SINGLE IMAGE SPATIALLY VARIANT OUT-OF-FOCUS BLUR REMOVAL

Stanley H. Chan and Truong Q. Nguyen

University of California, San Diego

ABSTRACT

This paper addresses an out-of-focus blur problem in which the foreground object is in focus whereas the background scene is out of focus. To recover the details of the background scene, a spatially variant blind deconvolution problem must be solved. However, spatially variant deconvolution is computationally intensive because Fourier-based methods cannot be used to handle spatially variant convolution operators. The proposed method exploits the invariant structure of the problem by first predicting the background. Then a blind deconvolution algorithm is applied to estimate the blur kernel and a coarse estimate of the image is found as a side product. Finally, the background is recovered using total variation minimization, and fused with the foreground to produce the final deblurred image.

Index Terms— Blind deconvolution, spatially variant, out-of-focus blur, image restoration.

1. INTRODUCTION

1.1. Spatially variant out-of-focus blur problem

For an image consisting of two layers of depth – a foreground object and a background scene – only one of the two can be in focus due to different depth. The one not being focused is blurred, and we refer to this type of blur as out-of-focus blur. The goal of this paper is to present a method that restores the image so that both foreground and background are sharp.

The first challenge associated with this problem is the *spatially variant* property due to different blurs occuring on the foreground and background. Spatially variant problems are computationally intensive to solve, because the blur has to be modeled as a non-circulant matrix (which cannot be diagonalized using discrete Fourier transform matrices). The second challenge is the need for *blind* deconvolution as both image and blur kernel are unknown. Blind deconvolution is difficult because simultaneous recovery of image and kernel is an ill-posed non-linear problem.

In this paper, we present a single image spatially variant blind deconvolution algorithm. The algorithm takes a single image as the input and separates the foreground and background using alpha matting methods. To handle the spatially variant issue, instead of using the classical approach which models the non-circulant convolution matrix as a sum of circulant matrices, we propose a new photometric model that allows us to transform the variant problem into an invariant problem. We show that by inpainting (filling in) the occluded region in the background image, not only does the variant problem becomes invariant, but also ringing artifacts resulting from the classical approach are suppressed. Additionally, we present an efficient blur kernel estimation algorithm that combines the concepts of blur from image gradient, strong edge selection, joint deblurring
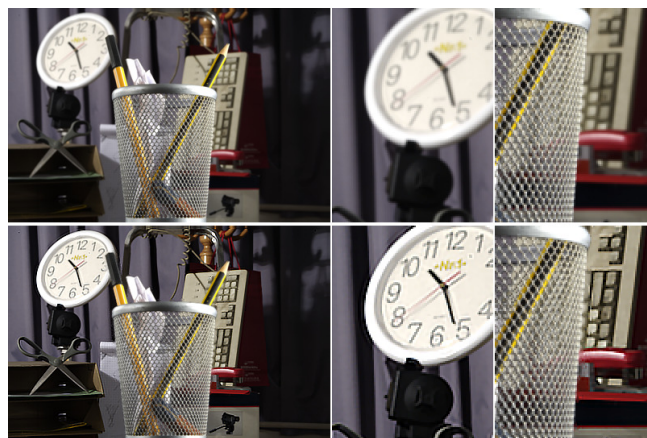
Contact: `h5chan@ucsd.edu, nguyent@ece.ucsd.edu`

Fig. 1: Result of the proposed algorithm for image No. 25. Top row: Input image. Bottom row: Deblurred image.

and kernel estimation and kernel denoising. Finally, with the estimated kernel we perform a Total Variation (TV) minimization to restore the image. Fig. 1 shows a recovered image.

1.2. Related work

In a single image restoration problem, if the blur is spatially invariant then the output is related to the input as $\mathbf{g} = \mathbf{h} * \mathbf{f} + \eta$, where $\mathbf{g}$ is the observed blur image, $\mathbf{f}$ is the unknown sharp image, $\mathbf{h}$ is the blur kernel, η is the noise and $*$ denotes convolution.

If $\mathbf{h}$ is known, classical approaches such as TV minimization can recover the image reasonably well. With the new implementation by Chan et al [1], solving a TV problem can be done in less than a second for a moderate sized image in MATLAB. To achieve high quality results, more sophisticated algorithms such as [2] can be used. If $\mathbf{h}$ is unknown, blind deconvolution methods are needed to repeatedly estimate the blur kernel and predict the underlying image in an alternating minimization form. A fast and reliable blind deconvolution algorithm using image gradients is presented in [3] and an improved method is presented in [4].

If $\mathbf{h}$ is spatially variant, we can segment the image into regions (e.g., using alpha matte [5]) where each region is approximately invariant. This idea has been widely used in motion blur problems such as [6–8]. In [7], Jia estimates the blur kernel by analyzing the transparent area of the alpha matte. He then uses a spatially variant Lucy-Richardson algorithm to deblur the image. However, as discussed in [7], ringing artifacts exist even if the true blur kernel is known. Similar observations can be found in [8].

The reason for the ringing artifacts is due to a popular model by Nagy and O'Leary [9] which expresses a spatially variant convolu-

tion matrix as a sum of invariant matrices:

$$\mathbf{H} = \mathbf{E}_1\mathbf{H}_1 + \mathbf{E}_2\mathbf{H}_2 + \cdots + \mathbf{E}_p\mathbf{H}_p, \tag{1}$$

where $\mathbf{E}_k$ is a diagonal indicator matrix with the (i,i)-th entry being either 1 or 0, and $\mathbf{H}_k$ is a block circulant matrix constructed from the blur kernel $\mathbf{h}_k$. This model is insufficient in handling the occlusion boundaries (see Section 2) and previous discussion on this issue can be found in [10–12].

2. BLUR MODEL

2.1. Classical Model

The insufficiency of the model presented in [9] is that according to Eq. 1, a sharp foreground and blurred background image is formed by

$$\mathbf{g} = \alpha(\mathbf{h}_F * \mathbf{f}) + (1 - \alpha)(\mathbf{h}_B * \mathbf{f}), \tag{2}$$

where $\mathbf{h}_F$ (= delta function) and $\mathbf{h}_B$ are the blurs associated with the foreground object and the background respectively, and α is the alpha matte that indicates the location of foreground pixels. The interpretation of Eq. 2 is that the image $\mathbf{f}$ is first blurred using $\mathbf{h}_F$ and $\mathbf{h}_B$, and then cropped and combined according to α.

If the classical model *were* valid for the formation of a sharp foreground and blurred background image, then one should be able to recover the image (reasonably well) by using methods such as [1], [7] or [8]. However, even with a good estimate of the blur kernel and a fine-tuned algorithm, ringing artifacts still appear at the foreground object boundary as shown in the left of Fig. 2.

To further understand the problem of the classical model, we synthesize a blurred image using Eq. 2. The right of Fig. 2 is a simulation of Eq. 2 in an extreme situation where $\mathbf{h}_F$ is a delta function and $\mathbf{h}_B$ is a "disk" function with large radius. Unwanted color bleeding is observed around the object boundary, which is wrong because the foreground color should not contribute to the background blur. The background color should be sky blue in the circled region.
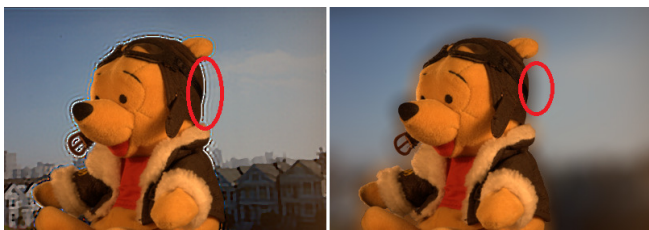

Fig. 2: Insufficiency of Eq. 1. Left: Result of spatially variant TV minimization [1]. Right: Simulation of Eq. 2 with $\mathbf{h}_F$ being a delta function and $\mathbf{h}_B$ being a "disk" function with large radius.

2.2. Proposed Model

The proposed model follows from the work by Asada et al [10], and has been previously used in [11, 12]. In [10], the authors show that if the foreground is in focus while the background is out of focus, the observed foreground object boundary is a weighted average of the *background* only. More precisely, suppose that the image $\mathbf{f}$ can be written as $\mathbf{f} = \alpha\mathbf{f}_F + (1 - \alpha)\mathbf{f}_B$, where $\mathbf{f}_F$ denotes the foreground image and $\mathbf{f}_B$ denotes the background image, the observed image according to [10] should be

$$\mathbf{g} = \alpha\mathbf{f}_F + (1 - \alpha)(\mathbf{h}_B * \mathbf{f}_B). \tag{3}$$

The interpretation of Eq. 3 is that the observed image $\mathbf{g}$ is the composition of $\mathbf{f}_F$ and $\mathbf{h}_B * \mathbf{f}_B$. Note that the convolution of $\mathbf{h}_B$ is applied to the $\mathbf{f}_B$ only, and is independent of $\mathbf{f}_F$.

The new model implies that if we can predict the occluded area in $\mathbf{h}_B * \mathbf{f}_B$ (since a large area of $\mathbf{h}_B * \mathbf{f}_B$ is occluded by $\mathbf{f}_F$), recovering the entire image $\mathbf{f}$ can be achieved by first recovering $\mathbf{f}_B$ from $\mathbf{h}_B * \mathbf{f}_B$ and then fusing $\mathbf{f}_B$ with $\mathbf{f}_F$ ($\mathbf{f}_F = \alpha\mathbf{g}$ can be pre-determined using $\mathbf{g}$ and α). Note that recovering $\mathbf{f}_B$ from $\mathbf{h}_B * \mathbf{f}_B$ is spatially *invariant*. In other words we have transformed a spatially variant problem to a spatially invariant problem, which is evidently more efficient to solve than the brute force minimization [12].

3. PROPOSED ALGORITHM

The proposed algorithm consists of three key components: inpainting the background, kernel estimation and image restoration (see Algorithm 1). Discussion on finding the alpha matte is skipped and we refer to [5] for a thorough survey.

Algorithm 1 Proposed Algorithm

Input: $\mathbf{g}$ and α.
Step 1: Inpainting background (Section 3.1)
$\mathbf{g}_B = \text{inpainting}(\mathbf{g}, \alpha)$.
Step 2: Kernel estimation (Section 3.2)
Initialize $\mathbf{f} = \mathbf{g}_B$.
for $i = 1 : m$ **do**
  $\tilde{\mathbf{f}} = \text{shock filter}(\mathbf{f})$.
  $\nabla\mathbf{f}^s = \text{edge selection}(\tilde{\mathbf{f}})$.
  $\tilde{\mathbf{h}} = \underset{\mathbf{h}}{\text{argmin}} \, \|\nabla\mathbf{f}^s * \mathbf{h} - \nabla\mathbf{g}_B\|^2 + \gamma\|\mathbf{h}\|^2$.
  $\mathbf{h} = \underset{\mathbf{h}}{\text{argmin}} \, \nu\|\mathbf{h} - \tilde{\mathbf{h}}\|^2 + \|\mathbf{h}\|_{TV}$.
  $\mathbf{f} = \underset{\mathbf{f}}{\text{argmin}} \, \|\mathbf{h} * \mathbf{f} - \mathbf{g}_B\|^2 + \lambda\|\nabla\mathbf{f} - \nabla\mathbf{f}^s\|^2$.
end for
Step 3: Image restoration (Section 3.3)
$\mathbf{f}_B = \underset{\mathbf{f}}{\text{argmin}} \, \mu\|\mathbf{h} * \mathbf{f} - \mathbf{g}_B\|^2 + \|\mathbf{f}\|_{TV}$.
Output: $\mathbf{f} = \alpha\mathbf{f}_F + (1 - \alpha)\mathbf{f}_B$, where $\mathbf{f}_F = \alpha\mathbf{g}$.

3.1. Inpainting Background

Given a blurry observation $\mathbf{g}$ and the alpha matte α, the term $\mathbf{g} - \alpha\mathbf{f}_F$ represents the observed background with foreground pixels removed. The goal of the background inpainting step is to fill the occluded pixels in $\mathbf{g} - \alpha\mathbf{f}_F$ so that we can have an invariant deconvolution procedure to recover $\mathbf{f}_B$.
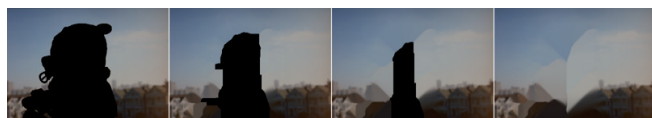

Fig. 3: Inpainting background step. From left to right: the intermediate result of inpainting at iteration 0, 20, 40 and final respectively.

Our inpainting step is based on the assumption that occluded pixels are likely to have similar color to its non-occluded neighborhood. Therefore, starting from the occlusion boundary and moving inwards, occluded pixels are replaced by the local mean of its 5×5

neighborhood. Fig. 3 illustrates a few iterations of the proposed in-painting algorithm. Note that the algorithm works well around the object boundary but works poorly in the central occluded region. However, the poorly filled pixels in the central occluded region has negligible effects to the deconvolution step as their geometric distance to the occlusion boundary is typically large. The filled background is denoted by $\mathbf{g}_B$.

3.2. Kernel Estimation

3.2.1. Shock Filter

As discussed in [3] and [4], salient edges are critical to blur kernel estimation. To obtain salient edges, we follow [3] and apply Shock filter to the observed image. Shock filtering is an iterative procedure of which the k-th iteration is given by

$$\mathbf{f}^{k+1} = \mathbf{f}^k - \beta \, \mathrm{sign}(\Delta \mathbf{f}^k) \|\nabla \mathbf{f}^k\|_1,$$

where initially $\mathbf{f}^0 = \mathbf{h}_G * \mathbf{f}_{in}$, $\mathbf{f}_{in}$ is the input to the shock filter ($\mathbf{g}_B$ in our case) and $\mathbf{h}_G$ is a Gaussian blur kernel of size 9×9 and $\sigma = 1$. $\nabla \mathbf{f} = [\mathbf{f}_x; \mathbf{f}_y]$ is the gradient of $\mathbf{f}$ and $\Delta \mathbf{f} = \mathbf{f}_x^2 \mathbf{f}_{xx} + 2\mathbf{f}_x \mathbf{f}_y \mathbf{f}_{xy} + \mathbf{f}_y^2 \mathbf{f}_{yy}$ is the Laplacian of $\mathbf{f}$. $\beta(=1)$ is the step size. The shock filtered image is denoted as $\tilde{\mathbf{f}}$ and is illustrated in Fig. 4.
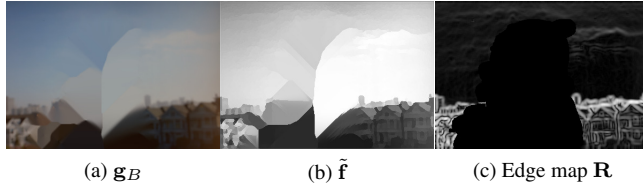

(a) $\mathbf{g}_B$ (b) $\tilde{\mathbf{f}}$ (c) Edge map $\mathbf{R}$

Fig. 4: (a) $\mathbf{g}_B$, the background image with occlusion region filled. (b) Shock filtered image $\tilde{\mathbf{f}}$ using $\beta = 1$ and 5 iterations. (c) Edge map $\mathbf{R}$ as defined in Eq. 4.

3.2.2. Edge Selection

Not all edges in the shock filtered image $\tilde{\mathbf{f}}$ can be used to estimate the blur kernel. To select informative edges from the shock filtered image, we modify the metric proposed in [4]:

$$\mathbf{R} = \frac{\sqrt{|\mathbf{h}_A * \mathbf{g}_x|^2 + |\mathbf{h}_A * \mathbf{g}_y|^2}}{\mathbf{h}_A * \sqrt{|\mathbf{g}_x|^2 + |\mathbf{g}_y|^2} + 0.5} \cdot (1 - \alpha), \tag{4}$$

where $\mathbf{h}_A$ is a 5×5 matrix with all entries being $1/25$ (i.e., a 5×5 uniform average) and $\cdot$ denotes element-wise multiplication. Small values of $\mathbf{R}$ should be ruled out, thus leading to a mask defined as

$$\mathbf{M} = \max \{\mathbf{R} - \tau_r, 0\}, \tag{5}$$

where τ_r is a threshold. Note that $[\mathbf{g}_x; \mathbf{g}_y]$, the gradients of $\mathbf{g}$, can be replaced by the gradients of $\mathbf{g}_B$ because gradients in the foreground region are discarded by $(1 - \alpha)$ in Eq. 4.

The final selected edges for kernel estimation are given by

$$\nabla \mathbf{f}^s = \nabla \tilde{\mathbf{f}} \cdot \max \left\{ \mathbf{M} \cdot \sqrt{|\tilde{\mathbf{f}}_x|^2 + |\tilde{\mathbf{f}}_y|^2} - \tau_s, 0 \right\},$$

where $\tilde{\mathbf{f}}$ is the shock filtered image, $\nabla \tilde{\mathbf{f}} = [\tilde{\mathbf{f}}_x; \tilde{\mathbf{f}}_y]$ are gradients of $\tilde{\mathbf{f}}$ and τ_s is a threshold.

3.2.3. Kernel Estimation and Refinement

With $\nabla \mathbf{f}^s = [\mathbf{f}_x^s; \mathbf{f}_y^s]$, the blur kernel can be estimated by solving the following unconstrained least-squares problem

$$\min_{\mathbf{h}} \|\nabla \mathbf{f}^s * \mathbf{h} - \nabla \mathbf{g}_B\|^2 + \gamma \|\mathbf{h}\|^2, \tag{6}$$

where $\gamma = 0.01$ is a regularization parameter. Closed-form solution for Eq. 6 exists, and is given by

$$\tilde{\mathbf{h}} = \mathcal{F}^{-1} \left[\frac{\overline{\mathcal{F}(\mathbf{f}_x^s)}\mathcal{F}(\mathbf{g}_{B_x}) + \overline{\mathcal{F}(\mathbf{f}_y^s)}\mathcal{F}(\mathbf{g}_{B_y})}{|\mathcal{F}(\mathbf{f}_x^s)|^2 + |\mathcal{F}(\mathbf{f}_y^s)|^2 + \gamma} \right],$$

where $\mathcal{F}$ is the Fourier Transform and $\overline{(\cdot)}$ is the complex conjugate.

The estimated kernel at this stage is typically noisy. The authors of [4] used a minimization procedure with sparsity constraint to denoise $\mathbf{h}$, but this method is time-consuming. In view of the fact that blur kernels are smooth, we proposed to solve the following minimization problem.

$$\min_{\mathbf{h}} \quad \nu \|\mathbf{h} - \tilde{\mathbf{h}}\|^2 + \|\mathbf{h}\|_{TV}, \tag{7}$$

where $\|\mathbf{h}\|_{TV}$ is the TV norm of $\mathbf{h}$, $\nu = 1000$ is a regularization parameter. Note that TV regularization $\|\mathbf{h}\|_{TV}$ is more suitable here than the Tikhonov regularization $\|\mathbf{h}\|^2$, because $\|\mathbf{h}\|^2$ tends to over-smooth the kernel if outliers are present.

3.2.4. Coarse Image Estimation

In the kernel estimation stage, a coarse image estimation is needed to iteratively improve the blur kernel. This step is performed by solving a least-squares problem

$$\min_{\mathbf{f}} \quad \|\mathbf{h} * \mathbf{f} - \mathbf{g}_B\|^2 + \lambda \|\nabla \mathbf{f} - \nabla \mathbf{f}^s\|^2, \tag{8}$$

where $\lambda = 0.002$ and the prior $\|\nabla \mathbf{f} - \nabla \mathbf{f}^s\|^2$ is used to preserve edges according to $\nabla \mathbf{f}^s$. The closed form solution of Eq. 8 is

$$\mathbf{f} = \mathcal{F}^{-1} \left[\frac{\overline{\mathcal{F}(\mathbf{h})}\mathcal{F}(\mathbf{g}_B) + \lambda(\overline{\mathcal{F}(\partial_x)}\mathcal{F}(\mathbf{f}_x^s) + \overline{\mathcal{F}(\partial_y)}\mathcal{F}(\mathbf{f}_y^s))}{|\mathcal{F}(\mathbf{h})|^2 + \lambda(|\mathcal{F}(\partial_x)|^2 + |\mathcal{F}(\partial_y)|^2)} \right],$$

where $\partial_x = [1, -1]$, $\partial_y = [1, -1]^T$.

3.3. Image Recovery

After the kernel estimation, an advanced image restoration method is applied to restore the background image. In particular, we consider the following TV-minimization problem

$$\min_{\mathbf{f}} \quad \mu \|\mathbf{h} * \mathbf{f} - \mathbf{g}_B\|^2 + \|\mathbf{f}\|_{TV}, \tag{9}$$

where $\mu = 7500$. Problem (9) is solved using [1]. To reduce ringing artifacts at image borders, symmetric padding is applied. The output of this problem is the deconvolved background $\mathbf{f}_B$. Together with $\mathbf{g}$ and α, the final deblurred image is $\mathbf{f} = \alpha \mathbf{f}_F + (1 - \alpha)\mathbf{f}_B$, where $\mathbf{f}_F = \alpha \mathbf{g}$ is the sharp foreground object.

4. EXPERIMENTAL RESULTS

To verify the proposed algorithm we tested over 27 training images at http://www.alphamatting.com/ in which images No.1-23

(a) Image No. 22 Input (b) α-matte (c) Background Filling Result (d) Estimated Kernel

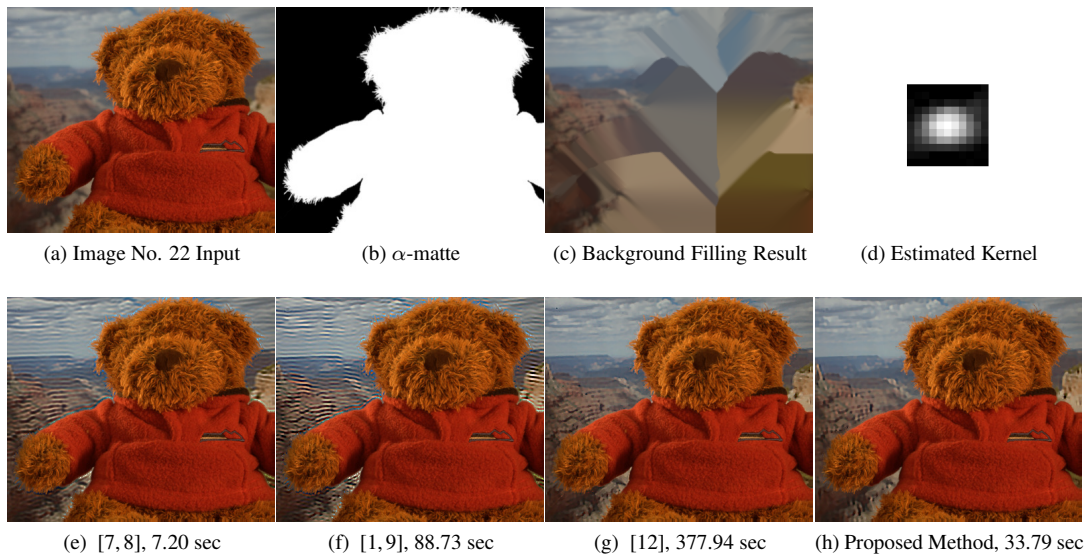(e) [7,8], 7.20 sec (f) [1,9], 88.73 sec (g) [12], 377.94 sec (h) Proposed Method, 33.79 sec

Fig. 5: Comparisons with existing spatially variant deconvolution algorithms.

are composed of a sharp foreground object placed in front of a monitor showing natural high definition 3D scenes and images No.24-27 are composed of a sharp foreground object placed in a real 3D scene. Ground truth alpha mattes are used in all of the following comparisons for the sake of fairness.

Three existing spatially variant deconvolution algorithms are compared. The first one is the spatially variant Lucy-Richardson algorithm used in [7, 8], and the second one is the variant version of [1]. In both methods, the operator **H** is defined in Eq. 1 (thus one of the subproblems in [1] is solved inexactly using conjugate gradient). The third method is the iterative reweighted least-squares (IRLS) method in [12]. [12] uses the new blur model (Eq. 3) to solve for $\mathbf{f}_F$, $\mathbf{f}_B$ and α simultaneously. However, since the ground truth α is used in our comparison, calculation of α in [12] is skipped. Other methods such as [2–4, 6] are not compared because [6] requires multiple images, [2] is non-blind, and [3, 4] are spatially invariant.

Fig. 5 shows the results of image No. 22. The run times are recorded based on a Dell XPS machine with Intel Qual Core 9550, 2.8GHz, 4GB RAM, Windows 7/ MATLAB 2010. For [12], the run time reflects the IRLS step only (kernel estimation and α estimation are excluded). For the proposed method, the run time includes kernel estimation, background filling and deblurring. Clearly, the proposed method gives better visual quality than [1, 7–9] , and is faster than [12]. More results can be found at http://videoprocessing. ucsd.edu/~stanleychan.

5. CONCLUSION

A spatially variant blind deconvolution algorithm is presented in this paper. The proposed algorithm aims to restore images consisting of a sharp foreground object and blurred background scene. By separating the foreground and background using alpha matte and inpainting the background, the spatially variant problem is transformed to an invariant problem. A blind deconvolution algorithm and TV-minimization is then applied to restore the background. Experiments show that the proposed algorithm outperforms existing spatially variant Lucy-Richardson algorithms and TV minimization.

6. REFERENCES

[1] S. Chan, R. Khoshabeh, K. Gibson, P. Gill, and T. Nguyen, "An augmented Lagrangian method for total variation video restoration," *IEEE TIP*, 2011, to appear. Preprint available at http://videoprocessing.ucsd.edu/~stanleychan/deconvtv.

[2] Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," *SIGGRAPH*, 2008.

[3] S. Cho and S. Lee, "Fast motion deblurring," in *SIGGRAPH*, 2009.

[4] L. Xu and J. Jia, "Two-phase kernel estimation for robust motion deblurring," in *ECCV*, 2010.

[5] J. Wang and M. Cohen, "Image and video matting: A survey," *Foundations and Trends in Computer Graphics and Vision*, vol. 3, no. 2, 2007.

[6] S. Cho, Y. Matsushita, and S. Lee, "Removing non-uniform motion blur from images," in *ICCV*, 2007.

[7] J. Jia, "Single image motion deblurring using transparency," in *CVPR*, 2007.

[8] S. Dai and Y. Wu, "Motion from blur," in *CVPR*, 2008.

[9] J. Nagy and D. O'Leary, "Restoring images degraded by spatially variant blur," *SIAM Journal on Scientific Computing*, vol. 19, no. 4, 1998.

[10] 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 PAMI*, vol. 20, no. 2, pp. 155–167, 1998.

[11] M. McGuire, W. Matusik, H. Pfister, J. Hughes, and F. Durand, "Defocus video matting," in *SIGGRAPH*, 2005.

[12] S. Dai and Y. Wu, "Removing partial blur in a single image," in *CVPR*, 2009.