Depth-Aware Motion Deblurring*

Li Xu   Jiaya Jia

Department of Computer Science and Engineering

The Chinese University of Hong Kong

{xuli, leojia}@cse.cuhk.edu.hk

Abstract

Motion deblurring from images that are captured in a scene with depth variation needs to estimate spatially-varying point spread functions (PSFs). We tackle this problem with a stereopsis configuration, using depth information to help blur removal. We observe that the simple scheme to partition the blurred images into regions and estimate their PSFs respectively may make small-size regions lack necessary structural information to guide PSF estimation and accordingly propose region trees to hierarchically estimate them. Erroneous PSFs are rejected with a novel PSF selection scheme, based on the shock filtering invariance of natural images. Our framework also applies to general single-image spatially-varying deblurring.

1. Introduction

Image motion blur removal [4, 28, 22, 3, 27, 13, 16] aims at recovering structure information from motion blurred images. Most prior methods do not consider ubiquitous depth variation. When objects are not far from the camera, spatially-varying point-spread-functions (PSFs) arise along with abrupt change along the boundaries, making a near point be blurred more than a distant one. Existing non-uniform deblurring methods cannot be directly applied in this case because depth discontinuity commonly yields *sharp* PSF variation.

The existence of depth variation complicates deblurring a lot. Even with the depth information, the possible *small sizes* of depth layers can be a critical problem, due to the lack of necessary structural information to constrain corresponding optimization. So correctly removing blur in this case needs to address the robustness issue in PSF computation. It obviously differs from other uniform and non-uniform blur removal problems, where a whole image or sufficiently large regions are available.
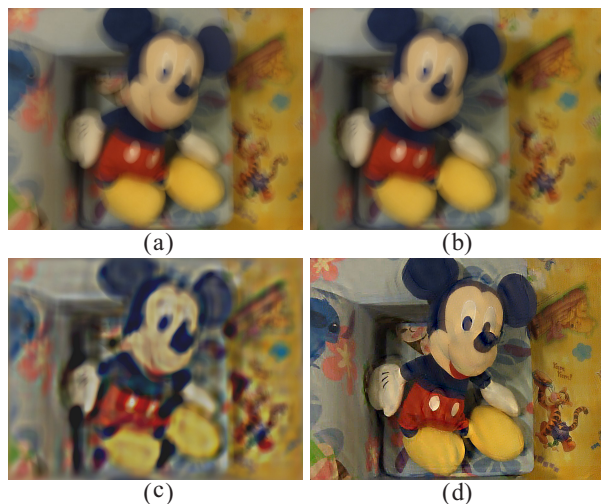
Figure 1. Stereo motion deblurring. (a-b) Captured images. (c) Image restored by the method of [4]. (d) Our result.

In this paper, we propose a hierarchical estimation framework, based on *region trees* to progressively generate credible blur kernel estimates. To detect and ameliorate problematic PSF estimates, we contribute a novel *PSF selection* scheme to find the most suitable kernel from a few candidates utilizing an important *shock filtering invariance* property. This scheme is very effective to refine complex PSFs even in small regions.

To know depth for PSF estimation, we employ a stereopsis configuration, i.e., with two captured images, to provide necessary constraints for depth map inference. The hardware to take stereo images is simply an off-the-shelf stereo camera [5], in which two independent and calibrated lens and CCDs are assembled. We allow complex depth and assume primarily in-plane translational camera motion. Our method also tolerates small rotational camera shake since robust PSF estimation is performed respectively for regions. A pair of stereo images is shown in Fig. 1(a)-(b), captured by the hand-held stereo camera [5]. The results of a prior motion deblurring method [4] and of ours are respectively shown in (c) and (d).

In addition, our new region-tree-based PSF estimation scheme can deal with traditional single-image non-uniform deblurring without depth variation consideration. In this case, the same PSF estimation procedure can be applied to small-size rectangular regions after regularly partitioning the input image. This approach is thus very general, robust against uncontrolled camera motion. Several challenging examples are shown in this paper and in the project website[1].

2. Related Work

In single image deblurring, Fergus *et al.* [4] proposed a variational Bayesian approach to estimate the blur kernel by maximizing the marginal probability. Efficient marginal likelihood approximation was introduced in [16]. Detailed analysis of the MAP problem in motion deblurring was provided in [15]. Several methods [22, 10, 3, 27] followed the line of altering the traditional MAP framework to estimate the blur kernel and latent image iteratively. They do not favor the trivial delta kernel solution by either explicitly re-weighting strong edges [22], predicting sharp edges using filters [10, 3, 27], or directly imposing normalized sparsity measures on image gradients [13]. These methods do not handle spatially-varying PSF estimation.

Multi-image methods were proposed. Ben-Ezra and Nayar [1] attached a video camera to a conventional high-resolution still camera to facilitate PSF estimation. The hybrid camera system was extended by Tai *et al.* [24] to compute general motion blur with optical flow. Li *et al.* [17] developed a three-camera system to remove object motion blur and restore scene depth. In [28], Yuan *et al.* took an additional noisy-unblurred image to make PSF estimation robust. Two blurred images were also used in [20] for motion deblurring. Assuming constant-velocity camera motion, Li *et al.* [18] used a set of motion-blurred frames to create a deblurred panoramic image.

Taking account of camera motion, Tai *et al.* [25] used a projective motion model. Joshi *et al.* [9] proposed a hardware solution with motion inertia sensors to estimate camera shake. Simplified models with 3 DOFs were introduced in [26, 6] to handle non-uniform blur. Hirsch *et al.* [7] tackled smoothly varying PSFs using overlapping windows with a filter flow implementation.

These methods do not consider the fact that many scenes comprise multiple depth layers, which cause abrupt PSF change. The work presented in [8, 23, 2], on the other hand, focused on partial detection or removal of blur caused by object motion. Contrary to these techniques, we deal with depth-variation-involved motion deblurring using prevailing stereo cameras.

In non-blind deconvolution where the PSF is known,

sparse natural image priors were used to improve image restoration [14, 22, 12]. Iteratively Re-weighted Least Squares (IRLS) [14] and variable substitution schemes [22, 12] were employed to constrain the solution. To suppress the ringing artifacts, Yuan *et al.* [29] proposed a progressive procedure to gradually add back image details.

3. Depth-Involved Blur Model

In this section, we analyze the motion blur model when taking depth and two-view stereopsis into consideration.

Depth-Related Blur If the camera center moves Δc during image capturing, the pixel in the image plane corresponding to a point with depth z moves $\Delta p = f/z \cdot \Delta c$ where f is the focal length. The point trajectory in the image forms the blur kernel. Translational depth-involved blur can be written as

$$B = I \otimes k^z + n, \tag{1}$$

where B is the observation, I is the latent image, and n represents inevitable noise. k^z is the PSF (or blur kernel) corresponding to depth z.

Two pixels x and y theoretically have their corresponding PSFs $k^{z(x)}$ and $k^{z(y)}$ *linearly scaled* in a ratio proportional to $z(x)/z(y)$. However, this property does not always hold practically due to possible depth and PSF estimation errors, making directly constructing a set of linearly-scaled PSFs for different depth layers highly unstable. We, therefore, do not rely on this assumption and only weakly require that PSFs for layers with similar depth values do not vary significantly, which can be well satisfied empirically. In our system, PSFs for different regions are separately and hierarchically estimated in a tree structure.

Two-View Stereopsis To infer usable depth layers for deblurring, we employ the two-view stereo configuration. In traditional stereo matching, unblurred matching image I_m and reference I_r are related by disparity, expressed as

$$I_m(x) = I_r(x + \frac{f}{z(x)}\Delta b) = I_r(x + d_m(x)), \tag{2}$$

where Δb is the baseline distance between the two camera projection centers, $z(x)$ denotes the depth of x in the matching view, and $d_m(x)$ is the disparity of the point that projects to x in I_m [21]. Disparity d is inversely proportional to depth. In what follows, for simplicity's sake, we use d to alternatively represent disparity and depth.

Eq. (2) holds for translation blurred image pairs in a depth layer. It is easy to verify that

$$I_m(x) \otimes k^z = I_r(x + d_m(x)) \otimes k^z, \tag{3}$$

if x is in depth layer z and is not close to the boundary. Empirically, slight camera rotation would not completely
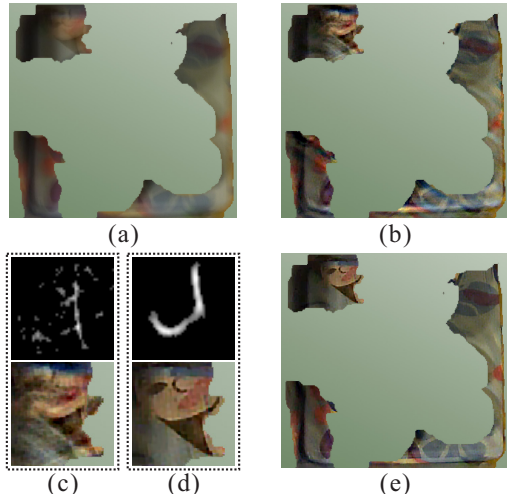
(a) (b)
(c) (d) (e)

Figure 2. Small-region deblurring. (a) One small-size depth layer. (b) Layer deblurring result of Xu and Jia [27]. (c)-(d) PSFs and close-ups of (b) and (e). (e) Our deblurring result.

fail the correspondence. All examples shown in this paper are captured by a hand-held camera.

Major Difficulties Ubiquitous depth variation produces complex depth layers. Traditional PSF estimation on small regions is not robust. We propose a region-tree scheme to reliably compute PSFs even for these small regions.

Fig. 2(a) shows one depth layer extracted from the image in Fig. 1(a). The results of the deblurring method [27], shown in Fig. 2(b)-(c), contain noticeable visual artifacts owing to the small region size. (d) and (e) are results of our method to be detailed below.

4. Our Approach

We estimate depth layers and layer-specific PSFs, which are then used to deblur the whole images. The algorithm is given in Table 1. Briefly, the disparity maps for both views are computed from the blurred image pair and are refined with the restoration results in the second pass. For robust PSF estimation, we propose a region tree scheme, corresponding to Steps 2-4 in Table 1.

4.1. Disparity Estimation

Initially we have no unblurred images for disparity estimation. We thus down-sample the two blurred views with ratio 2 (denoted as $B_m^\downarrow$ and $B_r^\downarrow$) to roughly reduce blur. Based on the correspondence of blurred pixels expressed in Eq. (3), we estimate coarse disparity maps $d^\downarrow$ by minimizing objective functions

$$E(d_r^\downarrow) = \|B_m^\downarrow(x - d_r^\downarrow(x)) - B_r^\downarrow(x)\|^2 + \gamma_d \min(\nabla d_r^{\downarrow 2}, \tau)$$
$$E(d_m^\downarrow) = \|B_m^\downarrow(x) - B_r^\downarrow(x + d_m^\downarrow(x))\|^2 + \gamma_d \min(\nabla d_m^{\downarrow 2}, \tau),$$
(4)

Depth-Aware Motion Deblurring
First-Pass Estimation
1.   Stereo-matching-based disparity estimation.
2.   Region-tree construction.
3.   PSF estimation for top-level regions in the trees.
3.1.   PSF refinement.
3.2.   PSF candidate generation and selection.
4.   If it is not the leaf-level, propagate PSF estimates to lower-level regions and go to Step 3.1.
5.   Blur removal given the PSF estimate.
Second-Pass Estimation
6.   Update disparities based on the deblurred images.
7.   PSF estimation by repeating Steps 2-5.

Table 1. Method Overview
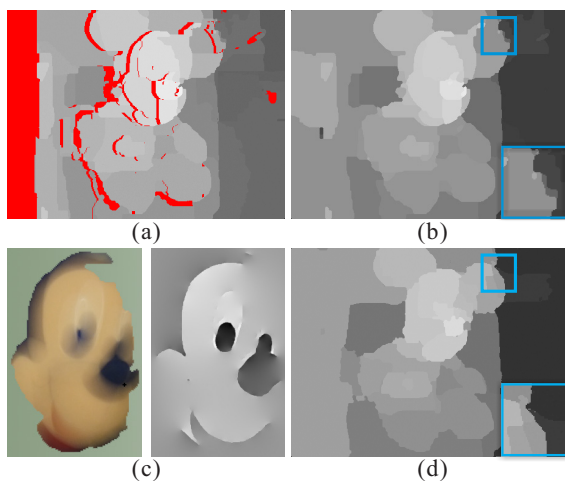

(a) (b)
(c) (d)

Figure 3. Disparities for the reference view. (a) Initial disparity map with many small layers. Detected occlusion is marked as red. (b) Quantized disparity map after hole filling. (c) One leaf-node region in the tree together with the predicted ∇S_r. (d) Refined disparity map in the second pass.

where $\gamma_d \min(\nabla d^2, \tau)$ is the truncated smoothing function for regularization. Blurred images $B_m^\downarrow$ and $B_r^\downarrow$ can be used for pixel matching inside each depth layer given Eq. (3). After initial stereo matching solved by graph-cuts [11], $d_r^\downarrow$ and $d_m^\downarrow$ are up-sampled to the original resolution to form coarse depth maps. Note that this process cannot handle well region boundaries due to the violation of the stereo matching condition. We therefore exclude boundary pixels for the following PSF estimation for robustness' sake. They will be refined in the second pass with the updated estimates, as indicated in Steps 6-7 in Table 1.

After stereo matching, we perform cross-checking [21], which makes use of one-to-one mapping between correspondence, to detect occlusion. To *fill holes* caused by removing the unreliable occlusion disparities, based on the fact that only the surface with relatively small disparities can be occluded, we assign the smallest neighborhood dis-
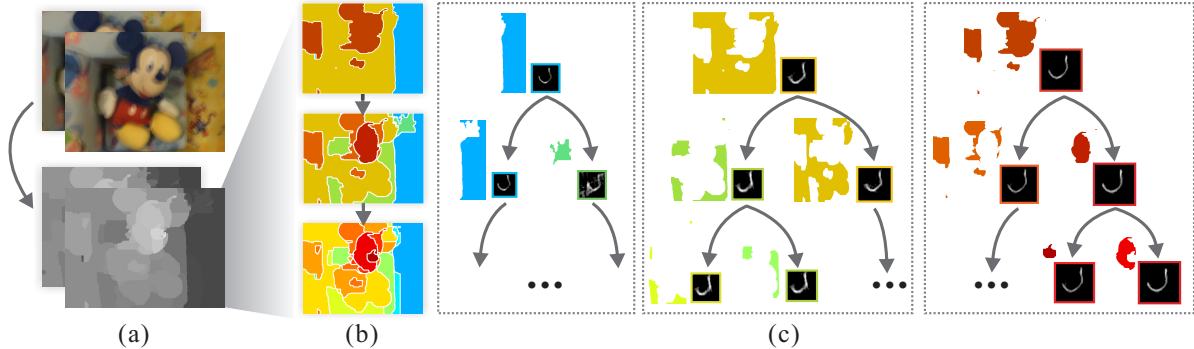
Figure 4. System illustration. (a) Stereo image pair for depth-aware PSF estimation. (b) A three-level region tree. (c) Top-down PSF estimation in the tree.
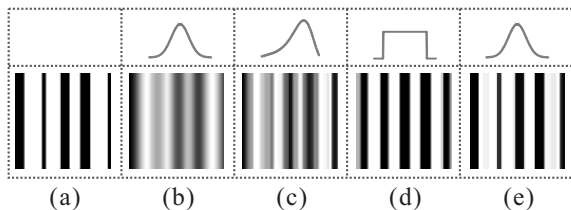

Figure 5. Importance of optimization initialization. (a)-(b) Input image and its blur degraded version (Gaussian PSF shown on top). (c) Deblurring result using the state-of-the-art method [27]. (d) Restored image using a simple box PSF. (e) Result of our method with the box PSF initialization.

parity values. One initial disparity map is shown in Fig. 3(a) given the input images in Fig. 1(a)-(b).

The initial disparity maps may contain many layers, excessive for PSF estimation. We quantize disparity values until at most l regions remain, where l is set to the approximate PSF width or height, which is larger. We denote the quantized maps as d_r and d_m respectively, and the disparity labels as $\{0, 1, ..., l-1\}$. One example is shown in Fig. 3(b) with $l = 25$. Note that the boundary depth is erroneous, which will be updated in the second pass.

4.2. Depth-Dependent PSF Estimation

We estimate a blur kernel for each depth layer. The novelty is to handle small-size layers, making estimating PSFs in them robust. This strategy is based on common wisdom that variable initialization plays a key role in non-convex optimization. The closer the initial point is to the optimal point, the better result can be obtained even without very effective constraints. For challenging motion deblurring, if we can provide a reasonable starting kernel, the refinement to an optimal status is very possible in small regions.

Here we show a toy example in Fig. 5. The input image in (a) is blurred with a Gaussian PSF, resulting in (b), which mixes up lines. It could fail state-of-the-art deblurring methods due to the small image size and lack of structure information. The result of [27] is shown in (c). This ex-

ample, however, can be nicely dealt with even with a rough box kernel initialization, because it at least separates a few blended structures, as shown in (d). The final deblurring result of our method is shown in (e) using the box PSF initialization. It is notably better than the result in (c).

We thus tackle kernel estimation in small regions by collecting coarse candidates, which convert image deblurring to semi-blind deconvolution. The original challenging problem also boils down to *finding a suitable initial PSF* for a depth layer.

We propose a general hierarchical estimation scheme. As discussed in Sec. 3, similar-depth layers are also with alike blur PSFs. Accordingly, we merge layers with similar depth values to form larger regions, enabling reliable blur kernel estimation. The PSF results are then used as initial estimates for smaller-region PSF refinement. We will provide a reliable PSF selection algorithm to further reduce errors.

4.2.1 Region Tree Construction

We construct a binary region trees for each view. As all region nodes in quantized disparity maps d_r and d_m are corresponding, in what follows, we only discuss how to build a tree in d_r.

Leaf nodes are constructed in the first place, each corresponding to a depth layer in d_r. With a total of l labels, l leaf nodes exist. Then we apply layer merging, as illustrated in Fig. 4(b). In the leaf level, for each depth layer $S(i)$ with depth i, it is merged with $S(j)$ if i and j are neighboring integers and $i = \lfloor j/2 \rfloor \cdot 2$. This makes nodes $S(0)$ and $S(1)$ be merged to $S(\{0,1\})$, $S(2)$ and $S(3)$ to $S(\{2,3\})$, and so forth. $S(\{0,1\})$ is the parent node of $S(0)$ and $S(1)$ in the tree. Similarly, we merge $S(\{0,1\})$ and $S(\{2,3\})$ to form $S(\{0,1,2,3\})$ in an even higher level. Repeating this process yields a binary tree with multiple levels. In practice, the finest three levels are enough for PSF estimation. The highest level we construct in Fig. 4(c) has a total of 3 regions/nodes.

With the corresponding trees in d_r and d_m, we estimate regional blur kernels in a top-down fashion, ultimately to reliably infer PSFs for all leaf nodes. As aforementioned, similar-depth layers have similar PSFs. Merging them for PSF estimation thus will not produce wildly inappropriate results. We describe in what follows the method to compute PSF for one middle- or low-level layer. It involves optimization by taking the parent-node PSF result as initialization, and PSF selection to reject remaining outliers. In the top level with large regions, PSFs are computed using the method of [27].

4.2.2 Iterative PSF Computation

For each middle- or leaf-level node, we denote the corresponding regions as S_r and S_m in the reference and matching views. Their parent coarse PSF estimates are available. We then construct a salient edge map for each current node to guide PSF update. The edge maps are denoted as ∇S_r and ∇S_m, similar to the P map defined in [3]. The only difference is that ∇S_r and ∇S_m are sets for specific depth layers. One example of ∇S_r is shown in Fig. 3(c).

Joint PSF Estimation   We base our PSF estimation on image gradient blur models with the Tikhonov regularization. Our objective function is defined jointly on the reference and matching views as

$$\mathrm{E}(k) = \sum_{i \in \{r,m\}} \|(\nabla S_i) \otimes k - \nabla \mathrm{B}_i\|^2 + \gamma_k \|k\|^2, \qquad (5)$$

where $\nabla \mathrm{B}$ is the gradient map of B. For pixels not belonging to region S, we set their derivatives to zeros. There exists a closed-form solution of Eq. (5):

$$k = F^{-1} \left(\frac{\sum_i \overline{F_{\partial_x S_i}} F_{\partial_x} F_{B_i} + \sum_i \overline{F_{\partial_y S_i}} F_{\partial_y} F_{B_i}}{\sum_i F_{\nabla S_i}^2 + \gamma_k F_{\mathbf{1}}^2} \right), \qquad (6)$$

where F_{S_i} represents the Fourier transform of S_i, $\overline{F}$ denotes the complex conjugate operator, and F^{-1} is the inverse Fourier transform. $F_{\mathbf{1}}$ is the Fourier transform of a delta function with a uniform energy distribution. $F_{\nabla S_i}^2$ is expressed as $\overline{F_{\partial_x S_i}} \cdot F_{\partial_x S_i} + \overline{F_{\partial_y S_i}} \cdot F_{\partial_y S_i}$.

Eq. (6) jointly estimates PSFs in the stereo images, which is more robust against noise than deblurring using a single image. We show in Fig. 6(a) the initial and refined PSFs of a region in the middle level. The blur kernel is improved from its initial estimate passed from the parent node.

4.2.3 Candidate PSF Selection

The PSF estimates can occasionally be erroneous especially for small regions. One failure case is shown in Fig. 6(b). To detect and correct them, after completing PSF estimation for all nodes in one level of the tree, we detect obviously
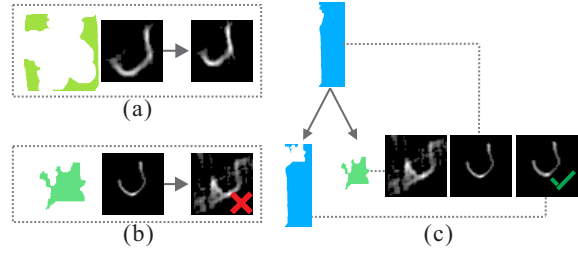

Figure 6. PSF estimation and selection. (a)-(b) Two PSF refinement examples. From left to right are depth layers, initial PSFs from the parent nodes, and the refined PSFs. (c) PSF selection from the current estimate, parent PSF, and neighboring PSFs. 3 candidates are shown. The finally chosen PSF is marked.

incorrect PSFs and improve them using a new PSF selection algorithm.

Wrong PSF estimates are in general noisy and have dense values. We use the PSF entropy

$$H(k_i) = -\sum_{x \in k_i} x \log x$$

to detect them. If one PSF has its entropy notably larger than its peers in the same level, we label it as "unreliable". It is achieved empirically by determining whether the difference between $H(k_i)$ and the mean value in this level is larger than a threshold (20% of the mean empirically) or not. Note that this scheme could misclassify less-sparse kernels caused by long exposure as unreliable, which however is *not* a problem. In our method, the "unreliable" PSFs are not simply dropped, but instead are reconsidered in a PSF selection scheme, together with other candidates obtain in the following two ways.

- We pick the parent-node PSF estimate as one candidate.
- PSF of the sibling node labeled as *reliable* is another candidate.

One example is shown in Fig. 6(c) with three candidates. We select a suitable PSF from them for the current region. It has been known as a very difficult problem given only the blur observation. One naive method is to compare the *reconstruction errors*

$$\varepsilon = \|\mathrm{I} \otimes k - \mathrm{B}\|^2, \qquad (7)$$

by taking the candidate kernels k into the metric. It is also known as the likelihood evaluation, favoring a delta kernel, even with sparse priors enforced on image gradients [14].

One example is presented in Fig. 7, where each of the two patches is blurred with a Gaussian kernel and a motion blur PSF. Four blurred images are resulted in. We plot their reconstruction errors using the red curves by taking PSFs of different scales into Eq. (7). It is noticeable that small
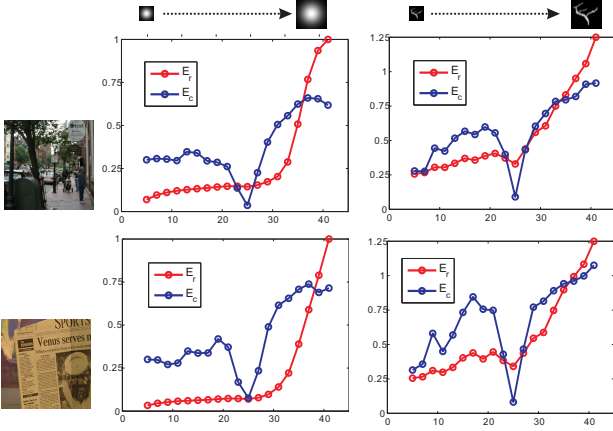
Figure 7. PSF selection. Two regions, each blurred with two PSFs (size 25×25), generate a total of four examples. The charts (x axis: PSF size. y axis: normalized error) show the reconstruction errors (in red) and our correlation values (in blue) when applying kernels of different sizes to deblurring.

PSFs unexceptionally yield small errors. The ground truth kernels (with size 25×25) do not correspond to even local minima in the plots.

To address this issue, prior work either employs the coded-aperture technique [14], or computes a marginalized probability based on a sufficiently large image [15]. They are not usable in our system. We propose a new PSF selection scheme making use of the *shock filtering invariance* nature of unblurred natural images. The only requirement is that the latent images contain salient edges, which can be easily satisfied.

For each PSF k, we restore a latent image (or region) I^k by minimizing the objective function

$$E(I^k) = \|I^k \otimes k - B\|^2 + \gamma \|\nabla I^k\|^2, \qquad (8)$$

where $\|\nabla I^k\|^2$ is a simple regularization term. It has a closed-form solution. If k is correct, the recovered I^k should faithfully contain the salient edges inherent in the image. To find out how credible the deconvolution result is, we apply shock filtering [19] to each slightly Gaussian smoothed I^k, which yields

$$\tilde{I}^k = I^k - \text{sign}(I^k_{\eta\eta}) \|\nabla I^k\|, \qquad (9)$$

where $I_{\eta\eta}$ is the second-order derivative along the gradient, $\text{sign}(\cdot)$ is a sign operator. Smoothing I^k helps reduce noise but does not greatly affect the shock filtering result on significant edges. I^k is converted to a piecewise constant image/region after applying Eq. (9). The discontinuities are at the inflection points in a total-variation-preserving fashion, which remarkably change the shape of originally *smooth* structures. The step-like edges, in contrast, will not be largely altered.

To evaluate image restoration by Eq. (8) when applying different kernels k, we propose computing the *correlation*
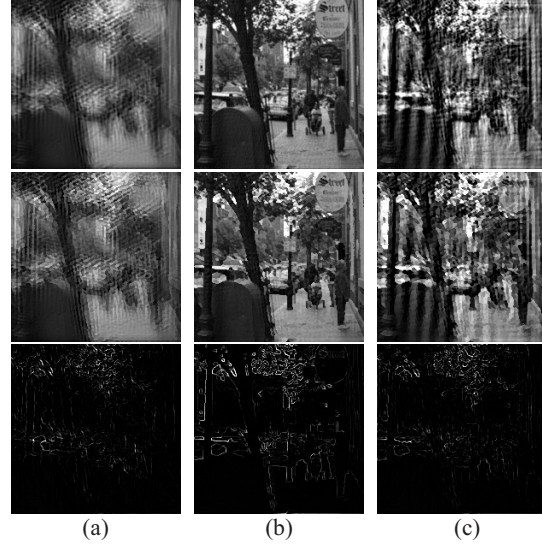

(a)   (b)   (c)

Figure 8. Shock filtering and image correlation. Top row: I^k restored with a PSF smaller than the ground truth (21x21) (a), the correct PSF (25x25) (b), and a large PSF (29x29) (c). Middle row: respective $\tilde{I}^k$s. Bottom row: the correlation map linearly scaled for visualization. The correctly deblurred image (b) yields a correlation map with the most salient structures.

of gradient magnitudes between I^k and $\tilde{I}^k$. Only salient edges, after shock filtering, will not be changed significantly, which yield large correlation values. Correlation-based energy $E_c(k)$ is computed as

$$E_c(k) = 1 - \text{corr}(\|\nabla I^k\|_2, \|\nabla \tilde{I}^k\|_2), \qquad (10)$$

where $\text{corr}(\cdot)$ is the correlation operator expressed as

$$\text{corr}(X, Y) = \frac{\mathbf{E}(X - \mu_X)\mathbf{E}(Y - \mu_Y)}{\sigma_X \sigma_Y}.$$

$\mathbf{E}$ is the expectation operator. σ and μ denote signal standard deviation and mean.

The effectiveness of $E_c(k)$ to evaluate the deconvolution quality is illustrated in Fig. 7 using the blue plots. It is obvious that *ground truth PSFs correspond to the minima of* $E_c(k)$. The success is fully explainable. For PSFs smaller than the correct ones, the restored image still blurs, as shown in Fig. 8(a). In this case, almost all edges are altered during shock filtering, making the two images less correlated. For larger PSFs or those with errors, as shown in Fig. 8(c), ringing artifacts and other structural problems are introduced in image restoration, which also ruin edges and accordingly reduce the correlation values.

We show in Fig. 6(c) the PSF candidates. The associated E_cs are 0.38, 0.14, and 0.12 respectively. Wrong PSF estimates can be quickly and robustly rejected in this process. We further show in Fig. 9(a) and (b) the comparison of deconvolution results using the refined kernels in the top and bottom levels of the tree. The quality improvement also indicates that regional PSFs are ameliorated hierarchically.

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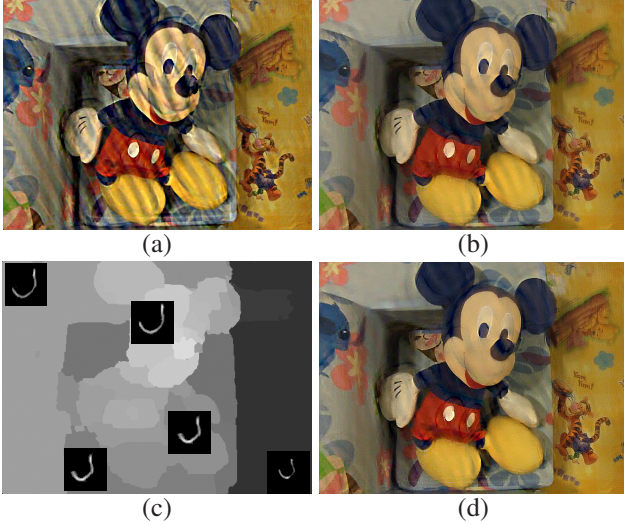
(c)   (d)

Figure 9. Comparison of deconvolution results after kernel estimation/refinement in the top level (shown in (a)) and the bottom level (shown in (b)). The depth layers computed in the second pass along with PSFs are shown in (c). The final restoration result is shown in (d).

4.3. Image Restoration and Results

We deblur the two views using the PSF estimates in the leaf nodes of the region trees (Step 5 in Table 1) and globally solve the objective function

$$E(I) = \|I \otimes k^d - B\|^2 + \gamma_f \|\nabla I\|^{0.8} \qquad (11)$$

with respect to all k^d estimates in the reference/matching view. Because deconvolving region boundary pixels is generally error-prone, γ_f is set 3 times larger for pixels with their distance to the boundary smaller than the kernel size. This trick can suppress visual artifacts. Eq. (11) is solved in the spatial domain using reweighted least squares (IRLS).

After the first-pass deblurring, we apply stereo matching to the deconvolved images to update depth maps (Step 6 in Table 1). One result is shown in Fig. 3(d). Then PSF estimation and refinement are applied again in the second pass. The updated depth helps improve PSF estimation, reducing ringing artifacts in the restored images, as shown in Fig. 9(c)-(d).

We use parameters $\gamma_d = 20$, $\gamma_k = 1$ and $\gamma_f = e^{-3}$ in experiments. Iterative PSF computation uses the same parameters as those in [3]. For an input image with size 1024×768, our Matlab implementation takes less than 6 minutes to estimate the depth maps, PSF set, and the restored images for both views.

A scene with large depth variation is shown in Fig. 10. The depth map contains many irregular segments, making it difficult for prior motion deblurring methods to produce comparable results. There exists a large structural difference between the two PSFs shown in the corners of (b).

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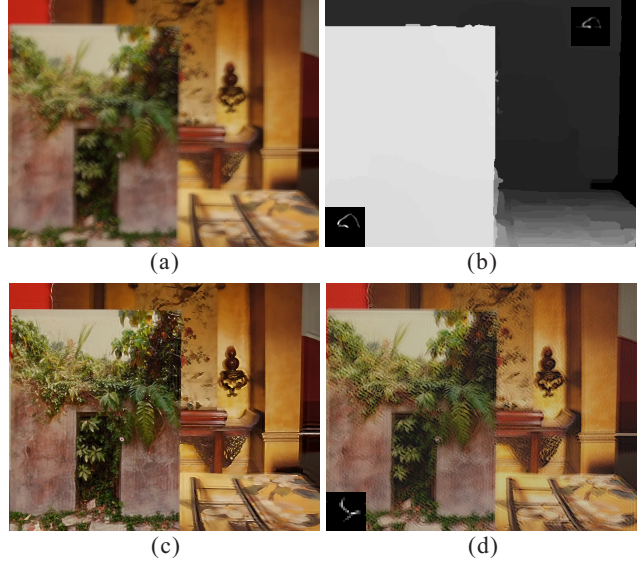
(c)   (d)

Figure 10. Depth-dependant motion deblurring. (a) Reference view from a hand-held stereo camera. (b) Our computed depth/disparity map. Two regional PSFs are shown. (c) Our deblurring result. (d) Restored image using the state-of-the-art deblurring method [4].

5. Single-image Non-uniform Deblurring

Our region tree based estimation framework is a general one, readily applicable to single image non-uniform motion deblurring when depth variation is not a critical issue. In this case, we directly construct a region quad-tree in the input image in a top-down manner, yielding 4 rectangular children regions for each node through regular partitioning, as shown in Fig. 11(a). Each child region is of one fourth of the size of its parent. This partitioning process continues until leaf node regions are small enough. The minimum size is empirically set to 200×200 pixels. The constructed tree is similarly employed in region-based PSF estimation using our hierarchical computation and selection scheme. Our only assumption is that PSF variation within a small region is not significant, which can be easily satisfied.

Given the image (Fig. 11(b)) provided by Joshi et al. [9], our result in this example is with high quality, comparable to that produced with the approach requiring additional hardware [9]. Note that our method does not distinguish defocus and motion blur and only requires that the PSF varies smoothly. It therefore can remove part of the defocus blur, as compared in the close-ups. More examples, as well as comparisons with other state-of-the-art methods, are provided in the project website.

Limitation Although our region-tree based blur kernel estimation and refinement scheme is general, we handle primarily translational camera motion, considering complex and abrupt depth variation. When depth change is not a
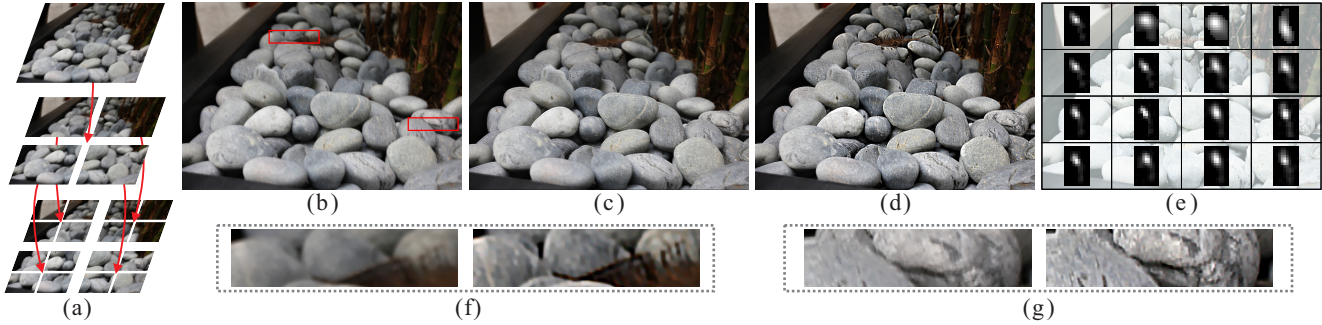
Figure 11. Non-uniform Deblurring. (a) Hierarchical region partitioning. (b) Blurred image provided in [9]. (c) Result of [9] with the facilitation of hardware sensors. (d) Our result. (e) Our regional PSF estimates. (f)-(g) close-ups.

critical issue, single image non-uniform deblurring can be yielded. Reducing more the estimation artifacts near region boundaries in deconvolution and improving the accuracy of depth computation are directions in our future work.

References

[1] M. Ben-Ezra and S. K. Nayar. Motion-based motion deblurring. *PAMI*, 26(6), 2004.

[2] A. Chakrabarti, T. Zickler, and W. T. Freeman. Analyzing spatially-varying blur. In *CVPR*, 2010.

[3] S. Cho and S. Lee. Fast motion deblurring. *ACM Trans. Graph.*, 28(5), 2009.

[4] 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), 2006.

[5] Fujifilm. Fujifilm finepix real 3d w1, 2009.

[6] A. Gupta, N. Joshi, C. L. Zitnick, M. F. Cohen, and B. Curless. Single image deblurring using motion density functions. In *ECCV (1)*, 2010.

[7] M. Hirsch, S. Sra, B. Schölkopf, and S. Harmeling. Efficient filter flow for space-variant multiframe blind deconvolution. In *CVPR*, 2010.

[8] J. Jia. Single image motion deblurring using transparency. In *CVPR*, 2007.

[9] N. Joshi, S. B. Kang, C. L. Zitnick, and R. Szeliski. Image deblurring using inertial measurement sensors. *ACM Trans. Graph.*, 29(4), 2010.

[10] N. Joshi, R. Szeliski, and D. J. Kriegman. Psf estimation using sharp edge prediction. In *CVPR*, 2008.

[11] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions via graph cuts. In *ICCV*, 2001.

[12] D. Krishnan and R. Fergus. Fast image deconvolution using hyper-laplacian priors. In *NIPS*, 2009.

[13] D. Krishnan, T. Tay, and R. Fergus. Blind deconvolution using a normalized sparsity measure. In *CVPR*, 2011.

[14] A. Levin, R. Fergus, F. Durand, and W. T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM Trans. Graph.*, 26(3), 2007.

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

[16] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Efficient marginal likelihood optimization in blind deconvolution. In *CVPR*, 2011.

[17] F. Li, J. Yu, and J. Chai. A hybrid camera for motion deblurring and depth map super-resolution. In *CVPR*, 2008.

[18] Y. Li, S. B. Kang, N. Joshi, S. Seitz, and D. Huttenlocher. Generating sharp panoramas from motion-blurred videos. In *CVPR*, 2010.

[19] S. Osher and L. I. Rudin. Feature-oriented image enhancement using shock filters. *SIAM J. Numer. Anal.*, 27, 1990.

[20] A. Rav-Acha and S. Peleg. Two motion-blurred images are better than one. *Pattern Recognition Letters*, 26(3), 2005.

[21] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1-3), 2002.

[22] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. *ACM Trans. Graph.*, 27(3), 2008.

[23] Q. Shan, W. Xiong, and J. Jia. Rotational motion deblurring of a rigid object from a single image. In *ICCV*, 2007.

[24] Y.-W. Tai, H. Du, M. S. Brown, and S. Lin. Image/video deblurring using a hybrid camera. In *CVPR*, 2008.

[25] Y.-W. Tai, P. Tan, and M. S. Brown. Richardson-lucy deblurring for scenes under a projective motion path. *PAMI*, 33(8), 2011.

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

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

[28] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum. Image deblurring with blurred/noisy image pairs. *ACM Trans. Graph.*, 26(3), 2007.

[29] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum. Progressive inter-scale and intra-scale non-blind image deconvolution. *ACM Trans. Graph.*, 27(3), 2008.