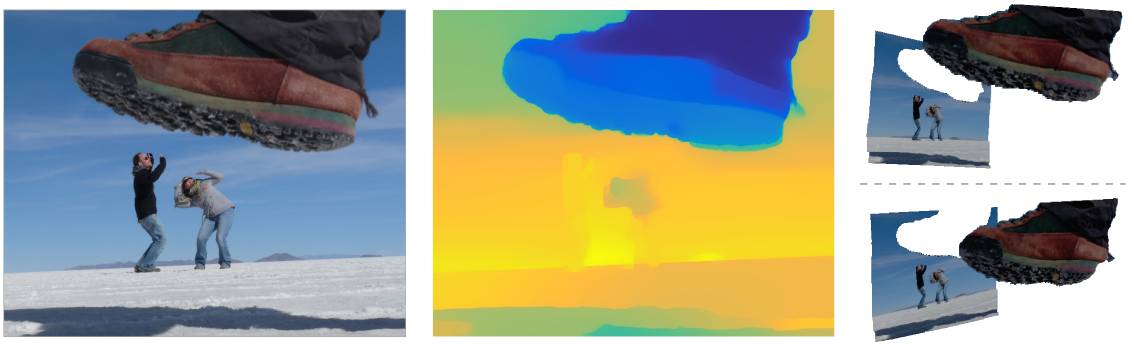
Break Ames Room Illusion: Depth from General Single Images

Jianping Shi[1]* Xin Tao[1]* Li Xu[2]* Jiaya Jia[1]*
[1] The Chinese University of Hong Kong [2] SenseTime Group Limited



(a) Input (b) Blur Scale Map (c) Two viewpoints in 3D

Figure 1: *The photo in (a) creates an illusion that a giant is going to tread on people. We infer depth from this single image as shown in (b) using special cues to be discussed later, which interprets this "forced perspective image" correctly from a geometry-layer point of view in (c).*

Abstract

Photos compress 3D visual data to 2D. However, it is still possible to infer depth information even without sophisticated object learning. We propose a solution based on small-scale defocus blur inherent in optical lens and tackle the estimation problem by proposing a non-parametric matching scheme for natural images. It incorporates a matching prior with our newly constructed edgelet dataset using a non-local scheme, and includes semantic depth order cues for physically based inference. Several applications are enabled on natural images, including geometry based rendering and editing.

Keywords: small-blur estimation, depth from defocus, single-image depth, out-of-focus

1 Introduction

Pictures remove depth information. Human visual perception follows semantic inference to roughly conjecture which object is close when one watches monocular videos or images, which has been commonly exploited in illusion generation involving "forced perspective". For example, in *The Lord of the Rings* movies, characters seem bigger or smaller than what they actually are by using special settings and different distances during the shot.

The famous experiment named *the Ames room illusion* makes a person standing in one corner seem to be a giant, while another one in the other corner appears to be a dwarf, when the room is viewed with one eye through a pinhole – i.e., cues from stereopsis are eliminated. An illusion is given in Fig. 1(a) where a gigantic foot almost tramples on the persons. But actually they are at different distances.

In previous work, inferring depth from monocular data was achieved by combining multiple cues [Saxena et al. 2009; Eigen et al. 2014; Ladicky et al. 2014], estimating obvious out-of-focus blur [Bae and Durand 2007; Zhu et al. 2013], using specially designed hardware [Levin et al. 2007], or by learning [Karsch et al. 2012]. The success of these approaches relies on their respective requirements or assumptions. For instance, learning-based methods need large amounts of training data; blur estimation requires purposely generated defocus. They may not work well when the scene is different from the assumed types, such as the example in Fig. 1(a). We provide more discussions in Section 2.

In this paper, we address this challenging illusion problem with a monocular cue. By exploring small-scale defocus properties which will be detailed in Section 4, we circumvent depth learning, special hardware, strong defocus or stereo configuration. Our method works on common photos produced from different devices, including mobile phones. In the example in Fig. 1, our estimated distances for the persons and the foot are clearly different, based on which further geometrical scene understanding can be accomplished.

Our main contributions are as follows. First, we provide analysis of the optical sharpness property and its visual phenomena in photos. Second, we propose an effective non-parametric estimator, together with an edgelet primitive dataset, to robustly recognize spatial sharpness variation. Our estimation is robust against visual artifacts. Third, we propose a solver with non-local smoothness connection, and establish correspondence between the estimated blur scale map and the inherent depth estimate. Finally, we demonstrate a few applications given single input images, including refocusing, stereopsis synthesis and geometry-aware image editing.

2 Related Work

We review previous work on depth estimation from monocular natural image/video and explain their respective requirements. While stereo matching [Scharstein and Szeliski 2002] and depth from active sensors [Khoshelham and Elberink 2012] are popular, they re-

*e-mail:{jpshi,xtao,xuli,leojia}@cse.cuhk.edu.hk

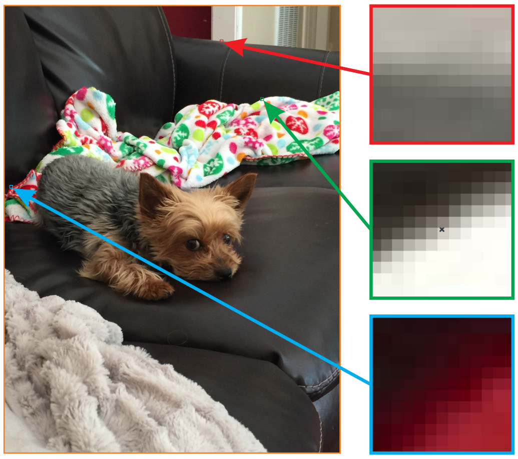


Figure 2: *A nicely focused image taken by an iPhone 6 camera and its closeup regions. Subtle defocus blur effect is noticeable when viewed closely. The top two patches have about 5 pixel width blur, where the bottom one is with about 8 pixels.*

quire multiple inputs or special hardware, and thus are not applicable to our problem.

Data-driven Single Image Depth Estimation Several methods incorporate various data-driven cues to estimate depth. Wu *et al.* [2008] presented an interactive system to infer surface normal with shading information. Occlusion is utilized in [Hoiem et al. 2007] for coarse depth order inference. Recently, RGBD data [Karsch et al. 2012; Eigen et al. 2014] were involved for depth estimation for specific scenes. Su *et al.* [2014] built a 3D object dataset for object depth inference. Karsch *et al.* [2012] transferred depth of similar scenes from video data. These methods need user interaction, special computation environment, or a large set of data from similar scenes for reference while our solution does not.

Depth from Defocus Depth from defocus is a well-studied problem. Generally two or more images captured from the same position but with different focus are used to infer depth. Early work like [Subbarao and Surya 1994] is a convolution/deconvolution transform method. Watanabe and Nayar [1998] introduced a class of broadband operators for passive depth from defocus. Ziou and Deschenes [2001] proposed a local image decomposition technique. Schechner and Kiryati [2000] compared depth-from-defocus and stereo methods. Our method takes only one natural image for depth layer inference, which can handle more data.

Coded Aperture In computational photography, coded aperture is popular to accurately estimate depth in order to produce all-in-focus images. The technique has been applied to image deconvolution [Levin et al. 2007] and light field refocusing [Veeraraghavan et al. 2007; Liang et al. 2008]. Differently designed coded apertures were also analyzed [Zhou and Nayar 2009] and extended in [Zhou et al. 2009]. Cossairt [2010] inserted an optical diffuser between camera lens and the sensor to extend depth-of-field. While these computational-camera approaches work very well, they require necessary hardware modification.

Defocus Blur Estimation Several previous methods also directly estimate defocus blur. Elder and Zucker [1998] assumed a step edge and a Gaussian blur kernel for defocus blur estimation. The first and second order derivatives from steerable Gaussian basis filters were used to calculate the center line of edges and blur responses. Saxena *et al.* [2005] proposed a supervised learning approach by comparing filter responses of image patches. Bae and Durand [2007] applied bilateral filtering to remove outliers, and employed an MRF to estimate a dense defocus map. Tai and Brown [2009] compared edge gradient magnitudes to evaluate the amount of out-of-focus blur.

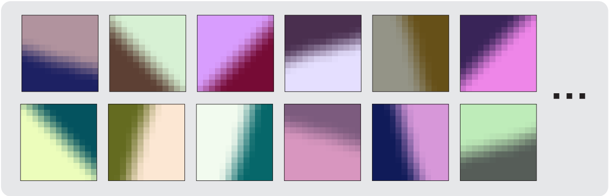


Figure 3: *Example patches affected by small defocus blur for evaluating existing defocus estimation methods.*

The other line is to model local statistics. Shi *et al.* [2015] detected small blur via sparse reconstruction statistics. Its major goal is to find the slightly blurred region. Its estimation, as a by-product, is not accurate enough. Zhuo *et al.* [2011] measured defocus blur according to the response of Gaussian blur in defocused regions. The method of [Zhu et al. 2013] generalized that of [Chakrabarti et al. 2010] to model the posterior of Gabor filter bank coefficients. These approaches were designed for blurred images, where edges in focused and out-of-focus regions are significantly different. Our data do not satisfy these requirements and thus could degrade their performance. More discussions will be provided in later sections.

3 Our Monocular Depth Cue and Its Analysis

Different from above methods, ours is based on exploiting small-scale defocus blur that exists in almost all photos. The blur kernel is an optical spot caused by a cone of light from a point source, which does not completely focus on the sensor plane. In this regard, all surface points that are not located exactly on the focal plane cause the blurriness. The largest blur scale or equivalently the largest size of the spot, which is still perceived by human as one point at a viewing distance, is used to determine the depth of field (DoF).

The small-scale defocus blur controls the level of sharpness on details, which is taken as our monocular cue for depth estimation. Taking off-the-shelf cameras as an example, Canon 5D Mark III has a pixel pitch of $0.00625mm$. It means a light point could span up to $0.03/0.00625 = 4.8$ pixels, where the 0.03 is the diagonal measure of a 35mm camera format. For the new iPhone cameras, the blur is with diameter about 4.0 pixels. Fig. 2 gives an illustration. The input natural image has small difference on the sharpness level due to blur variation of different locations. This type of change actually provides us surprisingly valuable information for understanding how depth varies.

Our defocus blur estimation differs from general point spread function (PSF) estimation in the following. The general PSF-analysis approaches aim at restoring motion blur and significant defocus PSFs, which are generally stronger blur than what we aim to deal with. Our subtle defocus blur here, on the other hand, is usually caused by optical lens, appears in a smaller scale, and exists nearly in all natural images. We have extensively evaluated existing methods for PSF analysis and found they do not perform satisfyingly on our data mainly due to the fact that structure variation caused by subtle defocus blur is small.

Also since this defocus blur effect is spatially-varying, common spatially-varying deblurring methods deal with motion blur with necessary motion and depth assumptions. In another line, several methods directly estimate blurriness from local patches. They also

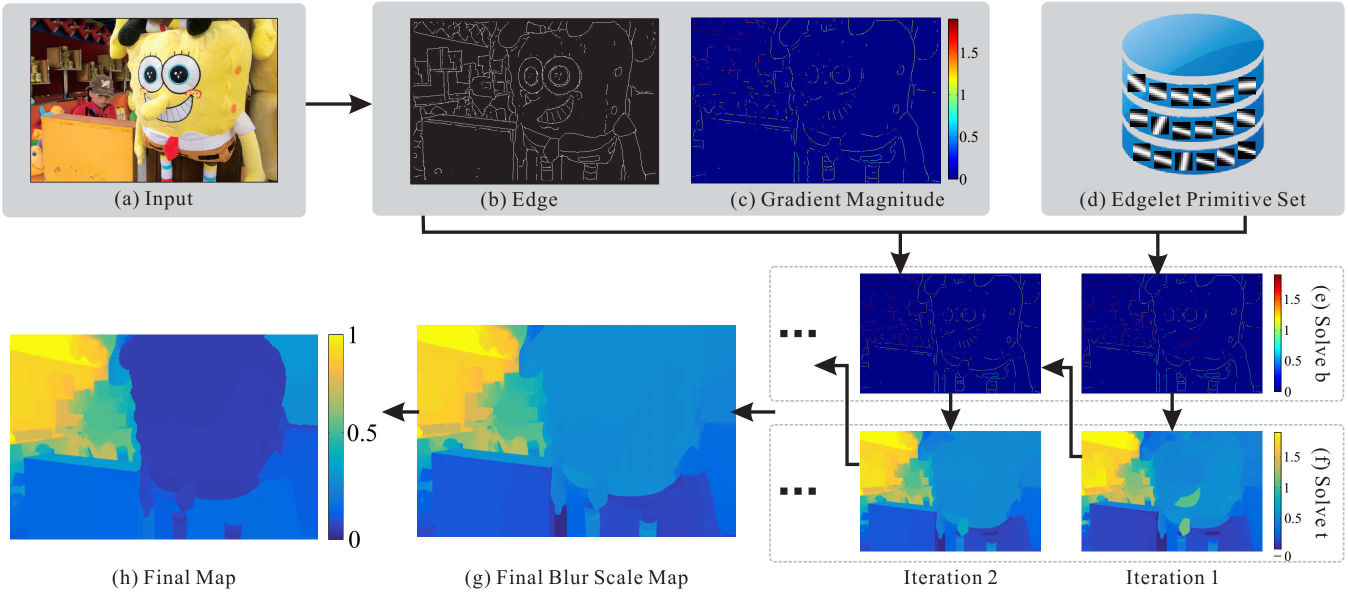


Figure 4: *Overview of our method. We detect edges in (b) from input (a). They are fitted into our matching and optimization framework that includes edgelet primitives (d). During alternating optimization, we solve for the sparse edgelet blur scale map shown in (e) and the global estimates in (f). The final depth map (h) is constructed by another optimization pass involving the final defocus blur scale estimate (g).*

assume blur is strong enough so that the employed metrics, such as gradient edge moments, local statistics and distribution information, can be used.

We conduct an experiment to evaluate these methods. We first synthesize 2,000 different patches that are affected by the this type of blur (a few representatives are shown in Fig. 3). Then we apply local edge gradient metric [Bae and Durand 2007] and patch statistics [Zhu et al. 2013] to these data. The estimated blur scales are with 42% and 33% errors respectively. They manifest it is still difficult to apply existing blur analysis to small blur estimation. Different from these methods, we propose a more suitable matching and optimization framework in what follows to address this problem.

4 Our Solution

Because the scale of general defocus blur is small and it is spatially varying by nature, its estimation on flat regions cannot be reliable. We start our method from edges and analyze the appearance it presents. Then the estimates are refined and propagated to other pixels for dense point inference. Our main steps are explained below and in Fig. 4.

- To estimate the blur scale level, fitting a parametric model on local statistics is not suitable because the blur patterns of camera lens are not spatially differentiable. The Gaussian approximation does not work on such small scales. We resort to non-parametric matching with edgelet primitive data.
- Our edgelet primitive set is constructed by sampling edge segment patches with varying directions, curvatures, and blur scales. It provides basis for preliminary local matching. The matching error is already a useful indicator on whether a patch contains an ideal underlying edge or not.
- With the edge primitive data, we establish a global function for dense scale estimation via matching. Since the objective function is highly non-convex and the matching space is large, we propose an effective numerical solver.

We elaborate on these steps in the following. We note our small-scale defocus blur estimation mainly reveals the change of blur scales among different pixels in one image. This type of spatial difference is essential to understand relative geometric connection among points.

4.1 Edgelet Primitive Set Ω

To capture how edges change under the subtle defocus blur effect, we construct a set of edgelet primitives with respect to scale variation. This edgelet set makes it possible to circumvent a parametric blur generation model that is hardly accurate when blur is small. Also our blur model does not assumes differentiable Gaussian kernels, but instead follows the finding of Watanable and Nayar [1998], which indicates general lens pillbox distributions are in shape of disks. Our experiments show that this disk kernel setup reduces up to 50% errors than Gaussian in our 200 tryouts.

We construct a edgelet primitive set Ω which considers possible edge variations. For different edge segments, we vary three separate dimensions including blur scale b, direction θ, and curvature r. The details on how to establish the set are provided in Section 6. A few examples are shown in Fig. 5. In total, our edgelet primitive set contains over 20,000 samples.

Preliminary Matching with Ω Now for any patch $\Theta(I)$ in an input image I, finding the corresponding edgelet is expressed as

$$\arg\min_{\theta,r,b}\sum_{i\in I}\|f(\Theta(\nabla I_i))-\mathcal{T}(\theta,r,b)\|, \qquad (1)$$

where $\mathcal{T}(\theta,r,b)$ describes the edgelet template with three parameters. $\Theta(I_i)$ is the local patch centered at pixel i. $\Theta(\nabla(I_i))$ denotes gradient magnitudes of $\Theta(I_i)$. $f(\cdot)$ is the normalization function. We normalize the intensity by dividing its 90% percentile to remove outliers. Both $\mathcal{T}(\theta,r,b)$ and $\Theta(\nabla(I_i))$ are vectorized. $\|\cdot\|$ is the $L2$-norm operator to compute Euclidean distance.

Advantages and Following Issues There are several benefits for this matching process. First, instead of computing blur param-

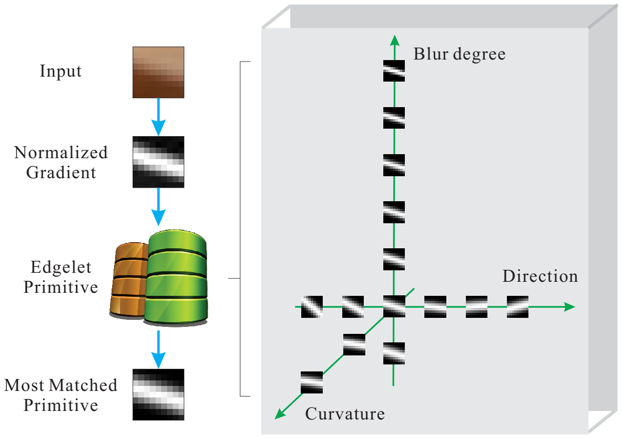


Figure 5: *The edgelet matching process. The input is first normalized in gradients. Then it finds its nearest neighbor in our edgelet set. The edgelet primitives vary in the three dimensions of direction θ, curvature r, and scale b.*

eters by solving an inverse problem, edgelet matching is a forward operation without difficult kernel estimation. Second, the edgelet primitives do not have constraints on the kernels used. We adopt the disk model instead of the Gaussian one. Finally, the matching cost in Eq. (1) reveals how well the template models the local structure. So it can easily screen out edges that are obviously different from our primitives during estimation and guarantee the quality of edges that are used.

To show this matching process can actually ensure edge quality, we gather patches and coarsely classify their contained edges as step, soft, crossing, and texture categories manually in 10 different scales as shown in Fig. 6(a). We apply above matching process to find the nearest neighbors in our primitive set. The average errors for the four categories of edges are respectively 1.02, 2.23, 4.27, and 3.26. The distribution for natural image patch matching error in (b) shows that ideal patches usually have small errors, which can be easily classified from others based on the matching errors.

We note, albeit profitable, applying matching of Eq. (1) to blur scale estimation is not trivial. Only patches with high-quality edges find decent matches, which are actually sparse in our image data. We address this problem in our system and efficiently propagate these sparse evidences to the whole image while rejecting errors caused by imperfect edge detection.

4.2 Our Model

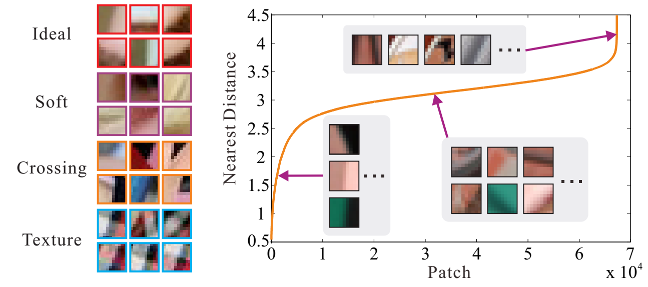
To reject large-error matching results, we reformulate Eq. (1) as probability inference as

$$p(I|\theta,r,b) \propto \prod_i \{\exp\left(-\rho(\|f(\Theta(\nabla(I_i))) - \mathcal{T}(\theta,r,b)\|)\right)^{m_i}\}, \quad (2)$$

where m_i is a binary value for each pixel i, which is 1 if this is an edge point and 0 otherwise.

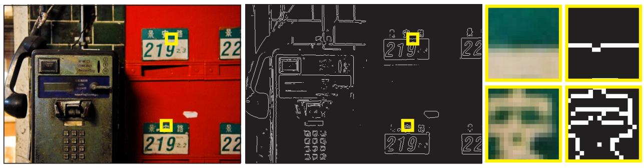
We obtain edge points by the edge detector of [Maini and Sohal 2006], which is robust to noise and outliers. We show an edge map in Fig. 7. Some of them are corners and texture, which cannot be well matched to our edgelet primitives since our data do not include these complicated examples. We thus apply a robust function ρ to suppress the influence of imperfect match. $\rho(x)$ is expressed as

$$\rho(x) = \ln((1-e)\exp(-|x|/\sigma) + e), \quad (3)$$



(a) Edge representative (b) Natural Image Error Distribution

Figure 6: *Edgelet matching errors reflect quality of edges. (a) Edge examples in our four coarse categories. (b) shows the natural image patch matching error distribution. The patches are sorted according to matching errors, where the x-axis values are accumulated patch numbers and y-axis ones are errors for best matchers. Similar types of edges gather and roughly fall into similar ranges of the matching error curve.*



(a) Input (b) Edge (c) Closeup

Figure 7: *Edge detection outputs texture and corners, which are not contained in our edgelet primitive set due to their structure complexity.*

where σ is the error-control parameter, and e is the basis of the natural logarithm. With the robust function in Eq. (3), edge segments with large errors in matching are subdued in further inference.

Joint Posterior with Non-local Affinity The likelihood (2) is incorporated into the final joint posterior as

$$p(\theta,r,b|I) \propto p(I|\theta,r,b)p(\theta)p(r)p(b). \quad (4)$$

For $p(\theta)$ and $p(r)$, since we have no specific prior knowledge, they simply follow uniform distributions. For $p(b)$, as objects occupy a group of pixels, blur scales should smoothly vary for most pixels. We adopt a non-local prior as

$$p(b) \propto \prod_{\{i,j\}\in W} \exp(-w_{ij}|b_i - b_j|^2), \quad (5)$$

where W contains pixel pairs within each local window. $w_{ij} = \exp(-|I_i - I_j|^2/\sigma_I - |x_i - x_j|^2/\sigma_x)$ captures the affinity between two pixels with the bilateral distance measure. It also functions as propagating sparse edge pixels blurriness to all others for input natural images faithful to its edge distribution.

The pairwise weight w_{ij} actually defines the effective range of influence for the selected edge points. Local smoothness with regard to only neighboring pixels cannot effectively propagate the sparse blur estimates across texture and details. In contrast, non-local regularization establishes much stronger connection between pixels according to their distance in the feature space. This becomes especially important to solve our problem where the number of edge estimates is not that large, which is further elaborated on in Section 6.

Further Refinement Now the maximum of the joint posterior becomes

$$\arg\max_{\theta,r,b} p(I|\theta,r,b)p(b). \tag{6}$$

Since we only need to estimate the scale b, traditional inference requires integration over r and θ, which is computationally costly. We pose the problem as joint inference, written as

$$\arg\max_{b} p(b|I) \approx \arg\max_{b}\{\max_{\theta,r} p(b,\theta,r|I)\}. \tag{7}$$

4.3 Solver

After taking negative logarithm, the problem to solve for b becomes

$$\sum_i \min_{\theta,r} m_i\rho(\|f(\Theta(\nabla I_i)) - \mathcal{T}(\theta,r,b)\|)) + \sum_{\{i,j\}\in W} w_{ij}|b_i - b_j|^2. \tag{8}$$

It is still nontrivial in optimization since this function involves non-convex penalties and non-local regularization. Discrete edgelet matching in the non-convex terms and information propagation from edges to all other pixels by non-local regularization can be achieved simultaneously due to their different natures in optimization.

Our scheme is to decouple these two different types of terms by introducing an auxiliary variable t. It changes the function to

$$\begin{aligned}\sum_i \min_{\theta,r} & m_i\rho(\|f(\Theta(\nabla I_i)) - \mathcal{T}(\theta,r,b)\|) + m_i\eta|t_i - b_i|^2 \\ & + \alpha \sum_{\{i,j\}\in W} w_{ij}|t_i - t_j|^2.\end{aligned} \tag{9}$$

This function approaches the original expression (8) when corresponding t_i and b_i are exactly the same. The term $\eta|t_i - b_i|^2$ is used to penalize the difference between t_i and b_i. η is a weight. Similar to the variable splitting strategy [Afonso et al. 2010], a very large η makes t_i and b_i move towards each other. We increase this weight in iterations to gradually tighten the connection between t_i and b_i. Eq. (9) enables an alternating optimization process to update the variables.

Scale b Computation When fixing t, Eq. (9) simplifies to

$$\min_b\{\sum_i \min_{\theta,r} m_i\{\rho(\|f(\Theta(\nabla I_i)) - \mathcal{T}(\theta,r,b)\|) + \eta|t_i - b_i|^2\}\}. \tag{10}$$

Regarding b, the global function is actually the sum of a few pixel-wise functions

$$\min_{\theta,r} m_i\{\rho(\|f(\Theta(\nabla I_i)) - \mathcal{T}(\theta,r,b)\|) + \eta|t_i - b_i|^2\}. \tag{11}$$

It involves two terms – one for patch difference to minimize the matching cost and the other is to reduce the distance between t and b in the pixel level. They are seemingly different in global optimization.

In fact, Eq. (11) can be understood differently based on the objective. We aim to estimate the values of b, corresponding to blur scales. As discussed in Section 3 for primitive set construction, due to the small scale of blurriness, we only consider 10 discrete values, which already cover general cases up to one pixel accuracy (more discussions in Section 6). For each possible value of b_i with index k, we find its nearest neighbor and compute the difference as

$$E(b_i,k) = \min_{\theta,r}\{\rho(\|f(\Theta(\nabla I_i)) - \mathcal{T}(\theta,r,b_i(k))\|), \tag{12}$$

Algorithm 1 Algorithm to estimate blur scales.

```
Initialization $t \leftarrow 0$
for l = 1 to maxiter do
    $b$-problem
    for each edge pixel $i$ do
        $\varepsilon \leftarrow 1E10$
        for each blur scale $b(k)$ do
            Compute $E(b,k)$ using KD-tree for the first time
            if $E(b,k) + \eta|t_i - b(k)|^2 \le \varepsilon$ then
                $k* \leftarrow k$, update $\varepsilon$
            end if
        end for
        $b_i \leftarrow b(k*)$
    end for
    $t$-problem
    Solve for $t$ in Eq. (15) using conjugate gradient descent
    $\eta \leftarrow 1.5\eta$
end for
```

where $b_i(k)$ denotes the kth value of b_i. In the matching process, we only need to vary θ and r to get the minimum matching difference. Then we store the temporary cost $E(b_i,k) + \eta|t_i - b_i(k)|^2$ for Eq. (11) given value $b_i(k)$.

When all the costs for b_i are obtained, we find the smallest one. It is guaranteed to be the minimum of Eq. (11) according to the computation procedure. We perform optimization for all pixels and eventually reach the global minimum for Eq. (10). Our nearest neighbor search is built upon a fast algorithm [Muja and Lowe 2009] with KD-tree acceleration.

We note that the robust function, which is in essence monotone, does not alter its minimum during the nearest neighbor (NN) search. We therefore use Euclidean distance in searching the NN without changing the minimum. So in our algorithm, although the NN search is performed in each iteration, the real computation is only in the first pass and its result is stored for further employment without any change. The overall inference is quite efficient.

Variable t Inference This step is to propagate sparse edge estimates to the whole image with non-local constraints

$$\sum_i m_i|t_i - b_i|^2 + \frac{\alpha}{\eta}\sum_{\{i,j\}\in W} w_{ij}|t_i - t_j|^2. \tag{13}$$

The function is quadratic; thus the closed-form solution exists. We note W involves many pixel pairs. Generation of the affinity matrix could be time and memory consuming, if not intractable. To address this issue, we write the energy function in the matrix form

$$(t_v - b_v)^T\mathbf{M}(t_v - b_v) + \frac{\alpha}{\eta}t_v^T\mathbf{W}t_v, \tag{14}$$

where matrix $\mathbf{M}$ encodes the indicator information and matrix $\mathbf{W}$ is the smoothing Laplacian. t_v and b_v are the variables in their vector form. Taking derivatives on t_v, we get

$$(\mathbf{M} + \frac{\alpha}{\eta}\mathbf{W})t_v = \mathbf{M}b_v. \tag{15}$$

Computing t_v involves matrix inversion of $\mathbf{M} + \frac{\alpha}{\eta}\mathbf{W}$, which is not necessarily sparse. We resort to a conjugate gradient (CG) method in order to avoid direct inversion. The computational expensive step in CG is to evaluate a matrix-vector product $(\mathbf{M} + \frac{\alpha}{\eta}\mathbf{W})q$ given a vector q. We employ an accelerated high-dimensional Gaussian filter for fast computation [Xu et al. 2013].

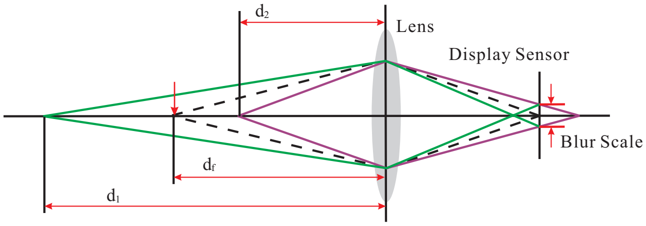
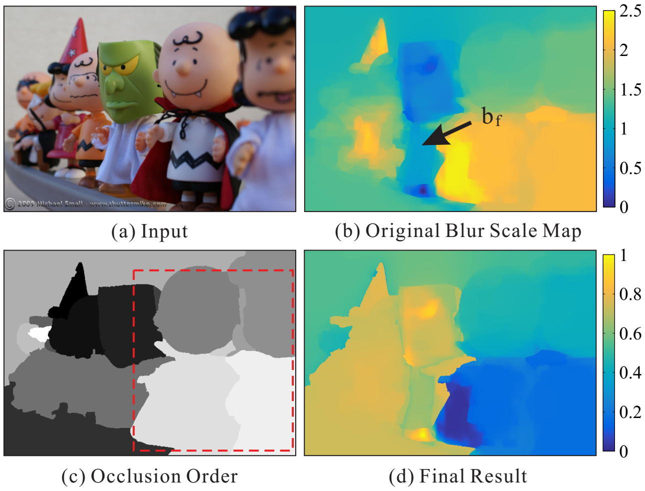


Figure 8: *A point causing small defocus blur could be in front of or behind the object plane that is perfectly focused.*



(a) Input (b) Original Blur Scale Map

(c) Occlusion Order (d) Final Result

Figure 9: *Blur scale to depth. (a) Input image. (b) Estimated defocus blur scale map b. (c) Coarse depth order map by high-level inference [Hoiem et al. 2007]. (d) Our final depth map.*

The overall algorithm is sketched in Alg. 1. Variable t and b are updated iteratively to finally approach each other. Intermediate results are shown in Fig. 4(e) and (f). They improve quickly in only a few iterations. Generally 5-8 iterations are enough for convergence.

5 Depth Order with Semantic Cues

Blur scale b can be mapped to scene depth z. The equation is

$$z = \frac{hA}{hA/f - A \pm b}, \tag{16}$$

where h is the distance between lens and sensor plane. A is the aperture diameter. f is the focal length.

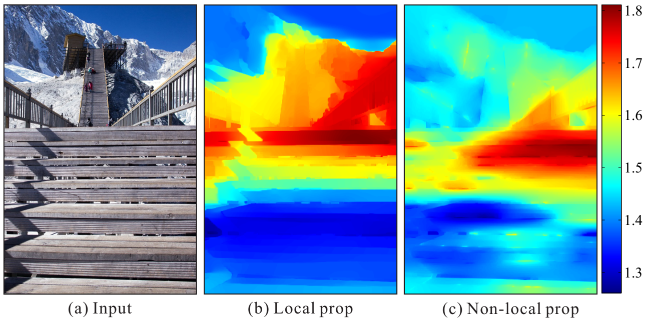
Depth Order Problem Definition Operator ± in Eq. (16) gives two possible depth choices in front of and behind exactly focused object plane. Its meaning is that *if one point is with a non-zero blur scale, it could be closer to or farther from the camera than points that are perfectly focused*, as illustrated in Fig. 8. This is a practical and important problem for depth inference from blur scales.

Depth Order Inference After algebraic operations, we get

$$b - b_f = \pm hA(1/z - 1/z_f) := \pm(d - d_f). \tag{17}$$

z_f is the distance, at which objects are perfectly in focus. We further denote $d = hA(1/z)$ and $d_f = hA(1/z_f)$ since we can combine camera parameters h, A and f as a single unknown.

To estimate the depth sign map, we initially employ a high-level semantic method [Hoiem et al. 2007] as guidance to infer very coarse depth orders among regions. This method employs a conditional random file (CRF) to compute foreground and background labels for neighboring regions when they form an occlusion boundary. One resulting map p is shown in Fig. 9(c). Obviously it does not present pixel-level information and labels are coarse only for several regions.



(a) Input (b) Local prop (c) Non-local prop

Figure 10: *Comparison of local and non-local propagation optimization results.*

We take estimate p as a coarse prior for our depth sign estimation. For each pixel i, we denote the binary sign variable as s_i, with its value selected from $\{-1, 1\}$. Based on Eq. (17), Our blur estimate b can be naturally linked to d as

$$d_i - d_f = s_i(b_i - b_f). \tag{18}$$

In practice, both d and s are not known in prior. We thus make use of the depth order map p to gather necessary information for their computation. With the p map, intriguingly we can express

$$s_i \cdot \text{sign}(b_i - b_f) = \text{sign}(d_i - d_f) \approx \text{sign}(p_i - p_f), \tag{19}$$

where operator sign$(\cdot)$ takes the sign. Variable s is now estimated by minimizing the Potts model, written as

$$\arg\min_{s_i} \sum_i |s_i \cdot \text{sign}(b_i - b_f) - \text{sign}(p_i - p_f)|^2 + \beta \sum_{\{i,j\} \in N} T(s_i \neq s_j), \tag{20}$$

where N is the set of neighboring pixels and β is the smoothing weight. T returns 1 when $s_i \neq s_j$ and 0 otherwise. So the entire objective function is to select a suitable order under the guidance of occlusion. The energy can be globally minimized using graph cuts [Rother et al. 2004].

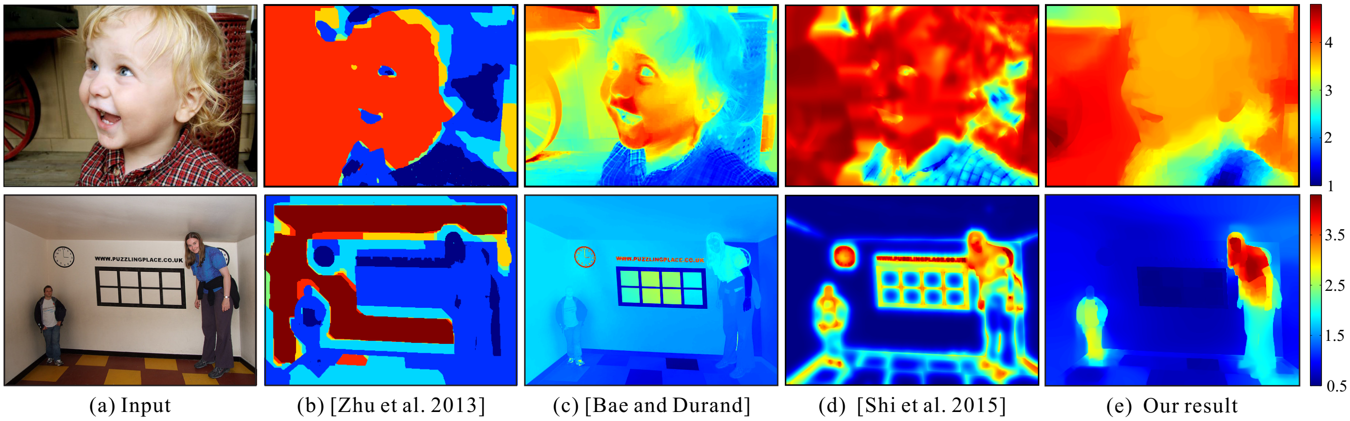
Our finally produced result with the depth order is shown in Fig. 9(d) where the front doll is made closer to the camera. After s_i is obtained, the normalized depth map can be recovered as

$$d_i = s_i(b_i - b_f) + d_f, \tag{21}$$

given a reference d_f. In fact, we can choose any d_f since it only affects where zero depth is set and does not change the relationship between foreground and background.

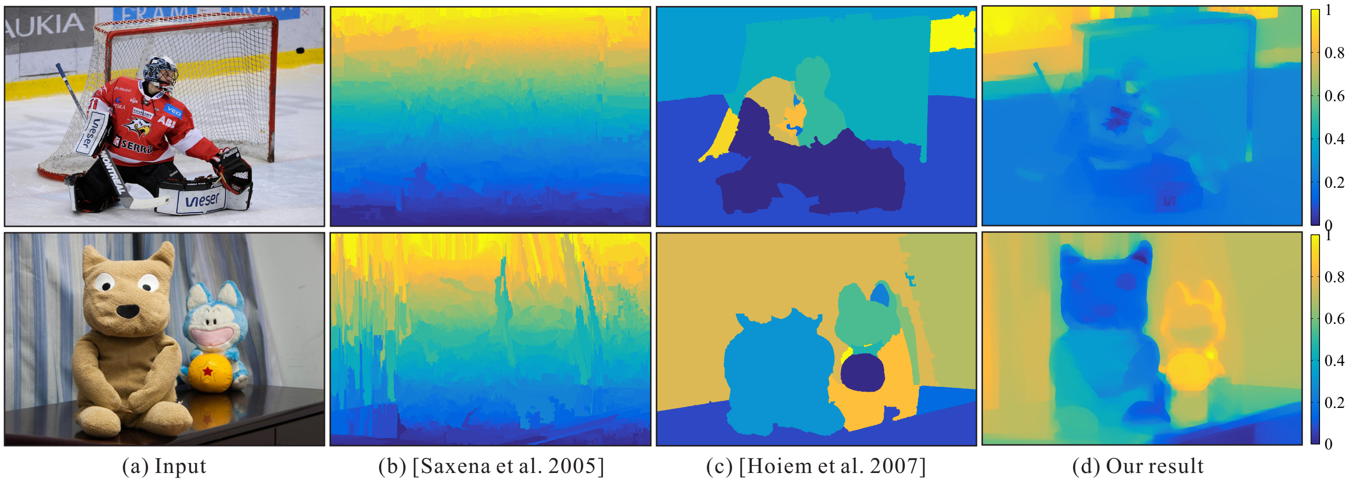
6 More Discussions

Edgelet Primitive Set Construction and Evaluation The latent edges are images sampled from the boundary of solid circles with different radius r in order to introduce orientation and curvature variation. Each patch is with size 13×13, sufficient to cover the small blurriness. The edgelet primitives are also produced with 10 different blur scales [Watanabe and Nayar 1998] with about 1 pixel interval. More scales are possible; but we found they are not necessary due to the limit of accuracy in the inference procedure. Subpixel accuracy after all may not be easily yielded in optimization.



(a) Input (b) [Zhu et al. 2013] (c) [Bae and Durand] (d) [Shi et al. 2015] (e) Our result

Figure 11: *Comparison with other single image defocus blur estimation methods. The Blue region is clear and the red pixels are blurred. The scale of the color bar indicates the blurriness.*



(a) Input (b) [Saxena et al. 2005] (c) [Hoiem et al. 2007] (d) Our result

Figure 12: *Comparison with data-driven image depth inference methods. The blur region is close to camera, whereas the yellow one is from afar. The scale of the color bar indicates the normalized depth.*

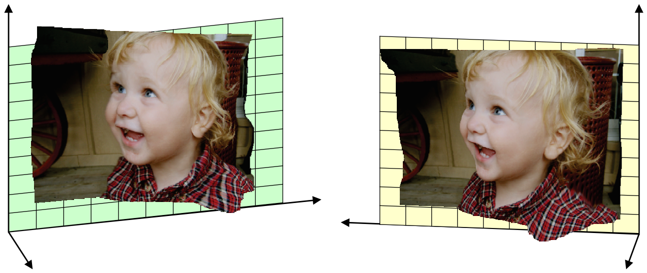


Figure 13: *3D view synthesis based on our single-image estimate.*

Local and Non-local Smoothness We compare results in Fig. 10 of our method and the modified one by substituting the local smoothness term $\prod \exp(-w_{ij}|b_i - b_j|^2)$ where i and j are neighboring for the non-local one in Eq. (5). The local smoothness form cannot propagate information similarly effective from sparse points as ours. The local smoothing result is thus with errors for pixels that are not close to any edge points.

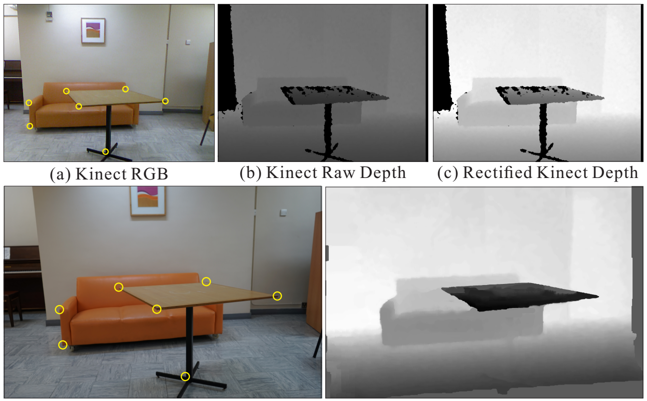
Edge Smoothness Ambiguity There always exists ambiguity when interpreting edge appearance. One patch in an image can be either a latent sharp edge diffused by the defocus blur or a smoother one affected by a smaller blurriness. There is no way to tell the difference from a single image. Our method also suffers from this ambiguity. But empirically natural images do not contain wildly changing edges, especially locally, which make our results mostly reasonable and usable in the applications described in Section 7.

Parameter Setting In the first iteration, we set η in Eq. (10) as zero to avoid cold start. Then in the subsequent t estimation procedure in Eq. (13), η is recovered as 10, and gets 1.5 times larger in each following iteration. α is set to 0.01. θ in Eq. (3) is set to 1 to make edge segments with error over 2 suppressed.

Computational Cost Our proposed solver has two major phases. The first is for edge-based defocus blur scale estimation. It needs to find the nearest neighbors for k edge pixels among n primitives. The KD-tree implementation can reduce the complexity to $O(k \log n)$. To process an 800×1000 image, it generally needs $10 \sim 15$ seconds on a single core of the i7 3.4GHz CPU. The result is stored and reused in subsequent iterations. The second phase is the global propagation by solving a large linear system. By conjugate gradient, it takes about 3 seconds for each iteration and 20 seconds for the whole process.

7 Experiments

We first compare our method with others to estimate blur or depth from single images or learning, and then show a few applications.


(a) Kinect RGB (b) Kinect Raw Depth (c) Rectified Kinect Depth

(d) Sony A7 RGB (e) Sony A7 depth after post-processing

Figure 14: *Depth from Kinect. (a) and (b) are the raw output from Kinect. (c) is the rectified depth aligned with the RGB image. (d) is another RGB image from the Sony A7 camera. We align (a) and (d) with the labeled matching points, and accordingly warp (c) to obtain (e). The final image and depth pair are (d) and (e).*

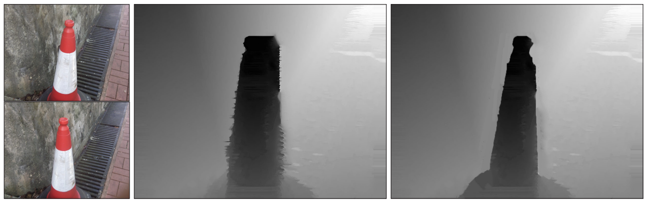

(a) Input (b) Initial depth (c) Final depth

Figure 15: *Ground-truth depth from Stereopsis. (a) contains the left and right images from the 3D camera. (b) is the initial stereo matching result. (c) is the final depth after user correction.*

7.1 Comparison with Defocus Estimation

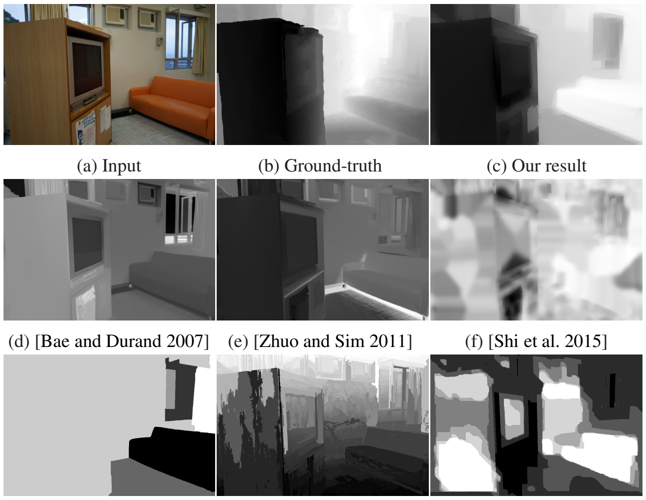
Given the examples in Fig. 11, we compare our results with those of defocus blur estimation methods [Bae and Durand 2007; Zhu et al. 2013], which have been discussed in Section 3. We also add comparison with recent small blur perception method [Shi et al. 2015]. For the first image, the boy's left shoulder is in perfect focus. The face is in depth of field, varying from left to right. Our result in Fig. 11 contains this type of varying-depth information while the other three do not. After novel view synthesis, our result is shown in Fig. 13. The second example in Fig. 11 is the Ames room image. We also show clear variation between the two persons.

Methods	Rel	Log10	RMSE	RelOrder
[Bae and Durand 2007]	1.2880	0.3600	0.3528	0.0921
[Zhuo and Sim 2011]	1.2597	0.3178	0.3210	0.1506
[Zhu et al. 2013]	1.0775	0.5793	0.4383	0.0386
[Shi et al. 2015]	1.6756	0.2866	0.3055	0.1854
[Saxena et al. 2005]	1.0027	0.3170	0.3006	0.1288
[Hoiem et al. 2007]	1.4317	0.4621	0.4082	0.0160
Ours	**0.8997**	**0.2637**	**0.2681**	**0.1954**

Table 1: *Quantitative evaluation on our data.*

7.2 Comparison with Data-driven Depth Inference

We show results of data-driven depth inference [Saxena et al. 2005] in Fig. 12(b). This approach has the vital need of proper data. When there is no sufficient similar image examples for training, it could fail. The results of region-based order cues [Hoiem et al. 2007] in (c) are also coarser than ours, as this method relies on correctly detected occlusion among layers. Relative depth difference may not be well captured.


(a) Input (b) Ground-truth (c) Our result

(d) [Bae and Durand 2007] (e) [Zhuo and Sim 2011] (f) [Shi et al. 2015]

(g) [Hoiem et al. 2007] (h) [Saxena et al. 2005] (i) [Zhu et al. 2013]

Figure 16: *Visual comparison on our benchmark data.*

7.3 Quantitative Evaluation

We note it is difficult to quantitatively measure the depth estimate because accurate depth information for a high-quality photo captured by phone and professional cameras is generally not available. We put effort designing the following experiments.

Kinect Data We first use Kinect to obtain a few depth images as ground-truth only for indoor scenes due to the infrared camera mechanism. Because the accompanying RGB images are with blurred edges, low dynamic ranges and strong noise, they do not present the required small-blur effect and thus are not usable in our method. We have to take another high-quality RGB image by a Sony A7 camera corresponding to each depth map.

When taking these RGB images, we put the Sony camera the same position as the Kinect and let the images cover the regions in the depth maps. Because there is inevitable displacement of the two cameras and difference on parameters such as focal length, ISO, resolution, and white balance, after taking these images, we manually align depth from Kinect and the RGB images. One example is shown in Fig. 14. We first align the raw depth-RGB pair captured from Kinect in (a) and (b) using the official rectification API to obtain the rectified depth (c). It is followed by a depth warping process from (c) to (e), with the warping field constructed from manually-labeled correspondences in (a) and (d) from the two cameras. Holes and noise are further touched up in postprocessing.

We note error-free pixel-wise alignment is impossible to produce even with labor-intensive manipulation. But these data are sufficiently usable to understand how reasonable a depth estimate is. In total, we produce 15 image-depth pairs using this strategy.

Stereopsis Data Another set of data are captured by a *Fujifilm Real 3D* camera mostly for outdoor scenes. Their ground-truth depth is obtained from stereo matching [Rhemann et al. 2011]. We involve manual interaction as well to correct obvious mistakes and misaligned boundaries. One example is shown in Fig. 15 where (a) contain the input stereo images. The initial stereo matching result

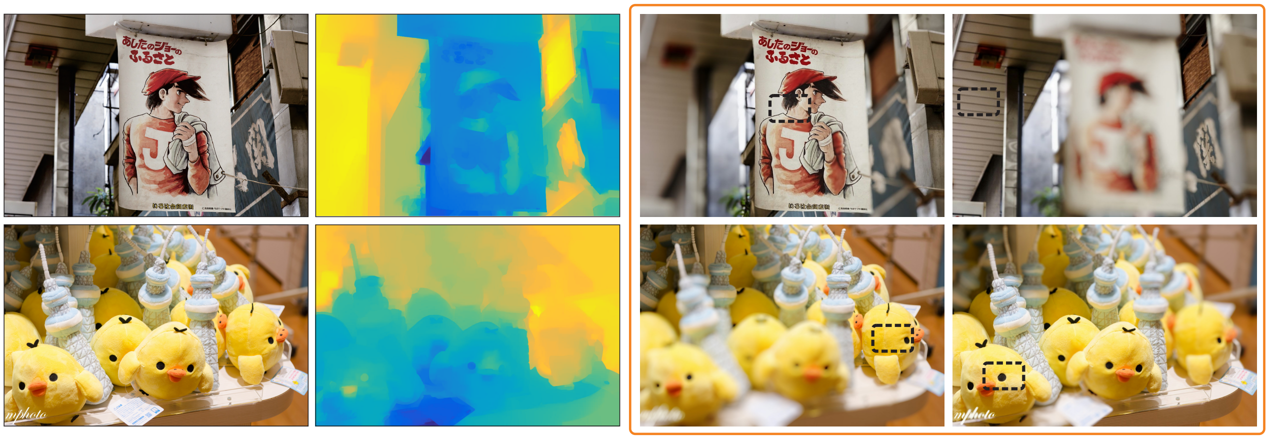

(a) Input (b) Blur Scale Map (c) Refocus Result

Figure 17: *Image refocus using our estimated blur scale maps. The black boxes highlight our focusing points.*

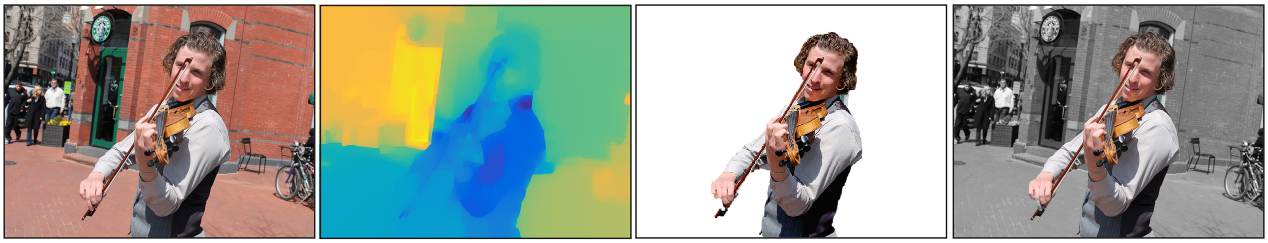

(a) Input (b) Blur Scale Map (c) Foreground (d) Background Color Removal

Figure 18: *Automatic foreground-background segmentation and background decolorization using our estimated blur map.*

is shown in (b). Our final depth map after user touchup is shown in (c) – many small errors are fixed. A total of 15 image pairs are produced via this process.

Evaluation Based on this data, we quantitatively evaluate methods of [Bae and Durand 2007; Zhuo and Sim 2011; Zhu et al. 2013; Shi et al. 2015; Saxena et al. 2005; Hoiem et al. 2007] and report the results in Table 1. Our images include indoor & outdoor scenes in different lightings, and contain objects with sharp & soft edges.

Table 1 manifests that our method performs reasonably under all evaluation metrics including relative error (Rel), Log10 error, root mean square error (RMSE), and relative depth order (RelOrder). These metrics are explained in what follows.

For an image with N pixels, its i-th pixel has depth estimate d_i compared with the ground-truth d^*. The relative error is defined as $\frac{1}{N}\sum_i (d_i - d_i^*)/d$. The Log10 error is set as $\sum_i |\log_{10}(d_i) - \log_{10}(d_i^*)|/N$ and the root mean square error is denoted as $\sqrt{\sum_i (d_i - d_i^*)^2/N}$ for images with N pixels. The relative depth order is to randomly sample 15,000 points in each image and combine them with their 8 nonlocal neighbors to form 120,000 2-tuples. Then the percentage of these tuples whose relative orders are consistent with the ground truth is counted.

A visual comparison with defocus and data-driven based methods on our data is shown in Fig. 16. For all images, the focal point is set at the closest object to avoid ambiguity for these previous methods. Our result in (c) contains acceptable depth layers.

7.4 Single Image Digital Refocus

Adding refocus effect to a single natural image is difficult without additional depth information. This task can be quickly accomplished with our general small-scale blur maps. In the refocus process, the blur kernel size is proportional to $|d - d_f|$. By selecting the focusing point manually, we actually define d_f. Then the blur effect applies to different pixels in a spatially varying manner.

Two examples with their corresponding blur scale maps are shown in Fig. 17. We refocus the image and make defocus strong as shown in (c). Our defocus blur estimation method thus enables promising single image shoot-and-refocus functionality from just a common still-image camera. The poster-boy example shows slight blur-map errors do not affect refocusing much.

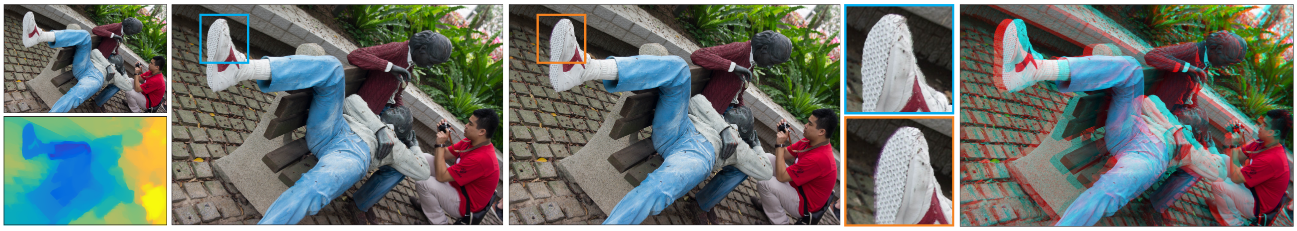
7.5 Foreground Extraction

A foreground layer in a natural image mostly corresponds to the region of interest. Its estimation has long been an important problem because it tells what people care in images. Our defocus blur estimates naturally serve this purpose because we can simply set the nearest region as foreground. For better segmentation incorporating color information, we apply grab-cut [Rother et al. 2004] where color and depth are both included as segmentation features.

One result is shown in Fig. 18(c). Based on it, we can generate more effects, such as fading background color to highlight the foreground person, as shown in (d).

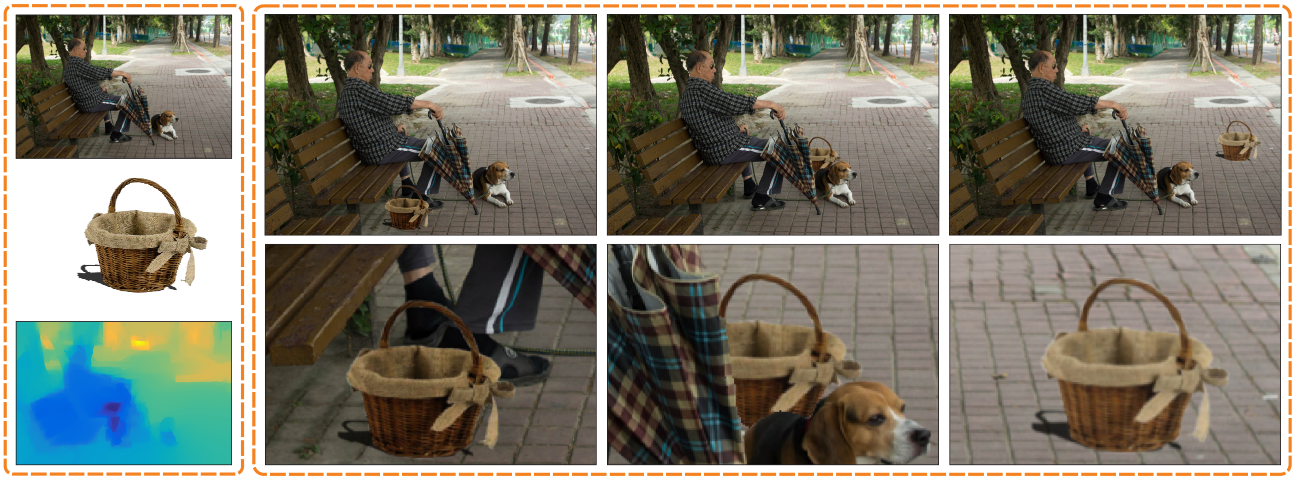
7.6 Anaglyph 3D generation

We also generate stereo pairs by rendering a new view based on our depth estimates. Disparity used in stereopsis is set as the reciprocal of our estimate d. Following traditional stereopsis synthesis, we generate anaglyph 3D images. Fig. 19 shows an example. Given the small-scale defocus blur map (b), a stereo pair in (c) and (d) are



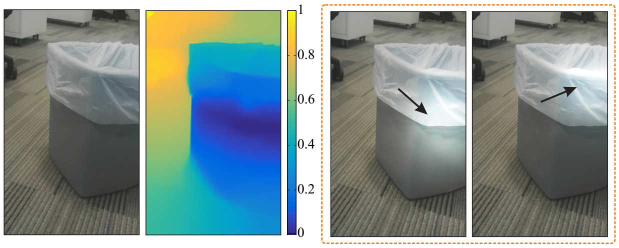
(a) Input & Blur Scale (b) Left Image (c) Right Image (d) Close-up (e) Anaglyph 3D Image

Figure 19: *Stereo image generation using our estimated defocus blur map.*



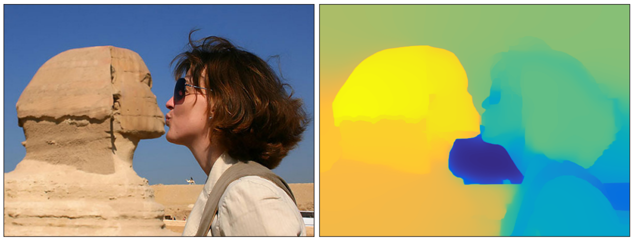
(a) Input & Blur Scale (b) Output Image and Closeup

Figure 20: *Blur-aware image composition given our defocus blur estimates and depth maps.*



(a) Input (b) Blur Scale Map (c) Manipulate Lighting

Figure 21: *Lighting manipulation using our estimated small-scale defocus map.*



(a) Input (b) Blur Scale Map

Figure 22: *One failure case.*

generated. The closeup is shown in (e). An anaglyph 3D image is contained in (f), which can be viewed via 3D red-cyan glasses.

7.7 Blur-aware Image Composition

Inserting an object to images [Jia et al. 2006] usually needs human interaction to deal with occlusion. Our estimates simplify this process. Fig. 20 shows an example for such composition. Given the basket and target scene images captured by the same camera, we estimate their depth and associated blur scales. Then we place the basket in three different positions in the target image. Occlusion is automatically generated when the basket is behind the objects in the scene. Depth-of-field is also generated in the composition results.

7.8 Manipulate Lighting

Our depth map is usable for re-lighting. We compute the surface normal from RGB image and depth map via the method of [Chen and Koltun 2013], and then use it for relighting. Fig. 21 shows an example that we add a point light source to lighten the bin. We do not involve or vary other physical properties, such as reflection and surface material, in this example for simplicity's sake.

8 Limitations and Conclusion

Our method infers small-scale defocus blur. So it works best for photos in their original resolution from cameras. If an image is largely resampled, the blur information could be weakened or eliminated. While our method can distinguish among small blur scales, it only indicates depth orders, rather than accurate depth values. Besides, our method obtains the blur scales on edges in the first step. If one region is occluded and the only visible part does not contain any of its usable edges, there is no information to let our system know it is a background region and its estimate could be similar to the nearest occluder. One example is the central sky region in Fig. 22. In specific cases, textures and edges in pictures inside photos may mislead blur estimation. As shown in Fig. 17, the drawing on the poster harms the blur scale map, in which case semantic information is needed. Moreover, additional motion blur will harm our estimated blurriness. Finally, as discussed in Section 6, it is not possible to resolve the edge smoothness ambiguity. Local errors could be caused when latent edge smoothness changes quickly.

To conclude this paper, we have proposed a non-parametric framework to estimate small-scale blur that exists in photos. Our experiments show this type of structure is surprisingly useful for inferring geometric information from even a single image. We introduced an optimization framework to make the problem trackable. A few geometry or depth related applications are enabled on single images.

Acknowledgements

We thank flickr users "Ben Millett" (benmillett), "Alan Travers" (alan-travels), "Mike Small" (shuttermike) and Blogspot user "mphoto888" (erasmushsu@gmail) for the pictures used in the paper. This work is supported by a grant from the Research Grants Council of the Hong Kong SAR (project No. 412911).

References

AFONSO, M. V., BIOUCAS-DIAS, J. M., AND FIGUEIREDO, M. A. 2010. Fast image recovery using variable splitting and constrained optimization. *TIP 19*, 9, 2345–2356.

BAE, S., AND DURAND, F. 2007. Defocus magnification. *Computer Graphics Forum 26*, 3, 571–579.

CHAKRABARTI, A., ZICKLER, T., AND FREEMAN, W. T. 2010. Analyzing spatially-varying blur. In *CVPR*, 2512–2519.

CHEN, Q., AND KOLTUN, V. 2013. A simple model for intrinsic image decomposition with depth cues. In *ICCV*, 241–248.

COSSAIRT, O., ZHOU, C., AND NAYAR, S. 2010. Diffusion coded photography for extended depth of field. *TOG 29*, 4, 31.

EIGEN, D., PUHRSCH, C., AND FERGUS, R. 2014. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, 2366–2374.

ELDER, J. H., AND ZUCKER, S. W. 1998. Local scale control for edge detection and blur estimation. *TPAMI 20*, 7, 699–716.

HOIEM, D., STEIN, A. N., EFROS, A. A., AND HEBERT, M. 2007. Recovering occlusion boundaries from a single image. In *ICCV*, 1–8.

JIA, J., SUN, J., TANG, C.-K., AND SHUM, H.-Y. 2006. Drag-and-drop pasting. *TOG 25*, 3, 631–637.

KARSCH, K., LIU, C., AND KANG, S. B. 2012. Depth extraction from video using non-parametric sampling. In *ECCV*, 775–788.

KHOSHELHAM, K., AND ELBERINK, S. O. 2012. Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors 12*, 2, 1437–1454.

LADICKY, L., SHI, J., AND POLLEFEYS, M. 2014. Pulling things out of perspective. In *CVPR*, 89–96.

LEVIN, A., FERGUS, R., DURAND, F., AND FREEMAN, W. T. 2007. Image and depth from a conventional camera with a coded aperture. *TOG 26*, 3, 70.

LIANG, C.-K., LIN, T.-H., WONG, B.-Y., LIU, C., AND CHEN, H. H. 2008. Programmable aperture photography: Multiplexed light field acquisition. *TOG 27*, 3, 55.

MAINI, R., AND SOHAL, J. 2006. Performance evaluation of prewitt edge detector for noisy images. *GVIP Journal 6*, 3, 39–46.

MUJA, M., AND LOWE, D. G. 2009. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Application*, 331–340.

RHEMANN, C., HOSNI, A., BLEYER, M., ROTHER, C., AND GELAUTZ, M. 2011. Fast cost-volume filtering for visual correspondence and beyond. In *CVPR*, IEEE, 3017–3024.

ROTHER, C., KOLMOGOROV, V., AND BLAKE, A. 2004. Grabcut: Interactive foreground extraction using iterated graph cuts. *TOG 23*, 3, 309–314.

SAXENA, A., CHUNG, S. H., AND NG, A. Y. 2005. Learning depth from single monocular images. In *NIPS*, 1–8.

SAXENA, A., SUN, M., AND NG, A. 2009. Make3d: Learning 3d scene structure from a single still image. *TPAMI 31*, 5, 824–840.

SCHARSTEIN, D., AND SZELISKI, R. 2002. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV 47*, 1-3, 7–42.

SCHECHNER, Y. Y., AND KIRYATI, N. 2000. Depth from defocus vs. stereo: How different really are they? *IJCV 39*, 2, 141–162.

SHI, J., XU, L., AND JIA, J. 2015. Just noticeable defocus blur detection and estimation. In *CVPR*, 1–8.

SU, H., HUANG, Q., MITRA, N. J., LI, Y., AND GUIBAS, L. 2014. Estimating image depth using shape collections. *TOG 33*, 4, 37.

SUBBARAO, M., AND SURYA, G. 1994. Depth from defocus: a spatial domain approach. *IJCV 13*, 3, 271–294.

TAI, Y.-W., AND BROWN, M. S. 2009. Single image defocus map estimation using local contrast prior. In *ICIP*, 1797–1800.

VEERARAGHAVAN, A., RASKAR, R., AGRAWAL, A., MOHAN, A., AND TUMBLIN, J. 2007. Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing. *TOG 26*, 3, 69.

WATANABE, M., AND NAYAR, S. K. 1998. Rational filters for passive depth from defocus. *IJCV 27*, 3, 203–225.

WU, T.-P., SUN, J., TANG, C.-K., AND SHUM, H.-Y. 2008. Interactive normal reconstruction from a single image. *TOG 27*, 5, 119.

XU, L., YAN, Q., AND JIA, J. 2013. A sparse control model for image and video editing. *TOG 32*, 6, 197.

ZHOU, C., AND NAYAR, S. 2009. What are good apertures for defocus deblurring? In *ICCP*, 1–8.

ZHOU, C., LIN, S., AND NAYAR, S. 2009. Coded aperture pairs for depth from defocus. In *ICCV*, 325–332.

ZHU, X., COHEN, S., SCHILLER, S., AND MILANFAR, P. 2013. Estimating spatially varying defocus blur from a single image. *TIP 22*, 12, 4879–4891.

ZHUO, S., AND SIM, T. 2011. Defocus map estimation from a single image. *Pattern Recognition 44*, 9, 1852–1858.

ZIOU, D., AND DESCHÊNES, F. 2001. Depth from defocus estimation in spatial domain. *CVIU 81*, 2, 143–165.