

Towards Digital Refocusing from a Single Photograph

Yosuke Bando^{†,‡}

[‡]TOSHIBA Corporation
yosuke1.bando@toshiba.co.jp

Tomoyuki Nishita[†]

[†]The University of Tokyo
{ybando, nis}@is.s.u-tokyo.ac.jp



Figure 1. Left: A single input photograph, focused on the person in the left. Middle: Created image, refocused on the person in the middle. Right: Created image, refocused on the person in the right.

Abstract

This paper explores an image processing method for synthesizing refocused images from a single input photograph containing some defocus blur. First, we restore a sharp image by estimating and removing spatially-variant defocus blur in an input photograph. To do this, we propose a local blur estimation method able to handle abrupt blur changes at depth discontinuities in a scene, and we also present an efficient blur removal method that significantly speeds up the existing deconvolution algorithm. Once a sharp image is restored, refocused images can be interactively created by adding different defocus blur to it based on the estimated blur, so that users can intuitively change focus and depth-of-field of the input photograph. Although information available from a single photograph is highly insufficient for fully correct refocusing, the results show that visually plausible refocused images can be obtained.

1. Introduction

Digital refocusing, a technique that generates photographs focused to different depths (distances from a camera) after a single camera shot, is attracting the attention of the computer graphics community and others in view of its interesting and useful effects. The technique is based on the light field rendering, and exploits the fact that a photograph is a 2D integral projection of a 4D light field [27], as was simulated by Isaksen et al. [17]. Ng et al. made this technique practical with their hand-held plenoptic camera [28],

eliminating the need for large and often expensive apparatus such as a camera array or a moving camera that was traditionally required to capture light fields. Since then, other novel camera designs have been emerging in order to improve the resolution of images and/or to reduce the cost of optical equipment attached to a camera [16].

In an attempt to make digital refocusing a more common tool for digital photography, we are interested in developing an image processing method for synthesizing refocused images from a single photograph taken with a conventional camera. If we had a sharp, all-focus photograph with a depth map of the scene, it would be straightforward to create depth-of-field effects by blurring the input photograph according to the depth, as some of the existing image-editing software do (e.g., the *Lens Blur* filter of Adobe Photoshop CS). In this paper we address a more general and challenging problem where an input photograph is focused to a certain depth with a finite depth-of-field and hence contains some defocus blur as shown in Fig. 1 left, and where we must first estimate “a sharp image with a depth map” from that photograph.

In general, this problem ultimately amounts to reconstructing a 4D light field from a single image, which is intractable. Therefore, in this paper we assume that spatially-variant defocus blur in an input photograph can be locally approximated by a uniform blur, and we restore a sharp image by stitching multiple deconvolved versions of an input photograph. We present a deconvolution algorithm that significantly speeds up one of the state-of-the-art methods

called *WaveGSM* [6]. And we also propose a local blur estimation method applicable to irregularly-shaped image segments in order to handle abrupt blur changes at depth discontinuities due to object boundaries. To create desired refocusing effects, we present several means of determining the amount of blur to be added to a restored sharp image based on the estimated blur, by which users can change focus and depth-of-field interactively and intuitively.

Even with the assumption described above, the problem is still ill-posed, and created images can have artifacts that might need to be retouched. We provide users with a means of modifying an estimated blur field to partially fix them.

2. Related Work

To our knowledge, techniques that synthesize refocused images from a single conventional photograph have not been reported in the literature. Kubota and Aizawa used two images, and generated arbitrarily focused images by assuming that a scene consisted of two depth layers, each of which was in focus in either image [19]. Fusion-based image enhancement methods can generate images with an extended depth-of-field from multiple input images [8].

Blind image deconvolution techniques restore the original sharp image from an observed degraded image without precise knowledge of a *point-spread function* (PSF) [20]. Most of the existing methods assume a PSF to be either spatially-invariant (uniform) [14, 10] or spatially-variant but slowly varying [29, 21] across the image, and are not directly applicable to removing defocus blurs in photographs where captured scenes have wide depth variations. Levin identified spatially-variant motion blur by examining the difference between the image derivative distribution along the motion direction and that along its perpendicular direction [23]. This technique cannot be applied to our problem, as defocus blur has no directionality.

Depth-from-focus/defocus techniques generate a depth map of a scene by estimating the amount of defocus blurs in images. Existing methods either use multiple images [30, 38, 26], or make an estimate at edges in a single image by assuming that a blurred ramp edge is originally a sharp step edge [30, 35, 22].

Spatially-invariant, non-blind image deconvolution still remains an active research area [5, 15]. We build our deconvolution algorithm on one of the state-of-the-art methods called *WaveGSM* [6] as an integral component of our digital refocusing method.

Other related work exploiting customized optical elements includes wavefront coding [13] for making an imaging system insensitive to misfocus; defocus video matting [25] with multiple video cameras with different focus settings; and motion deblurring with a fluttered shutter [32].

3. Proposed Method

3.1. Overview

Fig. 2 shows a block diagram of our method. From an input photograph $g_c(x, y)$ with $c \in \{\text{red, green, blue}\}$, we first restore a *latent image* $l_c(x, y)$, which would have been observed if defocus blur had not been introduced by the camera lens system. We use the standard disc PSF parameterized by radius r of the circle of confusion, referred to as *blur radius*, as a defocus blur model [4]:

$$h(x, y; r) = \begin{cases} 1/\pi r^2 & \text{for } \sqrt{x^2 + y^2} \leq r \\ 0 & \text{otherwise} \end{cases}, \quad (1)$$

and we generate multiple differently deblurred images $d_{c,j}(x, y)$ by deconvolving an input photograph with each of the predetermined $M + 1$ blur radii $\{r_j | j = 0, 1, \dots, M\}$. That is, we remove uniform defocus blur with blur radius r_j from $g_c(x, y)$ to obtain $d_{c,j}(x, y)$. This amounts to solving

$$g_c(x, y) = h(x, y; r_j) * d_{c,j}(x, y) + n_c(x, y), \quad (2)$$

where $*$ denotes convolution, and $n_c(x, y)$ is a noise term. Eqn. 2 is known to be an ill-posed inverse problem, whose solution is given in Sec. 3.2. The $M + 1$ blur radii are arranged in ascending order as $r_0 < r_1 < r_2 < \dots < r_M$, and $r_0 = 0$ so that $d_{c,0}(x, y) \equiv g_c(x, y)$. We typically use $r_j = 0.5j$, and $r_M = 10.0$ (in pixels).

From the deblurred images $d_{c,j}(x, y)$, we locally select the “best” image and stitch them together to obtain the latent image $l_c(x, y)$, the approach known as *sectional method* [37]. More precisely, we first estimate a blur radius field $r_{org}(x, y)$ which describes with what blur radius the input photograph is originally blurred around each pixel location (x, y) , as described in Sec. 3.3. We then linearly blend the deblurred images as

$$l_c(x, y) = \frac{r_{j+1} - r_{org}(x, y)}{r_{j+1} - r_j} d_{c,j}(x, y) + \frac{r_{org}(x, y) - r_j}{r_{j+1} - r_j} d_{c,j+1}(x, y), \quad (3)$$

where j is appropriately chosen for each pixel (x, y) such that $r_j \leq r_{org}(x, y) \leq r_{j+1}$.

Now that we obtained the latent image $l_c(x, y)$, we create an output refocused image $o_c(x, y)$ by blurring the latent image¹. Sec. 3.4 presents a method for determining a new blur radius field $r_{new}(x, y)$ to be added to the latent image in order to meet desired refocusing effects. To perform the synthesis in real-time, we again employ the sectional method, and we prepare multiple differently blurred images as $b_{c,j}(x, y) = h(x, y; r_j) * l_c(x, y)$ in the preprocessing stage. In the interactive refocusing stage, we perform linear interpolation similar to Eqn. 3 for a new blur radius field $r_{new}(x, y)$ and the blurred images $b_{c,j}(x, y)$ and $b_{c,j+1}(x, y)$.

¹Because convolution of two disc PSFs does not result in another disc PSF, refocused images cannot be obtained by directly convolving/deconvolving an input photograph.

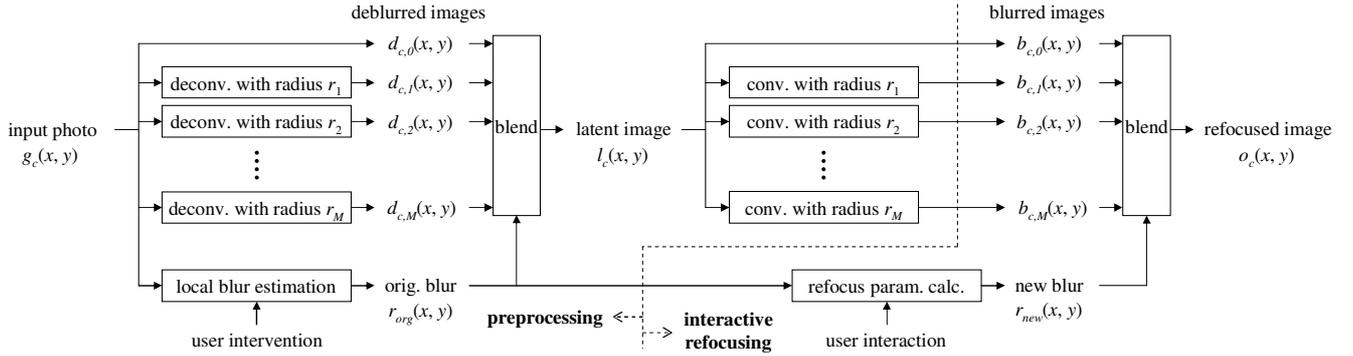


Figure 2. Block diagram of the proposed method.

3.2. Image Deconvolution

For notational convenience, this subsection uses a matrix-vector version of Eqn. 2 with subscripts omitted as [4]

$$\mathbf{g} = \mathbf{H}\mathbf{d} + \mathbf{n}, \quad (4)$$

where \mathbf{g} , \mathbf{d} , and \mathbf{n} are P -vectors representing $g_c(x, y)$, $d_{c,j}(x, y)$, and $n_c(x, y)$, respectively, with lexicographic ordering of P discretized pixel locations, and \mathbf{H} is a $P \times P$ matrix representing convolution by a PSF $h(x, y; r_j)$.

Since solving Eqn. 4 for \mathbf{d} as a least squares problem of minimizing $\|\mathbf{g} - \mathbf{H}\mathbf{d}\|^2$ is known to be ill-posed due to ill-conditioned matrix \mathbf{H} , one needs prior knowledge about which images are more likely to occur in nature. However, frequently-used Gaussian smoothness priors are not suitable for restoring sharp (hence not necessarily smooth) images. Therefore, recent methods exploit so-called *heavy-tailed priors*, according to which the distributions of band-pass filter outputs of (sharp) natural images have a narrower peak and a broader foot than Gaussians as shown in Fig. 3, allowing occasional discontinuities (such as edges) in restored images [15, 6]. These methods use *discrete wavelet transform* (DWT) as band-pass filters, but since restored images suffer from blocky artifacts arising from the dyadic image partitioning in DWT, they use *translation-invariant DWT* (TI-DWT) [11] to reduce such artifacts at the cost of significant increase in computational complexity.

We avoid this problem by using derivative filters instead of DWT, as they are translation-invariant and do not perform dyadic image partitioning. Specifically, we bring Bioucas-Dias’s wavelet domain method (WaveGSM) [6] into the gradient domain, because the *Gaussian scale mixture* (GSM) representation used in WaveGSM is also applicable to speeding up the non-linear optimization involving heavy-tailed priors in the gradient domain. Applying heavy-tailed priors to gradient distributions was shown to be useful in super-resolution [36] and camera shake PSF estimation [14]. Following Tappen et al. [36], we use a generalized

Laplacian distribution as a heavy-tailed prior model:

$$p(\mathbf{d}_x[i]) \propto \exp(-|\mathbf{d}_x[i]|^\alpha / \beta), \quad (5)$$

where $\mathbf{d}_x[i]$ denotes the i -th element of the derivative of \mathbf{d} with respect to x , and $p(\cdot)$ denotes a probability density function of an argument variable. We set $\alpha = 0.3$ and $\beta = 0.085$ with pixel values in range $[0, 1]$, so that Eqn. 5 approximates sample gradient distributions as shown in Fig. 3. We use the same prior for y derivatives, $\mathbf{d}_y[i]$.

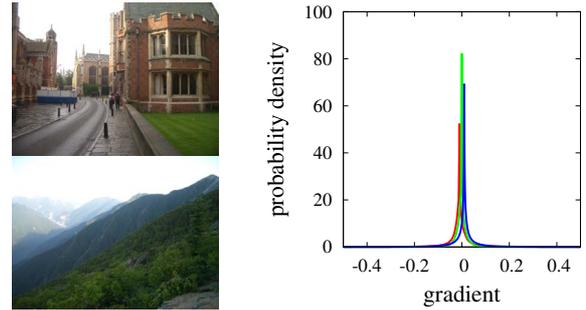


Figure 3. Left: Sample sharp images. Right: Gradient distributions of the top image (red) and of the bottom image (green), and the generalized Laplacian distribution we use (blue). For visibility, these plots are horizontally displaced. They all actually peak at zero.

Taking derivatives of Eqn. 4 leads to the following two gradient domain deconvolution equations:

$$\mathbf{g}_x = \mathbf{H}\mathbf{d}_x + \mathbf{n}_x, \quad \mathbf{g}_y = \mathbf{H}\mathbf{d}_y + \mathbf{n}_y. \quad (6)$$

Through the derivation described in Appendix A, the x part of Eqn. 6 leads to *expectation maximization* (EM) iterations involving the following system of linear equations:

$$(\mathbf{H}^T \mathbf{H} + w \mathbf{S}^m) \mathbf{d}_x = \mathbf{H}^T \mathbf{g}_x, \quad (7)$$

where \mathbf{H}^T is the transpose of \mathbf{H} , m is an EM iteration count, \mathbf{S}^m is a diagonal matrix representing the prior term that is

updated for each EM iteration, and w is a user-specified weight for the prior term (typically around 10^{-3}). The solution to Eqn. 7 for \mathbf{d}_x becomes the next estimate \mathbf{d}_x^{m+1} , from which \mathbf{S}^{m+1} is computed, and this process is iterated. Eqn. 7 can be solved rapidly by the *second-order stationary iterative method* [3], with the use of *fast Fourier transform* (FFT) for matrix multiplication by \mathbf{H} and \mathbf{H}^T . We set the observation as an initial estimate: $\mathbf{d}_x^0 = \mathbf{g}_x$. The y part of Eqn. 6 is solved similarly. After obtaining estimated latent gradients $\hat{\mathbf{d}}_x$ and $\hat{\mathbf{d}}_y$, we reconstruct the deblurred image $\hat{\mathbf{d}}$ by solving a Poisson equation [31]. As we use FFT, periodic boundary conditions are assumed. Edge tapering is performed to reduce boundary effects, and the DC component lost by the derivative filters is restored from the input photograph.

The time complexity of our method is $O(P \log P)$ in the number P of pixels owing to the use of FFT, which remains the same as that of WaveGSM. However, the amount of computation is significantly reduced in two respects. First, we have only $O(P)$ derivative coefficients to be updated, in contrast to $O(P \log P)$ TI-DWT coefficients. Second, WaveGSM performs $O(P \log P)$ TI-DWT and its inverse for each iteration, whereas our method performs derivative and its inverse (i.e., integral) operations only at the beginning (by deriving Eqn. 6 from Eqn. 4) and at the end (by solving a Poisson equation) of the whole deconvolution process.

3.3. Local Blur Estimation

Similar to the existing spatially-variant PSF estimation techniques, we divide an image into segments, and we assume the blur to be uniform within each segment. However, rectangular segmentation as in [29, 21] can produce segments that violate this uniformity assumption, as the blur radius can change abruptly due to depth discontinuities at object boundaries. Therefore, we perform color image segmentation [12] so that segments conform to the scene content. In what follows, we present a blur radius estimation method that is applicable to non-rectangular segments.

Our approach is to select the blur radius from the pre-determined $M + 1$ candidate blur radii $\{r_j\}$ that gives the “best” deblurred image for each segment. Unfortunately, *focus measures* [34, 18] are not suitable for this selection criterion, because digitally deconvolved images with wrong blur radii have different image statistics from optically mis-focused images. Instead, we measure the amplitude of oscillatory artifacts in deblurred images due to overcompensation of blur (examples can be seen in Fig. 7). For simplicity, we explain this phenomenon using the 1D version of Eqn. 2:

$$g(x) = h(x; r) * d(x) + n(x), \quad (8)$$

where the PSF is given by the following box function:

$$h(x; r) = \begin{cases} 1/2r & \text{for } |x| \leq r \\ 0 & \text{otherwise} \end{cases}. \quad (9)$$

In the frequency domain, Eqn. 8 is rewritten as

$$G(\omega) = \text{sinc}(r\omega)D(\omega) + N(\omega), \quad (10)$$

where uppercase letters represent the Fourier transforms of their lowercase counterparts, and ω denotes frequency. The Fourier transform of $h(x; r)$ is $\text{sinc}(r\omega)$ [9]. Neglecting the noise, an approximate solution to Eqn. 10 can be given by the following equation, known as pseudo-inverse filtering:

$$\hat{D}(\omega) = \frac{\text{sinc}(r\omega)}{\text{sinc}^2(r\omega) + \varepsilon} G(\omega), \quad (11)$$

where ε is a small number (around 10^{-3}) to avoid zero division at $\omega = k\pi/r$ ($k = \pm 1, \pm 2, \dots$). If $G(\omega)$ is non-zero at these frequencies, it is overly amplified (scaled by $1/\varepsilon$), which results in oscillation in the deblurred image. As it is often the case that $|G(\omega)|$ is a decreasing function with respect to $|\omega|$, major oscillation occurs at $\omega = \pm\pi/r$, which emerges as striped artifacts with an interval of $2r$ pixels.

Suppose we deblur a signal that has been blurred with radius r by a pseudo-inverse filter with radius R . Then at the major oscillation frequency $\omega = \pi/R$, we obtain the following equation from Eqns. 10 and 11 (similar for $\omega = -\pi/R$):

$$\hat{D}(\pi/R) = \frac{1}{\varepsilon} (\text{sinc}(\pi r/R)D(\pi/R) + N(\pi/R)). \quad (12)$$

Fig. 6(a) shows a plot of $|\hat{D}(\pi/R)|$ as a function of R , assuming that $|D(\omega)|$ is also a decreasing function and that $|N(\omega)|$ is constant (white noise) and is small compared to $|D(\omega)|$ except for high frequencies. From this plot we can expect to observe large oscillation in deblurred images for $R > r$. Therefore, the maximum radius with which pseudo-inverse filtering does not produce large oscillation is estimated to be the true blur radius. The above discussion is also applicable to the 2D case, as the Fourier transform of Eqn. 1 has a similar shape to circular sinc functions [9].

For each candidate radius r_j , we apply pseudo-inverse filtering to an input photograph with that radius, and we measure the amplitude of oscillation by the ratio of the number of pixels within each segment whose values are out of range $[\theta_{c,min} - \delta, \theta_{c,max} + \delta]$, where $[\theta_{c,min}, \theta_{c,max}]$ is the original range of pixel values within that segment of an input photograph for each color channel c , and δ is a small number around 0.1. This “oscillation measure” can be easily computed for arbitrarily-shaped segments. For reliability, however, we exclude too small or thin segments (e.g., under 100 pixels). From a set of blur radii $\{r_j\}$, we identify the maximum radius having the oscillation measure below a certain threshold as the true blur radius. If this measure never exceeds the threshold, which typically occurs for segments with minimal color variance, we do not make an estimate for those segments.

A blur radius field $r_{org}(x, y)$ is obtained by stitching the estimated blur radii. Values in segments where no estimate

was made as described above are interpolated from surrounding segments. We apply some smoothing to $r_{org}(x,y)$ in order to suppress occasional spurious estimates, and also to reduce step transitions that could lead to discontinuities in refocused images.

From a blur radius field $r_{org}(x,y)$ and deblurred images $d_{c,j}(x,y)$, we can reconstruct a latent image $l_c(x,y)$ by Eqn. 3. As we cannot guarantee the blur estimation to be perfect, we provide users with a simple drawing interface in which pixel intensity corresponds to the size of a blur radius, so that users can interactively modify the estimated blur radius field. Modification to the blur radius field is immediately reflected in the latent image.

3.4. Interactive Refocusing

To provide users with intuitive refocusing parameters, we associate a depth map $z(x,y)$ of the scene with the original blur radius field $r_{org}(x,y)$ through the ideal thin lens model [30]:

$$z(x,y) = \frac{F_0 v_0}{v_0 - F_0 - q_{org}(x,y) f_0}, \quad (13)$$

F_0, f_0 , and v_0 are the original camera parameters, which represent the focal length, the f-number, and the distance between the lens and the image plane, respectively, and $q_{org}(x,y)$ is the original *signed* blur radius field, such that $r_{org}(x,y) = |q_{org}(x,y)|$. The sign of $q_{org}(x,y)$ is related to the original focused depth $z_0 = F_0 v_0 / (v_0 - F_0)$ as: $q(x,y) < 0$ for $z(x,y) < z_0$, and $q(x,y) > 0$ for $z(x,y) > z_0$. As we can only estimate $r_{org}(x,y)$, we let users draw binary masks to specify the sign as shown in Fig. 4. Rough masks seem sufficient. Our drawing interface provides users with graph-cut image segmentation capability [7].



Figure 4. Top row: Input photographs. Bottom row: Corresponding masks. White indicates negative (nearer than the original focused depth), and black positive (farther).

Suppose that we change the camera parameters to F, f , and v , then a new *signed* blur radius field $q_{new}(x,y)$ is derived by using Eqn. 13 as

$$\frac{F_0 v_0}{v_0 - F_0 - q_{org}(x,y) f_0} = \frac{F v}{v - F - q_{new}(x,y) f}, \quad (14)$$

where we eliminated $z(x,y)$ to directly associate $q_{new}(x,y)$ with $q_{org}(x,y)$. Solving Eqn. 14 for $q_{new}(x,y)$ leads to

$$q_{new}(x,y) = \frac{f_0 v F}{f v_0 F_0} q_{org}(x,y) + \frac{v_0 F_0 (v - F) - v F (v_0 - F_0)}{f v_0 F_0}, \quad (15)$$

from which a new (unsigned) blur radius field to be added to the latent image is obtained as $r_{new}(x,y) = |q_{new}(x,y)|$.

The original camera parameters F_0, f_0 , and v_0 may be obtained from EXIF data [1] embedded in a JPEG file created by most of the recent digital cameras. However, some parameters are often unavailable, and EXIF data itself may not be available from converted or edited image files. In addition, it is not necessarily intuitive to manipulate the actual values when handling an image, not a camera. Therefore, we present three simplified versions of Eqn. 15, in which relative camera parameters are used.

$$q_{new}(x,y) = (v_r q_{org}(x,y) + A_0 (v_r - 1)) / f_r, \quad (16)$$

$$q_{new}(x,y) = (q_{org}(x,y) + q_{ofs}) / f_r, \quad (17)$$

$$q_{new}(x,y) = (u_r q_{org}(x,y) + q_{max} (1 - u_r)) / f_r, \quad (18)$$

where $v_r \equiv v/v_0$ is a relative image plane distance, $A_0 \equiv F_0/f_0$ is the original aperture, $f_r \equiv f/f_0$ is a relative f-number, $q_{ofs} \equiv ((v - F) - (v_0 - F_0))/f_0$ is a blur radius offset, $u_r \equiv vF/v_0 F_0$ is a relative product of the image plane distance and the focal length, and $q_{max} \equiv (v_0 - F_0)/f_0$ is the maximum blur radius. Eqn. 16 is derived by assuming the focal length to be constant as $F = F_0$, hence it has a good analogy to changing focus using a real camera. Eqn. 17 assumes $vF = v_0 F_0$. Though it is not realistic to change the parameters in this manner when handling a real camera, users can intuitively manipulate blur with a simple offset q_{ofs} . Eqn. 18 assumes $v - F = v_0 - F_0$. This is useful for refocusing among near objects while keeping far objects unaffected, as q_{max} corresponds to $z = \infty$ in Eqn. 13.

Using any one of the above equations, users can interactively do the following three types of refocusing operations.

Changing depth-of-field. This operation can be done by changing relative f-number f_r . Increasing f_r extends the depth-of-field, whereas decreasing f_r makes it shallower.

Changing focus. This can be done by changing v_r, q_{ofs} , or u_r depending on the refocusing equation in use. The other parameters A_0 and q_{max} can also be adjusted, which we typically set to $\max\{r_{org}(x,y)\}$ for good refocusing effects.

Auto-focusing. Users can simply specify a point in a photograph which they want to be in focus. An appropriate value is automatically computed for the parameter of the selected refocusing equation so that $q_{new}(x,y) = 0$ at the specified point (x_s, y_s) as

$$\begin{aligned} v_r &= A_0 / (A_0 + q_{org}(x_s, y_s)), \\ q_{ofs} &= -q_{org}(x_s, y_s), \\ u_r &= q_{max} / (q_{max} - q_{org}(x_s, y_s)). \end{aligned} \quad (19)$$

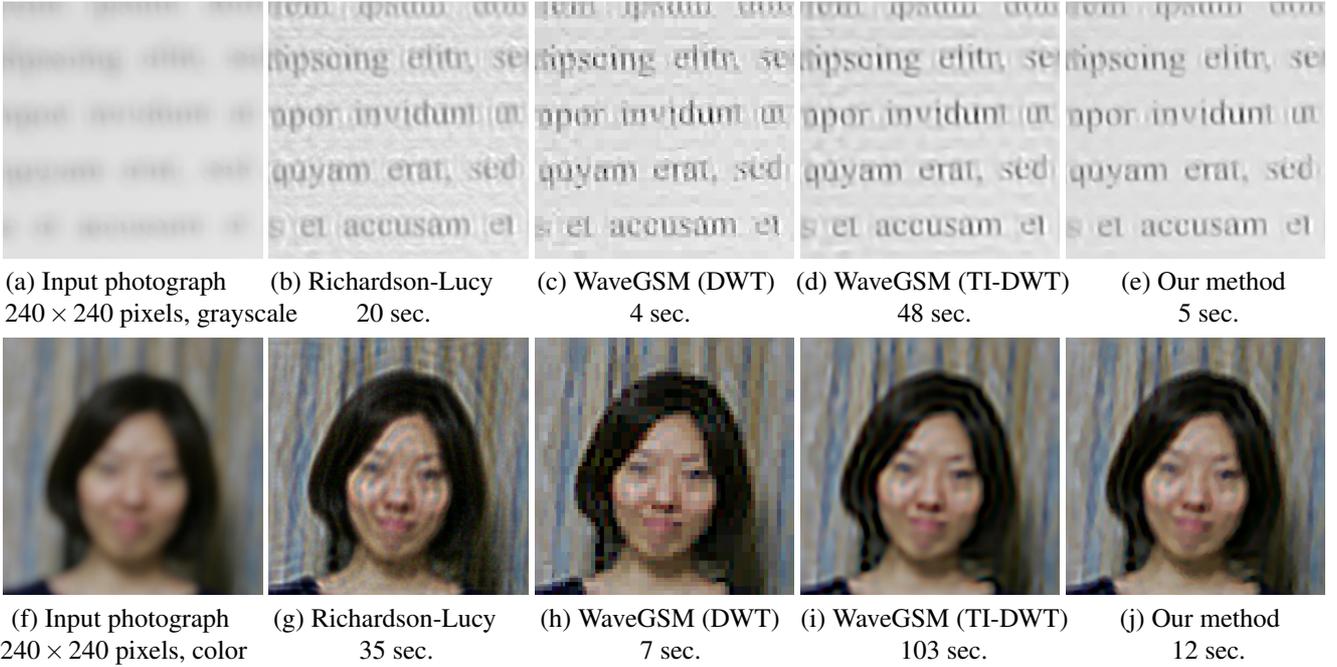


Figure 5. Comparison of four deconvolution methods and their computation times.

4. Results

All of the input photographs shown in this paper were taken with a Canon EOS-1D Mark II camera and a Canon EF 28-70mm wide aperture (F2.8) lens. The image format was JPEG with sRGB color space (gamma-corrected with $\gamma = 2.2$). We inverted this gamma-correction during deconvolution and blur estimation.

We first demonstrate the performance of our blur estimation and deconvolution methods for uniform defocus blur. For the images shown in Figs. 5(a)(f), in which the scenes have approximately uniform depths, we plotted their oscillation measure in Fig. 6(b), treating the whole image as one segment. The arrows show the estimated blur radii with a threshold of 0.01, which are 11 pixels for Fig. 5(a) and 7 pixels for Fig. 5(f). These results conform to visual inspection as shown in Fig. 7. Fig. 7 also shows that the number of out-of-range pixels (see Sec. 3.3) begins to increase as the pseudo-inverse filter radius exceeds the true blur radius.

Based on the estimated blur radii, we applied our deconvolution method, along with Richardson-Lucy [24, 33], WaveGSM with ordinary DWT, and that with TI-DWT. Fig. 5 shows the results. Since Richardson-Lucy does not exploit explicit image priors, it produced less sharp images with noise (between the alphabets in Fig. 5(b)) and halo artifacts (around the hair and face in Fig. 5(g)). WaveGSM with DWT resulted in blocky images as expected (see Sec. 3.2). Our method produced better (for the alphabet image) or comparable (for the face image) results as compared to WaveGSM with TI-DWT, running about 10 times faster.

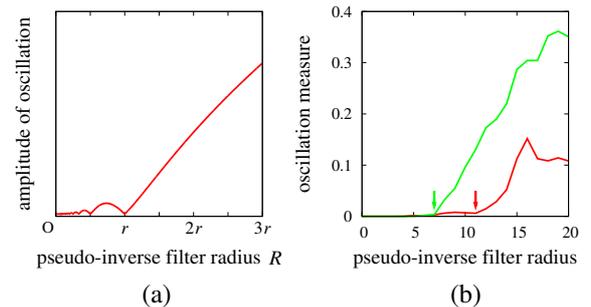


Figure 6. (a) Plot of the amplitude of oscillation $|\hat{D}(\pi/R)|$. (b) Plots of the oscillation measure for Fig. 5(a) (red) and Fig. 5(f) (green).

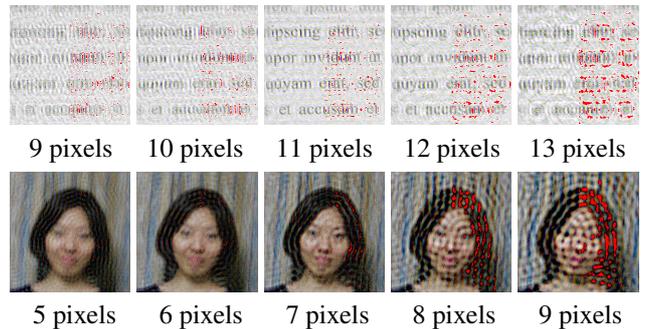


Figure 7. Results of pseudo-inverse filtering for Figs. 5(a)(f) with different blur radii. The out-of-range pixels are shown in red in the right half of each image.

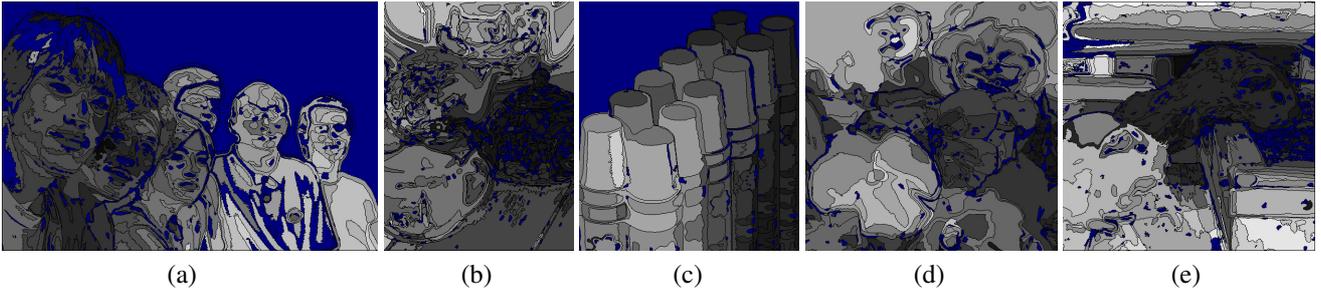


Figure 8. Results of our local blur estimation shown in gray-level. The maximum intensity (white) corresponds to a blur radius of 10 pixels. The blue regions indicate that no estimate was made there. The black lines show the segmentation boundaries.

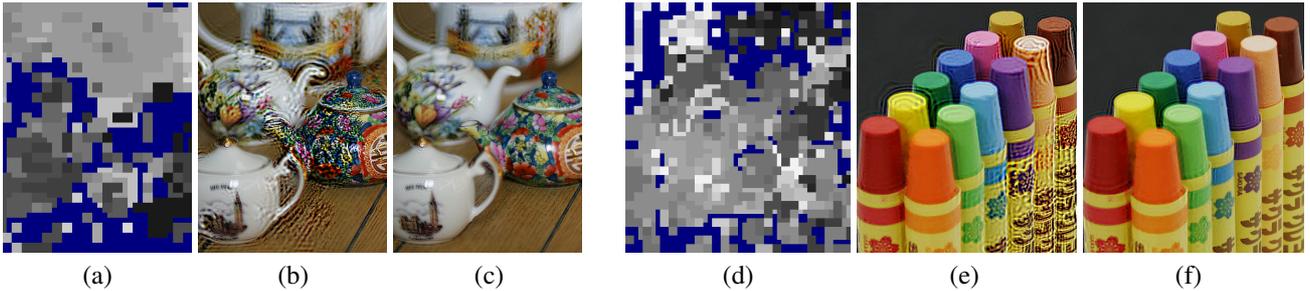


Figure 9. Comparison with the existing blur estimation method [29]. (a) Estimation result for the teapot image shown in Fig. 10(a). (b) Latent image based on (a). (c) Latent image based on our estimate shown in Fig. 8(b). (d) Estimation result for the crayon image shown in Fig. 12(a). (e) Latent image based on (d). (f) Latent image based on our estimate shown in Fig. 8(c).

Next, we show several local blur estimation results in Fig. 8. The input photographs are shown in the leftmost images in Figs. 1, 10, 12, 13, and 14. We performed relatively fine segmentation to ensure estimation locality. The estimated radii approximately correspond to the scene depths. For comparison, we applied the spatially-variant blur estimation method by Özkan et al. [29]. This method is based on local Fourier transform, hence it employs rectangular segmentation. The results are shown in Figs. 9(a)(d). It failed in regions around object boundaries and also failed to identify small blur radii, leading to noisy latent images as shown in Figs. 9(b)(e). The corresponding latent images based on our blur estimation are shown in Figs. 9(c)(f).

Next, we show an example of the user intervention mentioned in Sec. 3.3. Fig. 10(b) shows an image representing the estimated blur radius field after smoothing. Users can draw on this image to locally increase/decrease the values as shown in Fig. 10(c), for better visibility (Fig. 10(f) top) and ringing reduction (Fig. 10(f) middle and bottom). This can be done in an esthetic sense to obtain a visually pleasing latent image, and the edited blur radius field needs not correspond to the scene depth. This user editing operation took from a few to ten minutes for our examples shown below.

Finally, we show several refocusing examples in Figs. 12, 13, and 14, in which we changed the depth-of-field and moved the focus nearer to or farther from the camera. Out-

of-focus objects became sharp after they were brought into focus, as can be seen in the floret symbol at the bottom of the red crayon in Fig. 12(c) and the furry texture of the nearer marmot in Fig. 14 right.

When synthesizing Fig. 12(c) from Fig. 12(a), we used the refocusing equation Eqn. 16, which simulates focus changes of a real camera (see Sec. 3.4). We obtained the synthesis result that well approximates a real photograph shown in Fig. 12(d). For Fig. 1, we used Eqn. 17 for simple manipulation of blur radii. For Figs. 13 and 14, we used Eqn. 18 to keep distant objects unaffected as they are too blurry to be fully restored.

For an image size of 512×512 , our deconvolution took about 1 minute for each blur radius r_j , and the blur estimation 15 seconds on an Intel Pentium4 3.2GHz CPU. Although the theoretical time complexity is $O(P \log P)$, it seems $O(P)$ computation is dominant, and the deconvolution took 16 minutes and the blur estimation 4 minutes for a 4Mpixel image. Refocusing can be performed in real-time.

5. Conclusion

This paper has presented a method of digital refocusing from a single photograph, which allows users to interactively change focus and depth-of-field of a photograph after taking it with an unmodified conventional camera.

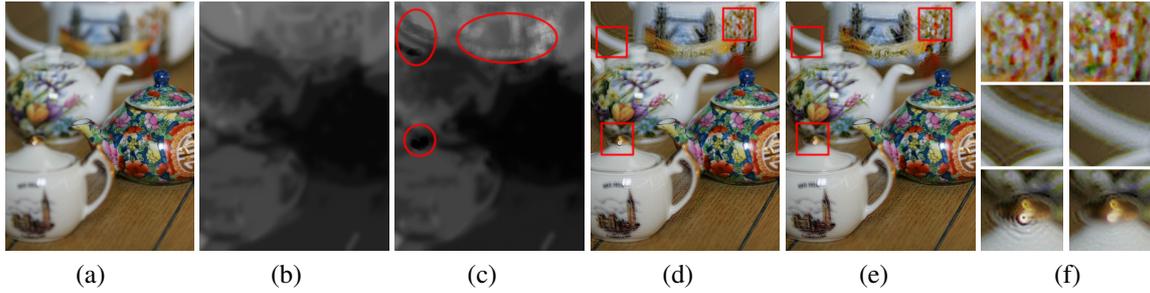


Figure 10. Example of user intervention for a blur radius field. (a) Input photograph. (b) Blur radius field after filling in the undefined (blue) regions in Fig. 8(b) and after smoothing. (c) Edited blur radius field. The red circles indicate the edited regions. (d) Latent image based on (b). (e) Latent image based on (c). (f) On the left are magnified crops from the red rectangles in (d) (before editing), and on the right from the corresponding red rectangles in (e) (after editing).

5.1. Limitations

We assumed that spatially-variant blur in an input photograph can be locally approximated by a uniform defocus blur. This directly leads to the following limitations.

First, in order for the blur estimation to be reliable, objects in a photograph should be larger than the blur radius around them, so that local segments contain a uniform blur with enough sample pixels. Hence, estimation will be erroneous for small or thin objects (e.g., a strand of hair). In other words, though our blur estimation method can handle depth discontinuities, these should not occur frequently.

Second, since depth-of-field effects are locally modeled as convolution by a single PSF, translucent objects are not accounted for. A similar problem occurs around occlusion boundaries [2], which we alleviated by smoothing a blur radius field and by blending deblurred images. The quality of refocused images will degrade particularly if occlusion boundaries frequently appear in a scene (e.g., bars of a cage), which is the case we already exclude in this paper as described above.

Third, sources of image degradation other than defocus blur, such as motion blur and image compression artifacts, can disrupt our blur estimation and deconvolution algorithms. Over/under-exposures also lead to loss of information, breaking the linear relationship between pixel values and captured light intensities. Blur estimation can be still conducted by excluding affected regions, but de-



Figure 11. Left: Saturated input photograph. Right: Result of deblurring.

convolution will produce artifacts around there as shown in Fig. 11. Transparent objects and specular highlights also induce similar artifacts as they distort the PSF shape.

5.2. Issues and Future Work

Along with the above limitations, there are several issues for our method to be discussed which suggest future research directions.

We used a simple disc PSF model, which seems sufficient for our PSF (calibrated and shown in the inset). Nevertheless, it is worth considering the use of more complex models and calibrated PSFs depending on a target imaging system.



It would be interesting to consider applying heavy-tailed priors also to blur estimation, which we did not in this paper because: we knew that the defocus PSF was a disc, which is much stronger prior knowledge about the PSF shape; and we assumed the blur to be uniform within each segment, which may be interpreted as a heavy-tailed prior that allows discontinuities in a blur radius field occasionally at segment boundaries. For better blur estimation, it would also be useful to improve segmentation quality.

We provided a means of modifying a blur radius field to fix ringing artifacts that may still remain. Skilled retouching software users could further improve the quality by directly working on the latent images. We would like to consider developing example-based touch-up tools for ordinary users.

Acknowledgments

We would like to thank many people including: anonymous reviewers for their constructive criticisms; Kenji Shirakawa, Hisashi Kazama, and Shingo Yanagawa (TOSHIBA) for their suggestions and help; and Tsuneya Kurihara (HITACHI) for proofreading the paper. The first author is grateful to Saori Horiuchi for her continuing support.

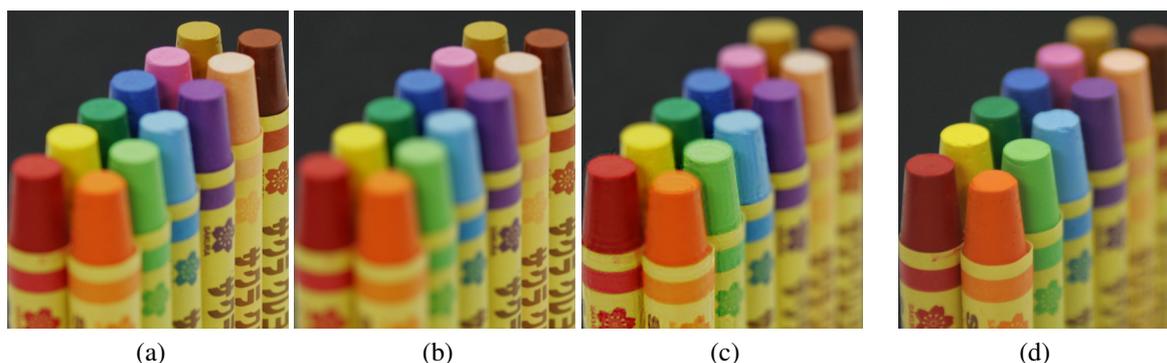


Figure 12. (a) Input photograph, focused on the brown crayon. (b) Created image with a shallow depth-of-field. (c) Created image, refocused on the orange crayon. (d) Ground truth photograph, focused on the orange crayon.



Figure 13. Left: Input photograph, focused on the flower in the center. Right: Created image, refocused on the flower in the top right corner.

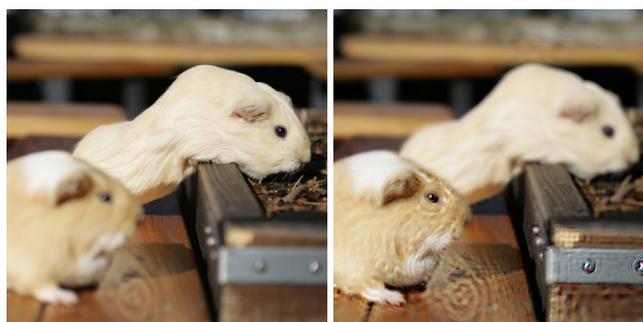


Figure 14. Left: Input photograph, focused on the farther marmot. Right: Created image, refocused on the nearer marmot.

References

- [1] EXIF: exchangeable image file format for digital still camera. <http://it.jeita.or.jp/document/publica/standard/exif/english/jeida49e.htm>.
- [2] N. Asada, H. Fujiwara, and T. Matsuyama. Seeing behind the scene: analysis of photometric properties of occluding edges by the reversed projection blurring model. *IEEE Trans. Pattern Anal. Machine Intell.*, 20(2):155–167, 1998.
- [3] O. Axelsson. *Iterative Solution Methods*. Cambridge University Press, 1996.
- [4] M. R. Banham and A. K. Katsaggelos. Digital image restoration. *IEEE Signal Processing Magazine*, 14(2):24–41, 1997.
- [5] J. Biemond, R. L. Lagendijk, and R. M. Mersereau. Iterative methods for image deblurring. *Proceedings of the IEEE*, 78(5):856–883, 1990.
- [6] J. M. Bioucas-Dias. Bayesian wavelet-based image deconvolution: a GEM algorithm exploiting a class of heavy-tailed priors. *IEEE Trans. Image Processing*, 15(4):937–951, 2006.
- [7] Y. Y. Boykov and M.-P. Jolly. *Interactive graph cuts* for optimal boundary & region segmentation of objects in N-D images. In *Proc. IEEE Int. Conf. Computer Vision*, pages 105–112, 2001.
- [8] P. J. Burt and R. J. Kolczynski. Enhanced image capture through fusion. In *Proc. IEEE Int. Conf. Computer Vision*, pages 173–182, 1993.
- [9] M. Cannon. Blind deconvolution of spatially invariant image blurs with phase. *IEEE Trans. Acous., Speech, and Sig. Processing*, 24(1):58–63, 1976.
- [10] L. Chen and K.-H. Yap. Efficient discrete spatial techniques for blur support identification in blind image deconvolution. *IEEE Trans. Signal Processing*, 54(4):1557–1562, 2006.
- [11] R. R. Coifman and D. L. Donoho. Translation-invariant denoising. In *Wavelets and Statistics*, volume 103 of *Lecture Notes in Statistics*, pages 125–150. Springer-Verlag, 1995.
- [12] D. Comaniciu and P. Meer. Robust analysis of feature spaces: color image segmentation. In *Proc. IEEE Conf. Computer Vision and Pattern Recog.*, pages 750–755, 1997.
- [13] E. R. Dowski and G. E. Johnson. Wavefront coding: a modern method of achieving high performance and/or low cost imaging systems. In *Proc. SPIE 3779*, pages 137–145, 1999.
- [14] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. *ACM Trans. Graphics*, 25(3):787–794, 2006.
- [15] M. A. T. Figueiredo and R. D. Nowak. An EM algorithm for wavelet-based image restoration. *IEEE Trans. Image Processing*, 12(8):906–916, 2003.
- [16] T. Georgeiv, K. C. Zheng, B. Curless, D. Salesin, S. Nayar, and C. Intwala. Spatio-angular resolution tradeoff in integral photography. In *Proc. Eurographics Symposium on Rendering*, pages 263–272, 2006.

- [17] A. Isaksen, L. McMillan, and S. J. Gortler. Dynamically reparameterized light fields. In *Proc. ACM SIGGRAPH 2000*, pages 297–306, 2000.
- [18] J. Kautsky, J. Flusser, B. Zitová, and S. Šimberová. A new wavelet-based measure of image focus. *Pattern Recognition Letters*, 23(14):1785–1794, 2002.
- [19] A. Kubota and K. Aizawa. Reconstructing arbitrarily focused images from two differently focused images using linear filters. *IEEE Trans. Image Processing*, 14(11):1848–1859, 2005.
- [20] D. Kundur and D. Hatzinakos. Blind image deconvolution. *IEEE Signal Processing Magazine*, 13(3):43–64, 1996.
- [21] R. L. Lagendijk and J. Biemond. Block-adaptive image identification and restoration. In *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, pages 2497–2500, 1991.
- [22] S.-H. Lai, C.-W. Fu, and S. Chang. A generalized depth estimation algorithm with a single image. *IEEE Trans. Pattern Anal. Machine Intell.*, 14(4):405–411, 1992.
- [23] A. Levin. Blind motion deblurring using image statistics. In *Proc. 20th Conf. Neural Info. Proc. Systems*, 2006.
- [24] L. B. Lucy. An iterative technique for the rectification of observed distributions. *The Astronomical Journal*, 79(6):745–754, 1974.
- [25] M. McGuire, W. Matusik, H. Pfister, J. F. Hughes, and F. Durand. Defocus video matting. *ACM Trans. Graphics*, 24(3):567–576, 2005.
- [26] S. K. Nayar and Y. Nakagawa. Shape from focus. *IEEE Trans. Pattern Anal. Machine Intell.*, 16(8):824–831, 1994.
- [27] R. Ng. Fourier slice photography. *ACM Trans. Graphics*, 24(3):735–744, 2005.
- [28] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan. Light field photography with a hand-held plenoptic camera. Tech. Rep. CSTR 2005-02, Stanford Computer Science, Apr. 2005.
- [29] M. K. Özkcan, A. M. Tekalp, and M. I. Sezan. Identification of a class of space-variant image blurs. In *Proc. SPIE 1452*, pages 146–156, 1991.
- [30] A. P. Pentland. A new sense for depth of field. *IEEE Trans. Pattern Anal. Machine Intell.*, 9(4):523–531, 1987.
- [31] P. Pérez, M. Gangnet, and A. Blake. Poisson image editing. *ACM Trans. Graphics*, 22(3):313–318, 2003.
- [32] R. Raskar, A. Agrawal, and J. Tumblin. Coded exposure photography: motion deblurring using fluttered shutter. *ACM Trans. Graphics*, 25(3):795–804, 2006.
- [33] W. H. Richardson. Bayesian-based iterative method of image restoration. *Journal of the Optical Society of America*, 62(1):55–59, 1972.
- [34] M. Subbarao, T. Choi, and A. Nikzad. Focusing techniques. *Optical Engineering*, 32(11):2824–2836, 1993.
- [35] M. Subbarao and N. Gurumoorthy. Depth recovery from blurred edges. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 498–503, 1988.
- [36] M. F. Tappen, B. C. Russell, and W. T. Freeman. Exploiting the sparse derivative prior for super-resolution and image demosaicing. In *3rd Int. Workshop on Statistical and Computational Theories of Vision*, 2003.
- [37] H. J. Trussell and B. R. Hunt. Image restoration of space-variant blurs by sectional methods. *IEEE Trans. Acous., Speech, and Sig. Processing*, 26:608–609, 1978.

- [38] Y. Xiong and S. A. Shafer. Depth from focusing and defocusing. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 68–73, 1993.

Appendix A

This appendix briefly describes a derivation of Eqn. 7 in Sec. 3.2 from (the x part of) Eqn. 6.

Assuming that the noise \mathbf{n}_x in the gradient domain can be modeled as a Gaussian with variance w and that the prior is independently applicable to each pixel location i , the posterior distribution of a latent gradient \mathbf{d}_x given an observation \mathbf{g}_x is given as

$$p(\mathbf{d}_x | \mathbf{g}_x) \propto p(\mathbf{g}_x | \mathbf{d}_x) p(\mathbf{d}_x) \propto \exp\left(-\frac{\|\mathbf{g}_x - \mathbf{H}\mathbf{d}_x\|^2}{2w}\right) \prod_{i=1}^P p(\mathbf{d}_x[i]). \quad (\text{A.1})$$

The latent gradient is estimated as the maximizer of (the logarithm of) Eqn. A.1 as

$$\hat{\mathbf{d}}_x = \arg \max_{\mathbf{d}_x} \left\{ -\frac{\|\mathbf{g}_x - \mathbf{H}\mathbf{d}_x\|^2}{2w} + \sum_{i=1}^P \ln p(\mathbf{d}_x[i]) \right\}, \quad (\text{A.2})$$

leading to non-linear optimization because the prior term is not quadratic: $\ln p(\mathbf{d}_x[i]) = -|\mathbf{d}_x[i]|^\alpha / \beta$ with $\alpha = 0.3$ (see Eqn. 5).

In order to solve Eqn. A.2 efficiently, we follow the WaveGSM approach, and we represent the heavy-tailed prior as a Gaussian scale mixture (GSM) as

$$p(\mathbf{d}_x[i]) = \int_0^\infty p(\mathbf{d}_x[i]|s) p(s) ds, \quad (\text{A.3})$$

where $p(\mathbf{d}_x[i]|s)$ is a zero-mean Gaussian with scale (or variance) s , weighted by $p(s)$. Regarding s as a “missing variable,” Eqn. A.2 is turned into an expectation maximization (EM) iteration as

$$\mathbf{d}_x^{m+1} = \arg \max_{\mathbf{d}_x} \left\{ -\frac{\|\mathbf{g}_x - \mathbf{H}\mathbf{d}_x\|^2}{2w} + \sum_{i=1}^P E_i^m [\ln p(\mathbf{d}_x[i]|s)] \right\}, \quad (\text{A.4})$$

where m is an iteration count, and $E_i^m[\cdot]$ denotes the expectation with respect to $p(s|\mathbf{d}_x^m[i])$, the probability density of scale s given the current (m -th) estimate $\mathbf{d}_x^m[i]$ of the latent gradient. Since $p(\mathbf{d}_x[i]|s)$ is a Gaussian, the prior term in Eqn. A.4 now becomes

$$E_i^m [\ln p(\mathbf{d}_x[i]|s)] = E_i^m \left[-\frac{(\mathbf{d}_x[i])^2}{2s} \right] = -\frac{(\mathbf{d}_x[i])^2}{2} E_i^m \left[\frac{1}{s} \right], \quad (\text{A.5})$$

which is quadratic with respect to $\mathbf{d}_x[i]$ since $E_i^m[s^{-1}]$ is fixed during m -th EM iteration (see [6] for more details):

$$E_i^m \left[\frac{1}{s} \right] = \frac{\alpha}{\beta |\mathbf{d}_x^m[i]|^{2-\alpha}}. \quad (\text{A.6})$$

Now that the objective function to be maximized in Eqn. A.4 is quadratic, taking its derivative with respect to \mathbf{d}_x and setting it to zero leads to the following system of linear equations, as presented in Eqn. 7:

$$(\mathbf{H}^T \mathbf{H} + w \mathbf{S}^m) \mathbf{d}_x = \mathbf{H}^T \mathbf{g}_x, \quad (\text{A.7})$$

where \mathbf{S}^m is a diagonal matrix representing the prior term whose i -th element is given by Eqn. A.6, and w serves as a weighting coefficient for it, which we treat as a user-specified value.