

SPACE-VARIANT KERNEL DECONVOLUTION FOR DUAL EXPOSURE PROBLEM

Miguel Tallón¹, Javier Mateos¹, S.Derin Babacan², Rafael Molina¹, Aggelos K. Katsaggelos³

¹Departamento de Ciencias de la Computación e I.A.
Universidad de Granada, Granada,
Spain
{mtallon,jmd,rms}@decsai.ugr.es

² Beckman Institute
University of Illinois at
Urbana-Champaign, IL USA
dbabacan@illinois.edu

³Electrical Engineering Computer
Science Department
Northwestern University, Evanston,
IL USA
aggk@eecs.northwestern.edu

ABSTRACT

In this paper we propose a space-variant kernel estimation method for effective deconvolution when combining different exposure image pairs. The proposed algorithm can be applied to images blurred by both camera and object motion in an efficient manner. The blur in the long exposure shot is mainly caused by camera shake or object motion, and the noise of the underexposed image is introduced by the gain factor applied to the sensor when the ISO is set to a high value. The main idea in this work is to incorporate a spatially-varying deblurring/denoising which is applied to image patches. The method exploits kernel estimation and error measures to choose between denoising and deblurring each patch. In addition, the proposed approach estimates all necessary parameters automatically without user supervision.

1. INTRODUCTION

The blur caused by camera shake and object motion is still a challenging problem, even more so when images are taken in dim environments. There are a number of problems that need to be addressed in single image blind deconvolution, including spatially-varying blur, and saturated pixels in the long exposure shots. Utilizing an accompanying short exposure image for the deconvolution provides valuable information that significantly improves the restoration. Unfortunately, although this additional image contains accurate information about image edges, it is generally contaminated with a high level of noise and color information might be lost.

A number of methods have been developed that use different exposure image pairs for blind deconvolution [1–3]. Most of the previous work assumes a space-invariant blur kernel, which is seldomly true in practice. This assumption could cause not enough suppression of the blur in some regions, and might lead to significant artifacts in other image parts. Recent works attempt to overcome this by using space-variant blur modelling and estimation [4], or by abandoning the blur kernel estimation completely [5]. In [4], the images are divided into patches and separate blur kernels are estimated for each patch. Although it is space-variant, this approach can easily lead to blocking artifacts. Furthermore the estimated kernel generally exhibits a high level of noise due to the use of noisy observation in the kernel estimation process. In [5] a method without kernel estimation is proposed. The two images are fused into a single image by first classifying areas of the image into blurry and sharp followed by a weighted linear combination of them.

In this paper, we propose a new method to combine a long exposure blurry image with a short exposure noisy one to obtain a sharp restoration that is both noiseless and free of blur. Firstly we divide our input images in overlapped patches and estimate the blur kernel in each patch. We then employ a fully-automatic procedure to estimate the unknown image from the estimated kernels and the image pair. The proposed algorithm selectively applies deconvolution or denoising to the image patches to extract the sharp features from

the image pair. Finally, the estimation of the overlapped patches is combined using a windowing function to recover a blocking-free restoration as proposed in [6]. Experimental results demonstrate that the proposed approach provides both high quality blur and image estimates even in challenging datasets.

2. PROPOSED ALGORITHM

We assume a linear and space variant degradation model, so that the observation processes can mathematically be expressed in matrix-vector notation as

$$\mathbf{y}_1 = \mathbf{H}\mathbf{x} + \mathbf{n}_1 \quad (1)$$

$$\mathbf{y}_2 = \mathbf{x} + \mathbf{n}_2, \quad (2)$$

where \mathbf{y}_1 and \mathbf{y}_2 are the $N_y \times N_x$ observed long- and short-exposure images, respectively, represented as column vectors of size $(N_y \times N_x) \times 1$, \mathbf{x} the unknown original image, \mathbf{n}_1 and \mathbf{n}_2 the noise components, and \mathbf{H} the unknown $(N_y \times N_x) \times (N_y \times N_x)$ space-variant blur matrix.

The proposed method takes as input the image pair $\mathbf{y}_1, \mathbf{y}_2$, the number of patches in which the image will be divided into horizontally and vertically p_x and p_y , respectively; the size of our spatially-varying blur (h_y, h_x) , the percentage of overlap between patches, and the windowing function to blend the patches. The algorithm consists of the following steps:

1. Registration of the image pair (Section 2.1)
2. Computation of the image overlapping patches (Section 2.2)
3. For each patch
 - (a) Kernel estimation, followed by kernel correction (Section 2.3)
 - (b) Deconvolution of the patch (Section 2.4)
4. Blending the restored patches to form the final image using a windowing function (Section 2.5)

We represent the images in the YC_bC_r colorspace. In the deconvolution stage, the luminance component is restored using the estimated blur kernels, and finally fused with the chrominance component of the blurred image. The steps of the algorithm are explained in detail in the following sections.

2.1 Registration

Due to camera and object motion between the subsequent exposures, as well as different exposure conditions, the image pair must be photometrically and geometrically registered. For photometric registration, we apply histogram equalization of the short-exposure image to increase the brightness using the histogram of the long-exposure image. The geometric calibration is done by extracting features from both images using Surf [7] and matching them using RANSAC [8]. The quality of the restoration highly depends on the accuracy of this stage.

2.2 Computation of patches

As we have already stated, our approach to space-variant kernel estimation is based on dividing the images in overlapping image

This work was supported in part by the ‘‘Comisi3n Nacional de Ciencia y Tecnolog3a’’ under contract TIC2010-15137. Special thanks to Ramya Hebbalaguppe for the object motion blur images.

patches and estimate the blur in each patch. For a given overlapping fraction $0 \leq \text{overlap} < 1$ in each direction (we assume the same overlapping in both directions for simplicity) and a number of patches p_x and p_y in the horizontal and vertical directions, respectively, the horizontal and vertical block sizes, are

$$\text{blockSize}_z = \left\lfloor \frac{N_z}{(1 - \text{overlap})p_z + \text{overlap}} \right\rfloor, z \in \{x, y\}. \quad (3)$$

2.3 Kernel estimation

The space-variant kernel is estimated per patch using \mathbf{y}_1 and \mathbf{y}_2 . Since all following equations are applied to each patch, we use $p = (p_v, p_u)$ with $1 \leq p_v \leq p_y$ and $1 \leq p_u \leq p_x$, to refer to the specific patch (p_v, p_u) . Thus \mathbf{y}_1^p refers to patch (p_v, p_u) of the long-exposure observation; equivalently \mathbf{y}_2^p is the same patch in the short-exposure image. The matrix \mathbf{Y}_2^p represents the image patch \mathbf{y}_2^p written as a $(N_y \times N_x) \times (h_y \times h_x)$ convolution matrix. Finally, \mathbf{h}^p represents the blur kernel in patch p . Values outside the boundaries of an image patch are filled with its neighboring patch values when available or by replicating pixel values in the image borders.

Using Eqs. (1) and (2), the estimate $\hat{\mathbf{h}}^p$ of a kernel \mathbf{h}^p is found as

$$\hat{\mathbf{h}}^p = \min_{\mathbf{h}^p} \|\mathbf{y}_1^p - \mathbf{Y}_2^p \mathbf{h}^p\|^2, \quad (4)$$

subject to the constraint $0 \leq \mathbf{h}_i^p \leq 1, i = 1, \dots, (h_x \times h_y)$.

In order to solve this constrained linear least-squares problem we use the Matlab function `lsqlin`. Once the kernel has been estimated following this procedure, we normalize it to sum up one. Notice that because \mathbf{y}_1^p and \mathbf{y}_2^p are photometrically registered the constraint $\sum_i \mathbf{h}_i^p = 1$ is generally satisfied. If that were not the case we could always introduce that constraint.

2.3.1 Kernel correction stage

In this stage, we apply a correction to the estimated kernels in patches with weak texture or saturated pixels, that is when there is not enough available information for the kernel estimation. The algorithm, that is detailed in Alg. 1, corrects the kernel of a patch if the difference with its neighboring kernels is higher than a threshold, and then is replaced by the mean of the kernels of the neighboring patches including itself. The threshold is automatically set to half the range of all neighboring differences. An alternative procedure for kernel correction is proposed in [4].

Algorithm 1 Proposed Kernel Correction Algorithm.

Input: The full space-variant kernel matrix with a kernel \mathbf{h}^p per patch p

Output: The corrected space-variant kernel matrix.

1. For each kernel \mathbf{h}^p .
 - (a) Compute the mean kernel \mathbf{h}_{mean}^p as the mean of its neighbor kernel patches including \mathbf{h}^p .
 - (b) Calculate $\text{diff}^p = \|\mathbf{h}^p - \mathbf{h}_{mean}^p\|_1$.
 2. Compute the threshold as $\text{thr} = \frac{\max(\text{diff}) + \min(\text{diff})}{2}$, where diff is the vector formed by $\text{diff}^p, p = 1, \dots, (p_x \times p_y)$
 3. For each kernel \mathbf{h}^p such that $\text{diff}^p > \text{thr}$, replace \mathbf{h}^p by \mathbf{h}_{mean}^p .
-

2.4 Deconvolution

After the kernel estimation and correction steps, we need to obtain an accurate estimation of the image for each patch. Then we apply deconvolution to all image pair patches. We incorporate the Bayesian deconvolution algorithm developed in [2] with some modifications to save memory and time. We skip the details of the

deconvolution algorithm due to the lack of space and provide an overview in the following.

The key idea in [2] consists of modelling the unknowns within a hierarchical Bayesian formulation and develop a blind deconvolution algorithm which jointly estimates the unknowing image and blur. However the blur estimation in [2] does not provide good results in our case due to the small size of the patches that makes the joint estimation of the image and blur unfeasible. However, since we already estimated an accurate kernel, we casted the restoration method in [2] to deal with the easier case of image estimation when the blur is known. Starting from Eqs. (1) and (2), we assume that the noises \mathbf{n}_1 and \mathbf{n}_2 in both observed images follow independent Gaussian distributions of zero mean and variances $\beta_1^{-1}, \beta_2^{-1}$, respectively, and use a TV-prior to model the prior knowledge on the image and Gamma distributions as prior for the noise and image model parameters. A variational approach is employed to obtain an approximation of the posterior distribution of the restored image and the parameters. The image estimation for each patch p is found as

$$\begin{aligned} \mathbf{x}^p &= \Sigma_x^p (\beta_1^p (\mathbf{H}^p)^t \mathbf{y}_1^p + \beta_2^p \mathbf{y}_2^p) \\ (\Sigma_x^p)^{-1} &= \beta_1^p (\mathbf{H}^p)^t \mathbf{H}^p + \alpha^p (\Delta^u)^t \mathbf{W}^p \Delta^u \\ &\quad + \alpha^p (\Delta^v)^t \mathbf{W}^p \Delta^v + \beta_2^p \mathbf{I}, \end{aligned} \quad (5)$$

where Δ^u and Δ^v are the discrete approximations to the gradient operator in the horizontal and vertical directions respectively, \mathbf{W}^p denotes the spatially varying weighting of each derivative penalty term, and β_1^p and β_2^p are the inverses of the noise variance corresponding to the observations \mathbf{y}_1^p and \mathbf{y}_2^p , respectively. The parameter estimates are found using

$$w_j^p = (\Delta_j^u(\mathbf{x}^p))^2 + (\Delta_j^v(\mathbf{x}^p))^2, j = 1, \dots, B \quad (7)$$

$$\mathbf{W}^p = \text{diag} \left[\frac{1}{\sqrt{w_j^p} + \varepsilon} \right], j = 1, \dots, B \quad (8)$$

$$\alpha^p = \frac{B}{2 \sum_j \sqrt{w_j^p}}, j = 1, \dots, B \quad (9)$$

$$\beta_1^p = \frac{B}{\|\mathbf{y}_1^p - \mathbf{H}^p \mathbf{x}^p\|^2} \quad (10)$$

$$\beta_2^p = \frac{B}{\|\mathbf{y}_2^p - \mathbf{x}^p\|^2}, \quad (11)$$

with $B = \text{blockSize}_x \times \text{blockSize}_y$. To calculate \mathbf{x}^p in Eq. (5) and the parameters in Eqs. (7)-(11) we use an iterative procedure. Initializing \mathbf{x}^p to \mathbf{y}_1^p , the long exposure image patch and using ε a small positive value to avoid division by zero, we update the parameter using Eqs. (7)-(11) and then for those parameters we calculate a new estimate \mathbf{x}^p utilizing Eq. (5).

The explicit computation of Σ_x^p in Eq. (6) is performed in [2] via the conjugate gradient method. Because of the computational cost for such an amount of patches, we employ a different procedure for image estimation similar to the one proposed in [9] which, in our case, is computationally much more efficient than conjugate gradient. In order to solve Eq. (5) we proceed as follows. First rewrite Eq. (5) as

$$(\Sigma_x^p)^{-1} \mathbf{x}^p = \beta_1^p (\mathbf{H}^p)^t \mathbf{y}_1^p + \beta_2^p \mathbf{y}_2^p$$

and then note that at a given pixel j ,

$$1 = \frac{(\beta_1^p (\mathbf{H}^p)^t \mathbf{y}_1^p + \beta_2^p \mathbf{y}_2^p)_j}{((\Sigma_x^p)^{-1} \mathbf{x}^p)_j}, j = 1, \dots, B. \quad (12)$$

Multiplying both sides of Eq. (12) by \mathbf{x}^p , results in the following iterative image update,

$$(\mathbf{x}_{new}^p)_j = (\mathbf{x}_{old}^p)_j \times \frac{(\beta_1^p (\mathbf{H}^p)^t \mathbf{y}_1^p + \beta_2^p \mathbf{y}_2^p)_j}{(\beta_1^p (\mathbf{H}^p)^t \mathbf{H}^p \mathbf{x}_{old}^p + \alpha^p [\mathbf{D}^t \mathbf{Z}^p \mathbf{D}] \mathbf{x}_{old}^p + \beta_2^p \mathbf{x}_{old}^p)_j}, \quad (13)$$

where $\mathbf{D}^t = [(\Delta^u)^t, (\Delta^v)^t]$, $\mathbf{Z}^p = \begin{bmatrix} \mathbf{W}^p & 0 \\ 0 & \mathbf{W}^p \end{bmatrix}$, \mathbf{x}_{new}^p denotes the new estimate of \mathbf{x}^p and \mathbf{x}_{old}^p denotes the estimation of \mathbf{x}^p before the parameters have been updated. All parameters in Eqs. (7-11) are calculated utilizing \mathbf{x}_{old}^p and used in the right hand side of Eq. (13). We have also experimented with other deconvolution methods such as Richardson-Lucy [10, 11] and the method in [6], but they provided results with higher amount of ringing artifacts.

2.4.1 Object motion correction

The proposed method results in high-quality image estimates in the case of camera shake. However, as with other deconvolution algorithms, it fails to estimate the image accurately when objects are moving in the scene. Notice that the modelling $\mathbf{y}_1^p = \mathbf{H}^p \mathbf{x}^p + \mathbf{n}_1^p$, in case of object motion blur, is quite unrealistic because the motion blur will very likely be variant within the patch.

In the case of object motion, the algorithm has to separate the background pixels (which are not blurred) from the object in the foreground. This can be achieved by using a segmentation algorithm, but this is generally computationally unfeasible. In this paper, we work on an image patch basis, selecting for each one of the patches if the deconvolution algorithm or a denoised version of the noisy patch should be used. Notice that due to the short exposure time of \mathbf{y}_2 , it is expected that this observation is not affected by motion blur. Consequently better results are expected by denoising \mathbf{y}_2^p than deblurring \mathbf{y}_1^p (in patches affected by object motion blur). Furthermore if the blur estimated by the proposed method is incorrect we can always try to only remove the noise in the second observation patch \mathbf{y}_2^p .

To select between deconvolving or denoising a patch, we first compute two restorations for each patch, one using the deconvolution method described above, and the other one applying the denoising method in [12] using the inverse of the value estimated in Eq. (11) as noise variance for the patch. These restorations are denoted by \mathbf{x}_1^p and \mathbf{x}_2^p , respectively. We then calculate two error measures for each one of the two restorations as

$$\begin{aligned} error_{1k} &= \|\mathbf{y}_1^p - \mathbf{H}^p \mathbf{x}_k^p\|^2 \\ error_{2k} &= \|\mathbf{y}_2^p - \mathbf{x}_k^p\|^2, \end{aligned} \quad (14)$$

for $k = 1, 2$. Then, the denoised patch is selected as the patch estimate if both $error_{12}$ and $error_{21}$ are greater than $errThreshold$; otherwise, the deconvolved patch is chosen. The justification for this decision process comes directly from the fact that in the case of object motion, the estimated kernel contains significant errors due to the mixing of moving and static parts and will not correctly explain the two observed images simultaneously. In this case, $error_{12}$ and $error_{21}$ will have large values, and therefore, the denoising result is used in these patches. Furthermore we also select the denoised patch when the denoising method provides low data-fit errors ($error_{11} > error_{12}$), but this rarely occurs in practice.

The running time for the deconvolution method, implemented in Matlab and summarized in algorithm 2, is quite reduced, being just 2 minutes in a 2.13 Ghz core 2 Duo laptop, with a kernel size of 47×47 , 6×6 patches and 512×512 image size.

2.5 Blending of patches

Once all patches have been estimated, in the final stage to avoid blocking artifacts, we merge all restored patches using a windowing

Algorithm 2 Proposed Deconvolution Algorithm.

Input: $\mathbf{y}_1^p, \mathbf{y}_2^p$ the long- and short-exposure image patches, respectively; \mathbf{h}^p the corrected kernel estimates in patch p
Output: \mathbf{x}^p the restored patch

Set $\mathbf{x}_{new}^p = \mathbf{y}_1^p$
Do

1. Set $\mathbf{x}_{old}^p = \mathbf{x}_{new}^p$
2. Find the estimation of the parameters using Eqs. (7)-(11) with $\mathbf{x}^p = \mathbf{x}_{old}^p$
3. Find \mathbf{x}_{new}^p , the new image patch estimate using Eq. (13)

While $\frac{\|\mathbf{x}_{new}^p - \mathbf{x}_{old}^p\|^2}{\|\mathbf{x}_{old}^p\|^2} > threshold$.

Set $\mathbf{x}_1^p = \mathbf{x}_{new}^p$, the deconvolved patch.

Obtain \mathbf{x}_2^p using the denoising algorithm in [12] with noise variance $(\beta_2^p)^{-1}$

Compute the error measures for \mathbf{x}_1^p and \mathbf{x}_2^p using Eq.(14)

if ($error_{12} > errThreshold$ AND $error_{21} > errThreshold$) OR $error_{11} > error_{12}$ then

Set $\mathbf{x}^p = \mathbf{x}_2^p$

else

Set $\mathbf{x}^p = \mathbf{x}_1^p$

function for each patch, win^p , as follows

$$\begin{aligned} \mathbf{x} &= \sum_p^P win^p \mathbf{x}^p \\ \sum_p^P win_l^p &= 1 \text{ for } 1 \leq l \leq (N_x \times N_y), \end{aligned} \quad (15)$$

with P the number of patches in the image.

We tried various windowing functions for win^p (Gaussian, rectangular and Hann), and found that a Hann window function over each patch, with a normalization to sum to one as proposed in [6] and described in Eq. (15) provided the best results.

3. EXPERIMENTS

In this section we report experiments obtained using the real image pairs acquired in dim environments with two different digital cameras: an Olympus C5060WZ is used in the first experiment, while the images in the second and third experiments were captured with a Canon EOS 400D. Results are shown in Figs. 1 and 2. The short-exposure observations have been photometrically and geometrically registered as explained in section 2.1 before displaying which explains the high level of noise and the blocking artifacts in the images. We have experimented with different blur kernel sizes and number of images patches depending on the input images. In all cases, we use an overlap percentage of 50% and a Hann window is used for the weights. The *threshold* and *errThreshold* in Alg. 2 are set to 10^{-4} and $(error_{11} + error_{22})$, respectively.

In the first experiment we worked with the image pair depicted in the first row of Fig. 1 which presents a severe motion blur and high level of noise. The exposure time for the blurred image is 1.3s with ISO 100 and 1/100s, ISO 400 for the dim image. The number of blocks is set to 6×6 , with a kernel size of 47×47 . The image in the second column of Fig. 2, represents the strategy chosen: patches in white represent a denoising restoration, for the rest we used our deconvolution approach. The proposed method successfully deconvolved and removed the noise of the input images. Note that small details such as the letters in the cup or in the book behind the vase are now visible.

The second experiment shows the behavior of our algorithm for an object motion scene together with a small camera shake. The

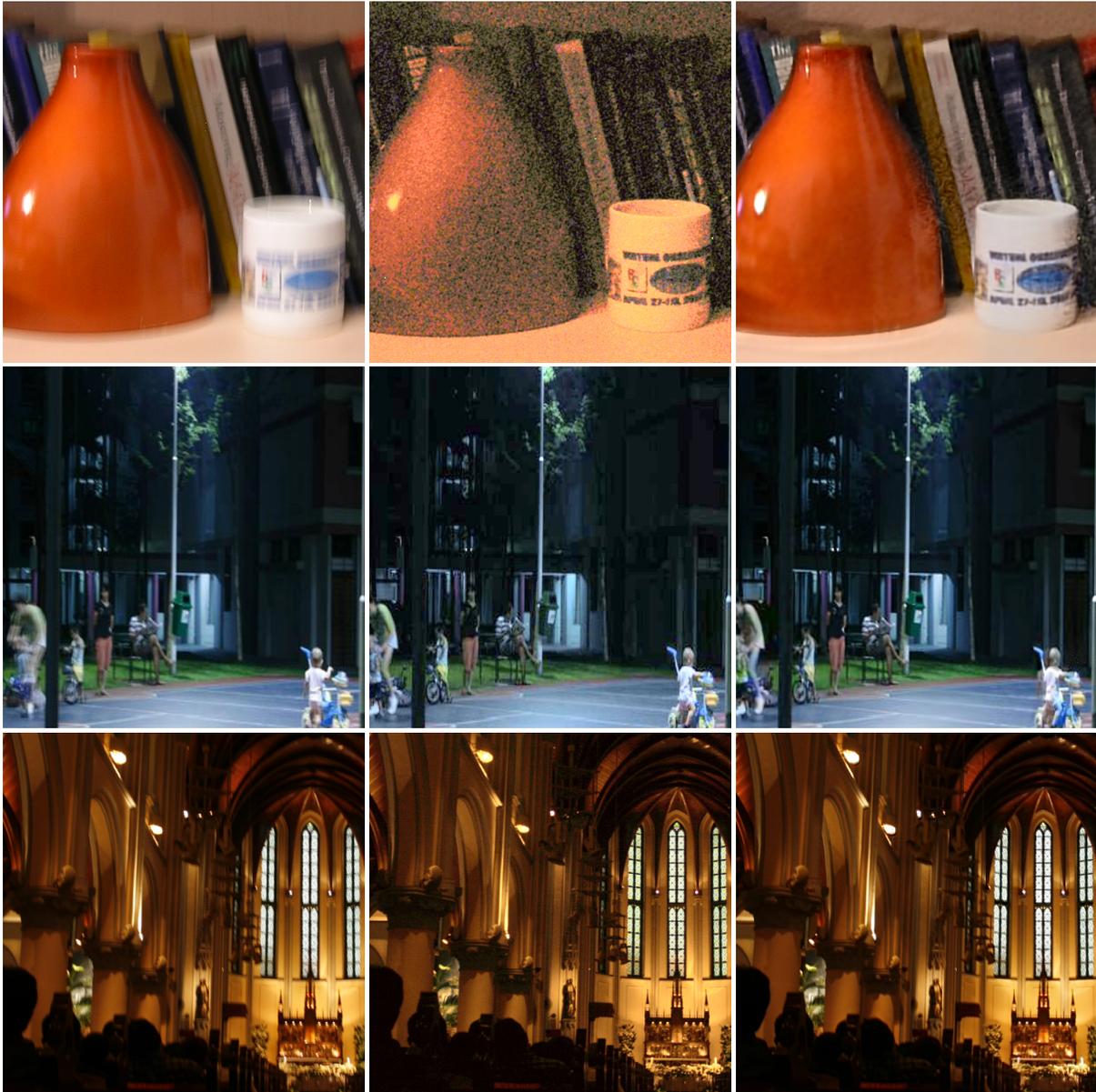


Figure 1: Real observed images and results of our algorithm. From left to right: observed long- and short-exposure image pair and final restoration.

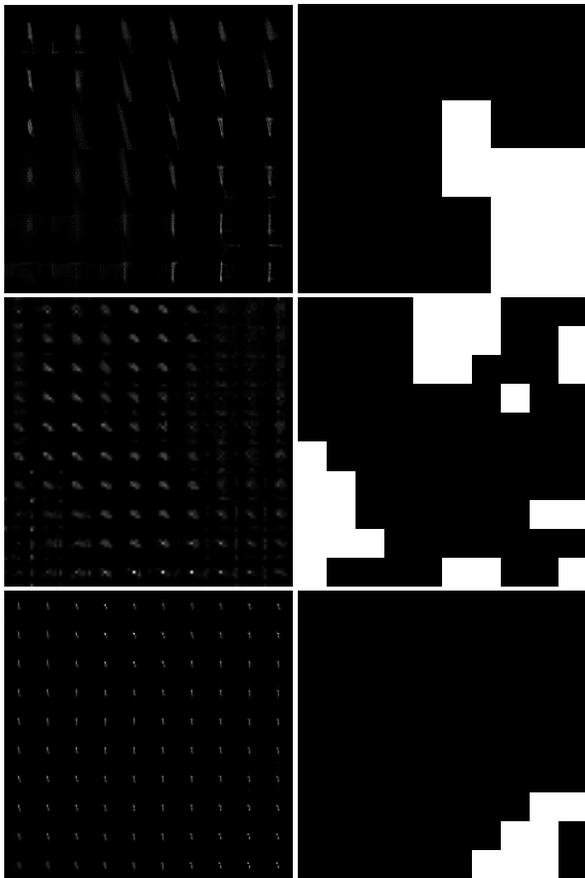


Figure 2: Results of the experiments. Each row corresponds to the results obtained for each experiment in Fig. 1. The column on the left shows the obtained kernel estimation. The column on the right depicts the result of the object motion correction. The white patches represents patches where denoising is applied.

exposure time in the blurred image is $1/4s$ in the absence of compensation (EV 0) and $1/15s$ with exposure compensation (EV -2) for the dim image, both shots with ISO 400. Since the blur is not so severe, we used a kernel size of 11×11 pixels and increased the number of blocks to 10×10 . Although the presence of noise in the under-exposed image is lower than in the previous case, this experiment illustrates how our algorithm detects patches where the kernel is not accurately estimated, based on the error measures detailed on section 2.4.1, due to objects motion. Note that the cyclist and the child, both moving, are detected (see Fig. 1, 2nd row, last column), and replaced by the denoised, sharp versions. Flat dark patches, where no information for blur estimation is available are also selected from the denoised version while the rest of the image is successfully deconvolved.

The last experiment illustrates the behavior of our algorithm in an extremely dim indoor scene, which mixes again both, motion and camera shake blur. The exposure time in the blurred image is $1/8s$ in the absence of compensation (EV 0) and $1/30s$ with exposure compensation (EV -2) for the dim image, both shots with ISO 400. Although the blur is not severe, the estimated kernel, depicted in the last row of Fig. 2, shows that the slight camera rotation was successfully detected by the blur estimation process, and the restoration in the last row of Fig. 1 demonstrates the accuracy of the proposed method. Note that the blur was successfully removed from the image and no noise from the short-exposure image (depicted Fig. 1, last row, central image) was introduced in the restored image (see, for instance, the area above the columns on the left hand side of the image). In this experiment we utilized a kernel size of 21×21 , and

10×10 .

In all tested images the quality of the restored image is high, providing a significant noise reduction and ghost free restoration in all experiments.

4. CONCLUSIONS

In this paper we have proposed a method to restore blurred images taken in dim environment with the help of a short-exposure crisp but noisy image. The developed algorithm can be applied for both, camera shake and object motion blur by using a space variant kernel estimation and relying on the noisy observation. The method minimizes blur artifacts and noise propagation in the recovery process. It is important to note that except for the size of the blur kernels, number of patches and its percentage of overlap, all parameters are estimated automatically by the proposed method. Future work includes the use of dynamic patch size.

REFERENCES

- [1] L. Yuan, J. Sun, L. Quan, and H-Y. Shum, "Image deblurring with blurred/noisy image pairs," *ACM Trans. Graph.*, vol. 26, 2007.
- [2] S. D. Babacan, J. Wang, R. Molina, and A. K. Katsaggelos, "Bayesian blind deconvolution from differently exposed image pairs," *IEEE Trans. on Image Processing*, vol. 19, no. 11, pp. 2874–2888, 2010.
- [3] M. Tallón, J. Mateos, S. D. Babacan, R. Molina, and A. K. Katsaggelos, "Combining observation models in dual exposure problems using the Kullback-Leibler divergence," *European Signal Processing Conference (EUSIPCO 2010)*, pp. 323–327, 2010.
- [4] M. Sorel and F. Sroubek, "Space-variant deblurring using one blurred and one underexposed image," in *Proc. Int. Conf. on Imag. Proc (ICIP)*, 2009, pp. 157–160.
- [5] M. Tico, N. Gelfand, and K. Pulli, "Motion-blur-free exposure fusion," in *Proc. Int. Conf. on Imag. Proc (ICIP)*, 2010, pp. 3321–3324.
- [6] M. Hirsch, S. Sra, B. Scholkopf, and S. Harmeling, "Efficient filter flow for space-variant multiframe blind deconvolution," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 607–614.
- [7] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, "Surf: Speeded up robust features," in *ECCV*, 2006, pp. 404–417.
- [8] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [9] A. Bovik, Ed., *Handbook of Image Video Processing*, Elsevier, second edition, 2005, Total Variation Regularization, pages 192, 193.
- [10] H. W. Richardson, "Bayesian-based iterative method of image restoration," *Journal of the Optical Society of America*, vol. 62, no. 1, pp. 55–59, 1972.
- [11] L. B. Lucy, "An iterative technique for the rectification of observed distributions," *Astron. J.*, vol. 79, pp. 745+, 1974.
- [12] J. Portilla, V. Strela, M. J. Wainwright, and E. P. Simoncelli, "Image denoising using scale mixtures of Gaussians in the wavelet domain," *IEEE Trans. Image Processing*, vol. 12, no. 11, pp. 1338–1351, 2003.