. (20). A nonblind LF-EFF deconvolution problem, *i.e.* estimation of a clean image patch given the blur kernel and blurred

(a) Input | (b) Ours (Prior 0.005) $\approx$ 17X gain with CPU | (c) Ours (Prior 0.009) $\approx$ 17X gain with CPU | (d) Ours (Prior 0.05) $\approx$ 17X gain with CPU | (e) Srinivasan *et al.* [9] **GPU-based**

Figure S3. Effect of prior in our LF-BMD (using dataset of [9]). (a) Input, (b) Ours with default smoothness regularization (SR) 0.005, (c) Ours with SR 0.009, (d) Ours with SR 0.05, and (e) State-of-the-art [9] for reference. In e, notice the ringing artifacts in the upper leaves and the suppressed veins of lower leaf (shown boxed). Our result with 0.05 prior is comparable to that of [9], yet with negligible ringing artifacts. Moreover, ours is CPU-based and yet achieves a speed-up of atleast an order ($\approx$ 17X) as compared to [9] which is GPU-based.

image patch, possesses multiple solutions due to zero crossings of filter response, saturation or noise effects, *etc*. Maximum a posteriori (MAP) estimation which imposes prior(s) on clean image patch is typically employed to obtain a single solution from the multiple solution space. A MAP estimation for nonblind deconvolution is given as

$$\hat{I} = \text{deconv}(h, B) = \min_{I} \|HI - B\|_2^2 + \|\nabla I\|_\alpha \tag{S2}$$

where H captures the blur-kernel information, ∇ is the gradient operator, and B and I are blurred and latent image patches, respectively. We considered four different deconvolution approaches: (a) A direct approach which considers Gaussian prior ($\alpha = 2$) and thus has a closed form solution, (b) A fast deconvolution using hyper-Laplacian prior ($0.5 \leq \alpha \leq 0.8$) which is solved using a lookup table [5], (c) A heavy-tailed prior ($\alpha = 0.8$) which is solved using iterative reweighted least squares process [6], and (d) RL deconvolution with smoothness prior which is solved using iterative process [8]. Figure S2 provides a representative example of LF deblurring quality (using Fig. 8) with different approaches, and Table. S1 gives the average time per subaperture image; it is evident that there exists a trade-off between visual quality and computational speed. In terms of visual quality, we empirically found out that RL [8] is the best, and the direct method comes second but with ringing artifacts (*e.g*., see Fig. S2(c)). In terms of computational time, the direct method is the most efficient, whereas RL (due to its iterative approach) is less efficient. We have selected RL method due to its superior deblurring quality. However, direct deblurring can be selected for computational efficiency, provided one can tolerate minor ringing artifacts.

S3. Analysis

We showed that our method produces comparable results with respect to the state-of-the-art (GPU-based) [9], yet with significant computational gain even on CPU. Moreover, our method deblurs full-resolution LFs, unlike [9] which can process *only* downsampled LFs. In this section, we consider the effect of noise in our LF-BMD and propound ways to suppress it, and analyse the effect of adding more subaperture images (SAIs) to estimate the MDF (instead of one SAI that we followed).

Noise in LF-BMD: LF images captured in low-light scenarios possess higher level of shot noise as compared to that of an analogous CC-camera (due to segregation of photons for angular resolution) [12]. As deblurring can be interpreted as enhancing the high-frequency content of the scene, LF-BMD also enhances the high-frequency noise (if present). As discussed in Sec. 4.1, we consider the center subaperture image to estimate the common LF-MDF using [11]. State-of-the-art CC-BMDs frame the objective function in image's gradient space so as to reduce the ill-conditionness [3, 11]. Unlike the gradient of scene features which form contiguous segments, the gradients of shot noise form isolated spikes. Harnessing this, we remove the less-contiguous segments from image-gradient to form the objective function, which reduces the ill-effects of noise in MDF-estimation. For nonblind deblurring (Sec. 4.2), we use the estimated MDF to obtain patch-wise kernels for individual subaperture images (Eq. (19)), and perform deconvolution using [8]. In case of noisy images, we use a higher smoothness prior (regularization of 0.05) for deconvolution to reduce the noise-effect in deblurred images. Our default regularization value is 0.005. To show how noise can be handled as well as give comparison with [9] on Fig. 7 (which uses our default setting), Fig. S3 provides the effect of varying regularization that clearly shows suppression of noise as the prior increases.

More SAIs to find MDF: Incorporating more SAIs does *not* produce any significant improvement in MDF, while *accentuating* the computational cost. MDF is estimated as $\hat{\omega_\lambda} = \min_{\omega_\lambda} \|H_{I_{k_{xy}}} \omega_\lambda - B_{k_{xy}}\|_2 + \|\omega_\lambda\|_0$. For a maximum 30 pixel blur, 3D rotation space binned by 1 pixel is $\mathbf{29}^3$. Considering a single SAI ($< 1\%$ data), the number of equations (or the number of SAI pixels) will be $\mathbf{10}$X as that of the number of unknowns, which is already an *overdetermined system (ODS)* and sufficient for MDF estimation [16,23]. Incorporating n more SAIs scales the number of equations by order of n (*but the effect of* more *ODS* $\approx$ *ODS*), while incurring additional cost for creating *individual* H_is and *handling* large matrix (n H_is stacked).

S4. Implementation Details

System Specifications: We used a PC with an Intel Xeon processor and a 16 GB RAM for all CPU-based experiments, and implemented our algorithm in `MATLAB`. The repeatedly used EFF routine is implemented in `C` for computational efficiency. We perform nonblind deblurring of eight subaperture images in parallel. For executing the code of [9], we used a GPU-server and employed a Pascal Titan X GPU. Running time reported in Table. 1 is obtained using these specifications. The camera we used for obtaining full-resolution light field examples is `LYTRO ILLUM 40 Megaray`.
Parameters: We employed Lytro Desktop App to download LF raw images and [1] to decode raw images into LF Matlab file. The camera parameters focal length f and lens-sensor separation u are obtained from Lytro metadata. As Lytro camera has constant aperture setting as $f/2$, we periodically sampled 197 subapertures in a circular disk of the aperture dimension to obtain k_x and k_y. We used camera metadata and a modified source code of [10] to produce discrete depth with respect to the center subaperture image in individual patches (as discussed in Sec. 4.2).

The sensor coordinate $\mathbf{x}$ corresponding to a scene point varies with subaperture k_{xy} due to parallax and lens effect (*e.g.*, in Fig. S1, for the case of $u > u_0$ the depth Z_s of a scene point maps to sensor coordinate at R through the centre pinhole, whereas shifted by RS through the shifted pinhole). As the depth estimate Z obtained using [10] is with respect to the center subaperture image, it is required to map this to other non-centered subaperture images for retaining one-to-one correspondence between x and Z (in Eq. 14). This we accomplished by warping the estimated depth (with coordinate x) to subaperture k_{xy} (with coordinate x') as $x' = x - \delta x_{k_{x,y}}$, where $\delta x_{k_{x,y}}$ is derived using similarity of ΔDOP and ΔDRS in Fig. 4 or S1:

$$\delta x_{k_{x,y}} = k \cdot \frac{u - u_0}{u_0}. \tag{S3}$$

where u_0 is a function of Z. This relation even holds true for the case of $u < u_0$ (which is verifiable using Fig. S1).
Development: Our algorithm comprises of two steps: blind deblurring of center subaperture image to estimate the common MDF and project the estimated MDF to other subaperture images to perform nonblind deblurring (in parallel) employing EFF. For the first step, as the MDF-based source code of the best CC-BMD [7] is not available and [13] provides only an executable code, we used a modified code of [11] to incorporate LF parameters. For the scale-space based alternative minimization for MDF and latent image, we used 5 scales with 6 iterations each. For all experiments, we used MDF regularization as 0.01 and total variation regularization as 0.005. For the second step, we implemented a C-based EFF code to obtain kernels corresponding to the patch centers using Eq. (19), and employ RL as the deconvolution method in Eq. (20). A pseudo-code is provided in algorithm 1.

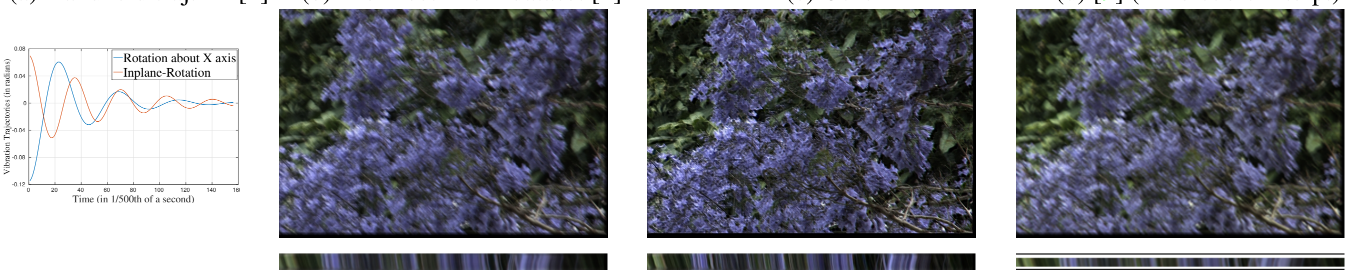
Algorithm 1 Light field blind motion deblurring

Require: Decoded motion blurred LF file (LF)	(using [1])
Estimate patch-wise depth using [10]	(following Sec. 5)
$centerSAI \leftarrow LF(0,0)$	
Estimate MDF using the $centerSAI$	(employing [11])
for all SAIs (in parallel) **do**	
Project blur in SAI patches using the estimated MDF	(using Eq. (19))
Patch-wise deconvolution using the projected blur	(using Eq. (20))
Merge individual patches using windowing operation	(Sec. 4.2)
end for	

S5. Additional Evaluations

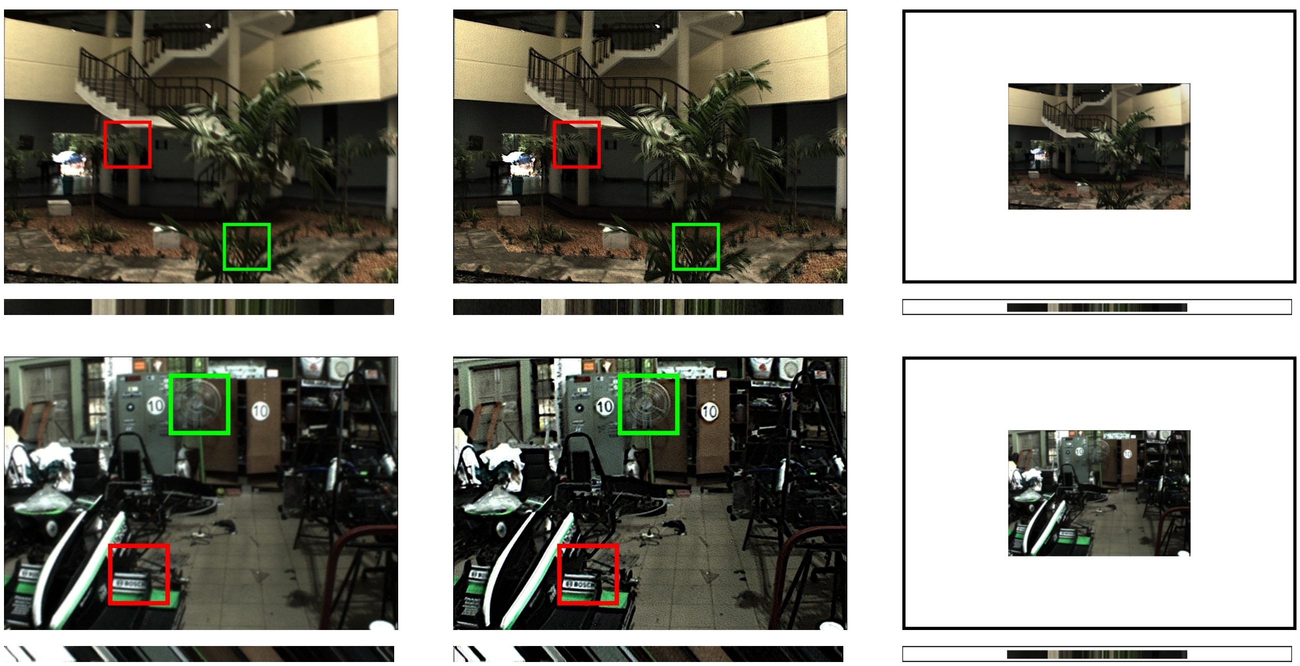
We provide additional comparisons against the state-of-the-art [9] in Figs. S4-S5. Fig. S4-top provides synthetic experiment results on datatset [1] using real handheld trajectory from [4]. Fig. S4(c)-top shows the result of [9] which is bicubic interpolated to match the full-resolution LF. It is evident from the figure that the interpolation of low-resolution deblurred

(a) Handheld traj. #1 [4] (b) 'Lorikeet' from dataset [1] (c) Ours (d) [9] (2X bicubic interp.)



(e) Vibration traj. [2] (f) 'Jarkanda' from dataset [1] (g) Ours (h) [9] (2X bicubic interp.)

Figure S4. Synthetic experiments in dataset [1] using real handheld [4] and vibration [2] trajectories. (a,e) Trajectories, (b,f) Inputs, (c,g) Ours, and (d,h) Bicubic interpolated result of [9]. Top-row gives a case of handheld trajectory. In d, note that the low-resolution result of [9] after interpolation fails to recover intricate details (*e.g.*, feathers in lorikeet's face). Bottom-row gives a case of irregular motion. Deblurring performance of [9] in h is quite low, possibly due to the inability of its parametric motion model in capturing vibratory motion.



(a) Input (c) Ours (e) Srinivasan *et al*. [9]

Figure S5. Additional real experiments. (a) Input, (b) Ours, and (c) State-of-the-art [9]. Top row gives a wide-angle scenario (of Fig. 8). Bottom row shows an image of garage. Note that the state-of-the-art [9] *cannot* process a full-resolution LF due to computational considerations, whereas ours performs full-resolution LF-BMD. Also, our method is CPU-based, unlike [9] which requires GPU.

image fails to recover intricate details (*e.g.*, the feathers in lorikeet's face), which further underscores the importance of performing LF deblurring at its full-resolution. Fig. S4-bottom shows a case of irregular motion using vibration trajectory from

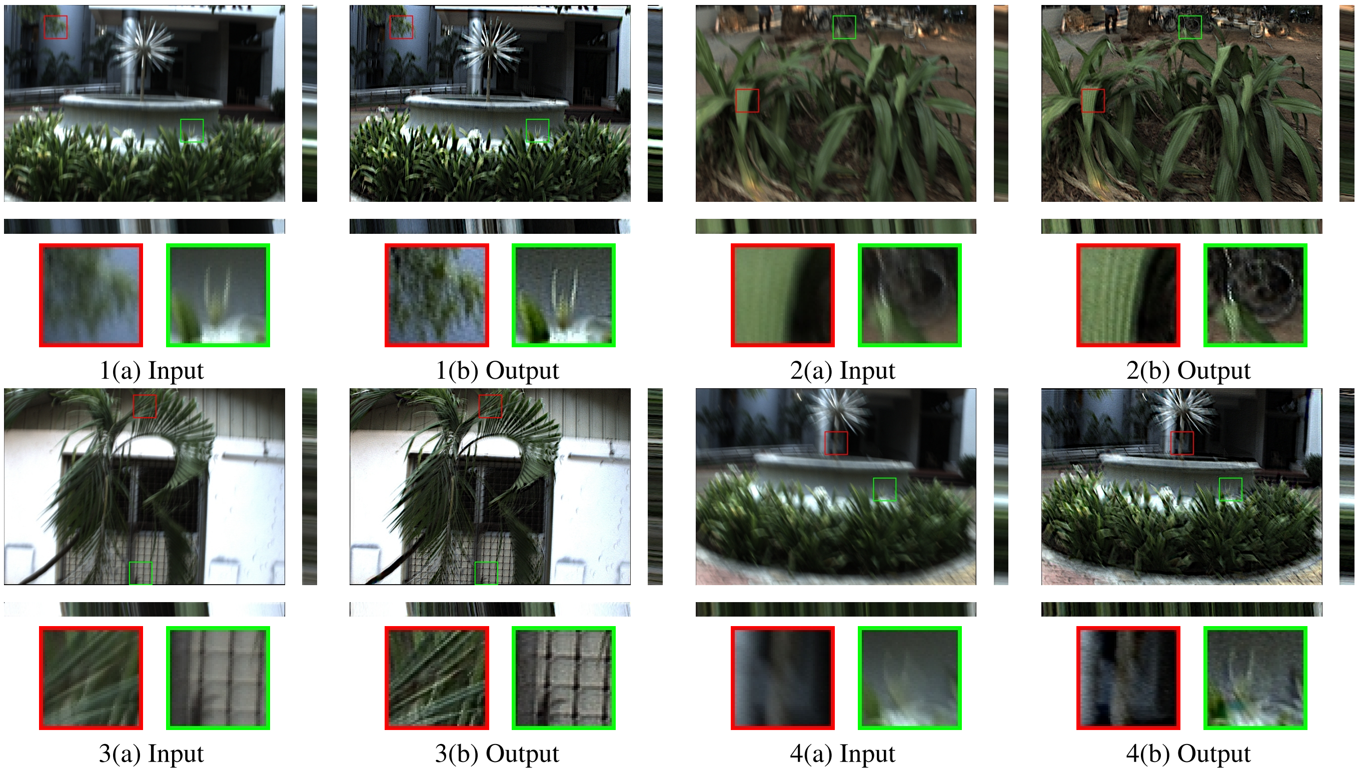


Figure S6. Four additional examples on full-resolution LFs ($\{433, 625, 15, 15\}$) captured using `Lytro Illum`. The first three examples (1-3) depict normal hand-shake blurs, whereas the fourth example gives an example of heavy motion blur. Notice the consistent EPIs in all examples. Also, patches are shown to highlight the deblurring performance.

[2]. Fig. S4(c)-bottom shows the result of [9], where the deblurring performance is inferior (possibly due to the inefficacy of its parametric motion in capturing irregular trajectory). Fig. S5 shows two additional evaluations on real full-resolution LF examples (top-row shows an outdoor wide-angle LF-image and bottom-row shows an indoor narrow-angle LF-image); and Fig. S6 shows four more additional examples, which yet again proves the effectiveness of our proposed method.

References

[1] D. G. Dansereau, O. Pizarro, and S. B. Williams. Decoding, calibration and rectification for lenselet-based plenoptic cameras. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1027–1034, 2013. 4, 5

[2] M. R. Hatch. *Vibration simulation using MATLAB and ANSYS*. CRC Press, 2000. 5, 6

[3] M. Hirsch, S. Sra, B. Schölkopf, and S. Harmeling. Efficient filter flow for space-variant multiframe blind deconvolution. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 607–614. IEEE, 2010. 3

[4] R. Köhler, M. Hirsch, B. Mohler, B. Schölkopf, and S. Harmeling. Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database. In *European Conference on Computer Vision (ECCV)*, pages 27–40. Springer, 2012. 4, 5

[5] D. Krishnan and R. Fergus. Fast image deconvolution using hyper-laplacian priors. In *Advances in Neural Information Processing Systems*, pages 1033–1041, 2009. 2, 3

[6] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM transactions on graphics (TOG)*, 26(3):70, 2007. 2, 3

[7] J. Pan, D. Sun, H. Pfister, and M.-H. Yang. Blind image deblurring using dark channel prior. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1628–1636. IEEE, 2016. 4

[8] W. H. Richardson. Bayesian-based iterative method of image restoration. *JOSA*, 62(1):55–59, 1972. 2, 3

[9] P. P. Srinivasan, R. Ng, and R. Ramamoorthi. Light field blind motion deblurring. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2, 3, 4, 5, 6

[10] M. W. Tao, S. Hadap, J. Malik, and R. Ramamoorthi. Depth from combining defocus and correspondence using light-field cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 673–680, 2013. 4

[11] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. *International journal of computer vision (IJCV)*, 98(2):168–186, 2012. 3, 4

[12] G. Wu, B. Masia, A. Jarabo, Y. Zhang, L. Wang, Q. Dai, T. Chai, and Y. Liu. Light field image processing: An overview. *IEEE Journal of Selected Topics in Signal Processing*, 2017. 3

[13] L. Xu, S. Zheng, and J. Jia. Unnatural l0 sparse representation for natural image deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1107–1114. IEEE, 2013. 4