PATCH-BASED BLIND DECONVOLUTION WITH PARAMETRIC INTERPOLATION OF CONVOLUTION KERNELS

Filip Šroubek, Michal Šorel, Irena Horáčková, Jan Flusser

UTIA, Academy of Sciences of CR
Pod Vodárenskou věží 4, Prague 8, 182 08, Czech Republic

(a) blurred input (b) our reconstruction (c) close-ups

Fig. 1. Space-variant deconvolution of photos blurred by real camera shake: (a) one input blurry image; (b) reconstruction using the proposed method with parametric blur interpolation; (c) close-ups: (left) input blurry image, (middle) reconstruction with naive intensity interpolation shows strong artifacts, (right) proposed method.

ABSTRACT

We propose a method for removal of space-variant blur from images predominantly degraded by camera shake without any knowledge of camera trajectory. Blurs are first estimated in a small number of image patches. We derive a novel parametric blur interpolation method and discuss conditions under which it can be used to exactly calculate blurs for every pixel position. Having this information, we restore the sharp image by a standard regularization technique. Performance of the proposed method is experimentally validated.

***Index Terms*—** blind deconvolution, space-variant convolution, interpolation

1. INTRODUCTION

Blur induced by camera motion is a frequent problem in photography occurring mainly in poor light conditions, when the exposure time increases. Image stabilization devices help to reduce motion blur of limited extent. For larger blur, deblurring the image offline using mathematical algorithms remains the only way to obtain a sharp image.

Homogenous blurring can be described by convolution with a point spread function (PSF). Unfortunately this is not the case for motion blur due to camera shake, especially if the focal length of the lens is short. The blur is typically different in different parts of the image [1]; see an example in Fig. 1(a). The cause of its spatial variance (SV) is not only camera motion but also other factors such as optical aberrations. In practice deblurring is even more complicated, since we usually have no or limited information about the blur. We face an extremely ill-posed blind deconvolution (BD) problem.

For certain types of camera motion, such as rotation, we can express the degradation operator as a linear combination of basis blurs (or images) and solve the blind problem in the space of the basis, which has much lower dimension than the original problem. Whyte *et al.* [2] considered rotations about three axes up to several degrees and described blurring using three basis vectors. For blind deconvolution, they used an algorithm analogous to [3] based on marginalization over the latent sharp image. Gupta *et al.* [4] adopted a similar approach, replacing rotations about x and y axes by translations. Such projections to a low-dimension subspace look promising but they are disguised parametric methods with their main limitation that they only work for a specific class of blurs (in this case constrained camera motion) and they completely ignore any additional blurs, such as optical aberrations, that typically appear in real cases.

Here we adopt a different approach, used e.g. in [5, 6], in which the blur operator is assumed to be locally space invari-

Thanks to GACR agency project GA13-29225S for funding.

ant and thus can be locally approximated by standard convolution. This way we can deal with more general SV blurs. A good approximation of the SV blur operator is achieved by estimating PSFs in a neighbourhood of every pixel, which is computationally demanding. To avoid computation in each pixel, methods [5–7] estimate PSFs in a subset of pixels and use linear interpolation to express the PSF in the rest of the image. Linear interpolation however does not describe well how the PSF changes with position. The PSFs must be therefore estimated quite densely.

In this paper, we propose a parametric interpolation method that accurately and quickly calculates PSFs in every pixel of the image from PSFs estimated in just a small number of image locations.

2. SPACE-VARIANT BLIND DECONVOLUTION

The blurred image g is modeled by a general linear operator H applied to the latent image u:

$$\begin{aligned} g(x, y) &= [Hu](x, y) \\ &= \int u(x-s, y-t) h(s, t, x-s, y-t) ds dt\,. \end{aligned} \quad (1)$$

The operator H is a generalization of standard convolution where h is now a function of four variables. We can think of this operation as a convolution with a SV PSF $h(s, t, x, y)$ that depends on the position (x, y) in the image. Standard convolution is a special case of (1), where $h(s, t, x, y) = h(s, t)$ for any (x, y).

We assume that SV PSF changes relatively slowly in the image space, which is typically the case for camera motion and optical aberrations. Then the blur can be considered locally constant and we can approximate it by convolution in every image patch. Let us divide the image g into overlapping patches denoted as g^p, where p is the patch index. The SV convolution model in (1) transforms into a set of convolutions

$$g^p = k^p * u^p\,, \quad (2)$$

where k^p is a convolution kernel that approximates $h(s, t, x, y)$ for (x, y) in the p-th patch. In every patch, we thus have a classical BD problem. We have several ways how to solve such problem. If multiple blurred images of the same scene are available we can apply stable multichannel methods proposed for example in [8]. A special case is dual exposure problem when we combine a long exposure blurry image with a short exposure noisy one [5, 6]. If only a single blurred image is used, we can apply single-channel BD methods proposed recently in [3, 9]. Solving BD in every patch would be extremely time consuming. Instead we propose to solve the SV BD problem in a more efficient way:

- Use BD methods on (2) to estimate PSFs in few patches distributed on a coarse grid.
- Apply our novel interpolation method (see Sec. 3) to generate PSFs on a dense grid.
- Generate SV blurring operator H from the interpolated PSFs and find the sharp image u by solving (1) using non-blind SV deconvolution method [10].

3. PARAMETRIC INTERPOLATION

A simple approach to interpolation as proposed in [5] is to take the estimated PSFs on a coarse grid and perform interpolation of PSF intensity values, i.e. for each pixel location we consider four closest PSFs on the coarse grid and run bilinear interpolation of their intensity values. This technique is illustrated in Fig. 2(a). The circled PSFs were estimated and the intermediate PSFs were interpolated. an advantage of intensity interpolation is that it can be efficiently implemented inside SV deconvolution [11]. However, the interpolated PSFs are often very different from the correct ones. Refer to the experiment in Fig. 4 and notice that intensity interpolation generates false PSFs that are not motion blurs. To solve this problem, we derive below a method (parametric interpolation), which interpolates camera motion SV PSFs very accurately.

Let us consider the image a camera captures during its exposure window. Light from a scene point $X_w = [x_w, y_w, z_w]^T$ projects on the image plane at a location $X = [x, y]^T$. Using homogeneous coordinates in the image plane $\bar{X} = [dX^T, d]$, the relation to X_w is given by

$$\bar{X} = K[RX_w + T]\,, \quad (3)$$

where R (3×3) and T (3×1) are the camera rotation matrix and translation vector, respectively. Upper triangular matrix K (3×3) is the camera intrinsic matrix. The third element d of the homogeneous coordinates corresponds to distance.

During the exposure window the camera position and orientation may change and therefore the extrinsic parameters R and T are function of time t. The projected point X moves along a curve parametrized by t, which we denote as $\mathcal{T}(X)$ and call it a point trace. It is important to draw a relation between this curve and the SV PSF h. The SV PSF $h(s, t, X)$ corresponds precisely to the rendered trace $\mathcal{T}(\bar{X})$. The trace $\mathcal{T}(\bar{X})$ is given by

$$\bar{X}(t) = K[R(t)K^{-1}\bar{X}_0 + \frac{1}{d_0}T(t)]\,, \quad (4)$$

where $\bar{X}_0 = [X^T, 1]^T = [x, y, 1]$ is the initial location of the point in the image plane using normalized homogenous coordinates and d_0 is the third element of the corresponding $\bar{X}$.

The following proposition expresses a trace as a combination of two other traces.

Proposition 3.1 *Let the distance d of all points from the camera be constant. Given traces $\mathcal{T}(\bar{A})$ and $\mathcal{T}(\bar{B})$ at positions A and B, a trace $\mathcal{T}(\bar{C})$ at a position $C = kA + (1-k)B$, $k \in \langle 0, 1\rangle$ is expressed as $\mathcal{T}(\bar{C}) = k\mathcal{T}(\bar{A}) + (1-k)\mathcal{T}(\bar{B})$.*

The proof is direct application of (4). This proposition shows that a linear interpolation of two traces in homogenous coordinates generates any trace at a position lying on a line connecting these two traces. Unlike simple intensity interpolation, which performs linear interpolation in the space of intensity values, we interpolate here in the space of coordinates. However, the drawback is that we must know the homogenous coordinates of the projected points and thus the distance of the point from the camera at every time t. The next corollary alleviates this shortcoming.

Corollary 3.2 *Let the camera motion be constrained to rotation along optical z axis and translation in x-y plane. Proposition 3.1 simplifies to $\mathcal{T}(C) = k\mathcal{T}(A) + (1-k)\mathcal{T}(B)$.*

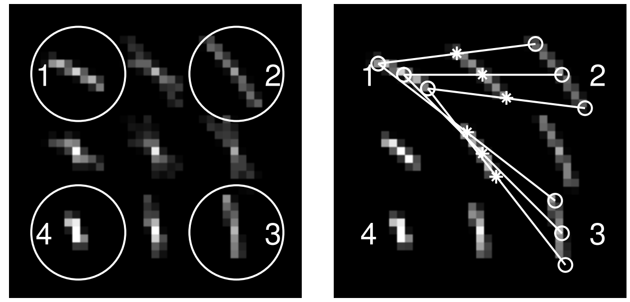
The proof is done by expressing the third element of homogenous coordinates and determining conditions under which it is equal to one. Using (4), the third element of $\bar{X}$ is

$$d = R_3(t)K^{-1}\bar{X}_0 + \frac{1}{d_0}T_3(t)\,, \tag{5}$$

where R_3 is the third row of R and T_3 is the third element of T. If the constrained camera motion of the corollary is assumed then $d = 1$. The above corollary shows that in the constrained camera motion the trace interpolation is independent of the scene distance and we can generate traces by working only with the projected points. Connecting pixels of a pair of traces (PSFs) that correspond to the same time t generates any PSF on the connected lines as illustrated in Fig. 2(b). Compare to intensity interpolation in Fig. 2(a), the PSFs are interpolated precisely.

If we lift the camera motion constraints given in Corollary 3.2 and assume an arbitrary camera motion, the parametric interpolation $\mathcal{T}(C) = k\mathcal{T}(A) + (1-k)\mathcal{T}(B)$ is inexact but the error is very small in most of the practical cases. Examining (5) reveals that the second term is negligible since scene distance from the camera (d) is typically much larger than the camera shift in the z optical direction (T_3). The interpolation error is thus predominantly produced by the first term, which is a function of rotation about x and y axes.

Fig. 3 plots the maximum interpolation error as a function of the number of known PSFs per row for rotation of up to $2°$ in x and y axes (10 Mpx camera, full-frame sensor 36×24mm, resolution 3600×2400 pixels). We assume that PSFs are correctly estimated at several locations equally spaced over the image plane and vary the density of estimation positions from 3 to 19 estimated PSFs per row. Then we calculate the maximum interpolation error among all neighbouring pairs of PSFs, which occurs close to image corners, where the rotational blur is the largest. Three curves correspond to three different focal lengths: $f = 20$mm (wide-angle lens), $f = 50$mm (normal lens), and $f = 200$mm (telephoto lens). Note that the PSF length for these three lenses generated by the camera rotation of $2°$ is roughly 200, 300, and 900 pixels, respectively, which can be regarded as extremely heavy blur.



(a) intensity interpolation (b) parametric interpolation

Fig. 2. PSFs at nine different positions for a combination of camera motion in the x-y plane and rotation around z axis: (a) illustrates five PSFs interpolated using simple linear intensity interpolation of four circled PSFs; (b) demonstrates parametric interpolation. For every time instance of the camera exposition we know the position in all four traces. By interpolating these corresponding positions we can generate any PSF in the image plane.

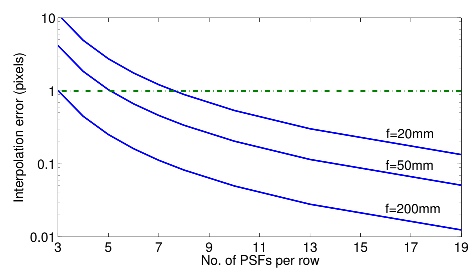


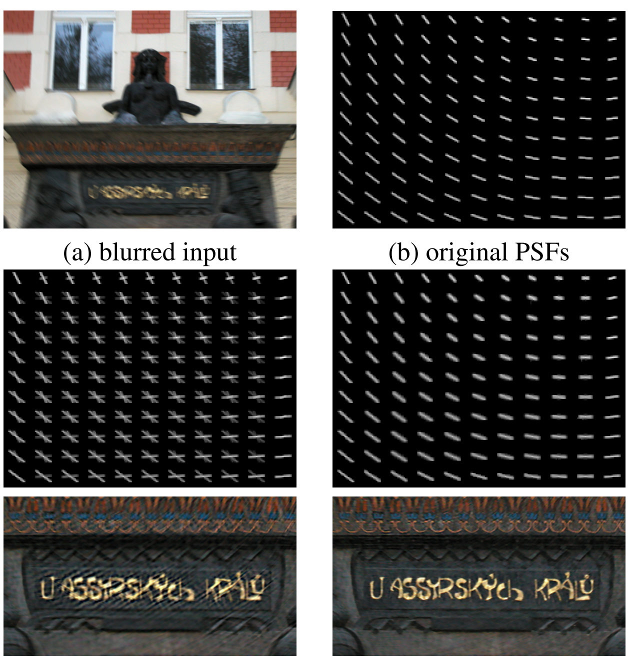
Fig. 3. Interpolation error for three different focal lengths ($f = 20, 50, 200$mm) as a function of estimated PSF density (from 3 to 19 PSFs per row) for the case of $2°$ rotation around the x and y axes.

Fig. 3 demonstrated that parametric interpolation easily achieves sub-pixel precision. The tricky part is how to derive the analytical form from the matrix representation we get as an output of BD methods. In other words, we need to track the curve and find a mapping between any two PSFs. The mapping matches pixels of the same time instance. This is possible, when the curve does not cross itself and has one prevailing orientation, which are assumptions satisfied in many practical cases, when the time of exposure is not too long. Let $a(x, y)$ and $b(x, y)$, $[x, y] \in [0 \ldots M, 0 \ldots N]$, denote two PSFs estimated by BD in patches located in A and B, respectively. For the parametric interpolation of two PSFs, the procedure we used can be outlined as follows:

- Bring both PSFs into normalized positions and denote them $a'(x, y)$ and $b'(x, y)$. We use principal axes normalization based on constraining second-order moments (see [12] for details). Hence, PSFs are oriented such that their principal axes coincide with the y-axis.
- Find mapping of rows y_A in $a'(x, y)$ to y_B in $b'(x, y)$ such that $\sum_{y=-\frac{N}{2}}^{y_A} \sum_{x=-\frac{M}{2}}^{\frac{M}{2}} a' = \sum_{y=-\frac{N}{2}}^{y_B} \sum_{x=-\frac{M}{2}}^{\frac{M}{2}} b'$. For each pair of mapped rows find mapping of x in a similar way, but in 1D. This way we get a mapping for every point $[x_B, y_B] = m(x_A, y_A)$.
- Bring the PSFs into their original position. Connect the matching points given by mapping m and find the pixel position of the interpolated PSF according to Corollary 3.2. The pixel intensity of the interpolated PSF is given by linear interpolation of intensities in $a(x, y)$ and $b(x, y)$.

In practice we need to interpolate in two dimensions (from four PSFs). This is solved by first interpolating in one dimension using two pairs of PSFs and finally interpolating two PSFs from the previous step in the other dimension.

4. EXPERIMENTS

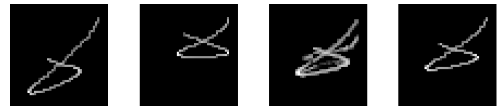


(a) blurred input (b) original PSFs

(c) intensity, SSIM=0.897 (d) parametric, SSIM=0.974

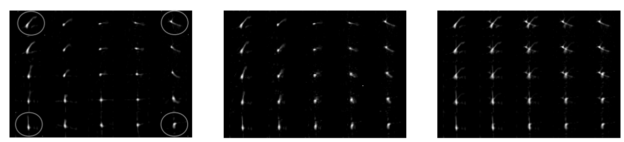
Fig. 4. Synthetic experiment of space-variant deconvolution

Fig. 4 illustrates advantages of parametric interpolation in SV deconvolution. A blurred image (a) was synthetically generated by modeling camera motion, which results in SV PSFs visualized on the 11×11 grid in (b). Using only four PSFs in the corners, we interpolated the remaining ones on the 11×11 grid using simple intensity interpolation (c) and proposed parametric interpolation (d). The similarity between the interpolated and true PSFs was assessed by the SSIM method [13] and is provided below figures. Intensity interpolation clearly generates PSFs that are incorrect and thus the reconstructed image exhibits strong artifacts as shown in the close-up in (c). Proposed interpolation generates PSFs similar to the true ones and the reconstructed image is almost perfect. If the PSFs have crossovers and/or no prevailing orientation, our implementation of the parametric interpolation often fails as illustrated in Fig. 5. This fault is not because of violation of theoretical assumptions but because our implementation works without the knowledge of trace parametrization.



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

Fig. 5. Parametric interpolation of PSFs (a) and (b) with crossovers generates false lines (c) compared to the correct PSF (d).



(a) (b) SSIM=0.992 (c) SSIM=0.967

Fig. 6. Comparison of interpolation methods on space-variant deconvolution in Fig. 1. SSIM compares to (a).

Fig. 1 shows an example of deblurring real photos. We took two pictures in a dark room blurred due to long exposure time. One of them is shown in Fig. 1(a). Next, we estimated 25 PSFs on the 5×5 grid (patch size 150×150) using the BD method in [8]. Each PSF took about 19s, adding up to $25 \times 19 = 475$s. The PSFs are plotted in Fig. 6(a). We also generated PSFs using just four estimated PSFs in the corners by simple intensity interpolation (Fig. 6(b)) and proposed parametric interpolation (Fig. 6(c)). Both interpolation methods consume negligible time and the entire process took in this case about $4 \times 19 = 76$s. This is 6-times less than the full estimation of 25 PSFs and the computational time decreases even more if a denser grid is used. Intensity interpolation generates incorrect PSFs whereas the proposed method returns PSFs similar to the estimated ones by BD. Finally, we deblurred the images, which took about 7s, using the SV method [10] and compare results achieved by both interpolation methods. Close-ups are shown in Fig. 1(c), linear interpolation in the middle and parametric interpolation on the right. The complete reconstructed image using parametric interpolation is in Fig. 1(b).

5. REFERENCES

[1] Michal Šorel and Jan Flusser, "Space-variant restoration of images degraded by camera motion blur," *IEEE Transactions on Image Processing*, vol. 17, no. 2, pp. 105–116, Feb. 2008.

[2] Oliver Whyte, Josef Sivic, Andrew Zisserman, and Jean Ponce, "Non-uniform deblurring for shaken images," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, June 2010, pp. 491 –498.

[3] Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T. Roweis, and William T. Freeman, "Removing camera shake from a single photograph," in *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, New York, NY, USA, 2006, pp. 787–794, ACM.

[4] Ankit Gupta, Neel Joshi, C. Lawrence Zitnick, Michael Cohen, and Brian Curless, "Single image deblurring using motion density functions," in *Proceedings of the 11th European conference on Computer vision: Part I*, Berlin, Heidelberg, 2010, ECCV'10, pp. 171–184, Springer-Verlag.

[5] Michal Šorel and Filip Šroubek, "Space-variant deblurring using one blurred and one underexposed image," in *Proc. IEEE International Conference on Image Processing*, 2009.

[6] Miguel Tallón, Javier Mateos, S. Derin Babacan, Rafael Molina, and Aggelos K. Katsaggelos, "Space-variant blur deconvolution and denoising in the dual exposure problem," *Information Fusion*, , no. 0, pp. –, 2012.

[7] Stefan Harmeling, Hirsch Michael, and Bernhard Scholkopf, "Space-variant single-image blind deconvolution for removing camera shake," in *Advances in Neural Information Processing Systems 23*, J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R.S. Zemel, and A. Culotta, Eds., 2010, pp. 829–837.

[8] Filip Šroubek and Jan Flusser, "Multichannel blind deconvolution of spatially misaligned images," *IEEE Trans. Image Processing*, vol. 14, no. 7, pp. 874–883, July 2005.

[9] Li Xu and Jiaya Jia, "Two-phase kernel estimation for robust motion deblurring," in *Proceedings of the 11th European conference on Computer vision: Part I*, Berlin, Heidelberg, 2010, ECCV'10, pp. 157–170, Springer-Verlag.

[10] Stephen J. Olivas, Michal Šorel, and Joseph E. Ford, "Platform motion blur image restoration system," *Appl. Opt.*, vol. 51, no. 34, pp. 8246–8256, Dec 2012.

[11] James G. Nagy and Dianne P. O'Leary, "Restoring images degraded by spatially variant blur," *SIAM JOURNAL ON SCIENTIFIC COMPUTING*, vol. 19, no. 4, pp. 1063–1082, 1998.

[12] Jan Flusser, Tomáš Suk, and Barbara Zitová, *Moments and Moment Invariants in Pattern Recognition*, Wiley, 2009.

[13] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.