

Space-Variant Restoration of Images Degraded by Camera Motion Blur

Michal Šorel and Jan Flusser, *Senior Member, IEEE*

Abstract—We examine the problem of restoration from multiple images degraded by camera motion blur. We consider scenes with significant depth variations resulting in space-variant blur. The proposed algorithm can be applied if the camera moves along an arbitrary curve parallel to the image plane, without any rotations. The knowledge of camera trajectory and camera parameters is not necessary. At the input, the user selects a region where depth variations are negligible. The algorithm belongs to the group of variational methods that estimate simultaneously a sharp image and a depth map, based on the minimization of a cost functional. To initialize the minimization, it uses an auxiliary window-based depth estimation algorithm. Feasibility of the algorithm is demonstrated by three experiments with real images.

Index Terms—Camera shake, deblurring, depth from motion, image stabilization, multichannel (MC) blind deconvolution, passive ranging, point spread function (PSF), regularization, shift-variant, space-variant restoration.

I. INTRODUCTION

SUBJECT to physical and technical limitations, the output of digital cameras is not perfect and a substantial part of image processing research focuses on removing various types of degradations.

One of the frequent degradations is the blur caused by camera motion, which can be described by the linear relation

$$\mathbf{z}(x, y) = \int \mathbf{u}(x - s, y - t) \mathbf{h}(x - s, y - t; s, t) ds dt \quad (1)$$

where \mathbf{u} is an original image, \mathbf{h} is called *point-spread function* (PSF) or *mask*, and \mathbf{z} is the blurred image. If the PSF does not depend on the position (x, y) in the image, i.e., $\mathbf{h}(x, y; s, t) = \mathbf{h}(s, t)$, the integral becomes convolution and we speak about *space-invariant* PSF. In this situation the discrete representation of \mathbf{h} by a matrix is called *convolution mask* or simply *mask*. We will use this term in the general space-variant case, as well in the sense that the mask is considered for each image pixel separately.

If we assume a planar scene perpendicular to the optical axis and steady motion of the pinhole camera¹ in a plane parallel

Manuscript received February 25, 2007; revised October 20, 2007. This work was supported by the Czech Ministry of Education, Youth, and Sports under the project 1M0572 (Research Center DAR). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Stanley J. Reeves.

The authors are with the Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, 18208 Prague 8, Czech Republic (e-mail: sorel@utia.cas.cz; flusser@utia.cas.cz).

Digital Object Identifier 10.1109/TIP.2007.912928

¹Pinhole camera is an approximation that assumes infinitely small aperture and neglects diffraction effects. This model is often used in computer vision.

to the scene, it is well known that the PSF is a space-invariant 1-D rectangular impulse in the direction of camera motion. In general cases, the PSF can be very complex depending on the camera motion, depth of scene and parameters of the optical system.

The important task to find the original image \mathbf{u} when we know the blurred image \mathbf{z} and possibly the PSF \mathbf{h} is called *restoration*, *deblurring* or, if \mathbf{h} is space-invariant, *deconvolution*. If even the PSF is not known, we speak about *blind* restoration or deconvolution. Blind restoration from only one image is an ill-posed problem. However, if we have at least two observations of the same scene, it gives us additional information that makes this problem tractable. This situation is denoted as *multichannel* (MC) restoration or deconvolution.

In this paper, we solve the problem of multichannel restoration from images blurred by camera motion. We consider the special case where the camera moves in only one plane parallel to the image sensor and does not rotate.

In the rest of this section, we give a survey of relevant literature and at the end we explain the idea behind so called variational methods we use in our algorithm.

There are many methods for the restoration of a single image degraded by known space-invariant blur, so called space-invariant *single channel (SC) nonblind* restoration techniques [1]. Many of them are formulated as linear problems that can be efficiently solved by elementary numerical algorithms, some others including important *anisotropic regularization* techniques [2], [3] can be reduced to a sequence of linear problems. Extension of these methods to a multichannel case is straightforward and many of them can be used in space-variant situations as well. An application of nonblind restoration in conjunction with the extraction of so called optical flow for motion deblurring can be found in [4].

Blind restoration requires more complicated algorithms as we need to estimate the unknown degradation.

Although a number of SC blind deconvolution algorithms were proposed [5], [6], their use is very limited even in space-invariant case because of a severe lack of information contained in just one image. Astronomy is a typical application area of these methods. Recently, a promising approach appeared employing statistics of the distribution of gradients in natural images [7]. In the MC blind space-invariant case, i.e., when we know two or more degraded images and the degradation does not change throughout the image, much more information is available. There are a number of methods successfully solving this issue [8]–[10].

If there are no constraints on the shape of the PSF and the way it can change throughout the image (general space-variant

blind restoration), the task is strongly underdetermined. A few results on this subject reported in literature followed the idea of sliding-window—PSF must be approximately space-invariant in a window of reasonable size and the result of identification is used as a starting point for the identification in subsequent windows [11]–[13].

In the case of space-variant motion blur, we know more about the shape of the PSF and so the number of unknowns can be reduced. The vast majority of algorithms still assumes that the PSF is space-invariant in a window of reasonable size [14]–[16]. This assumption was also used in algorithms for optical flow estimation from motion blur, for example, [17] and [18].

If we consider scenes with significant depth variations and space-variant blur caused by camera motion, these methods are not suitable as the condition of space-invariance is not satisfied, especially at the edges of objects. For this case, so far, the only approach that seems to give relatively precise results are multichannel variational methods that first appeared in the context of out-of-focus images in [19]. This approach was adopted by Favaro *et al.* [20], [21] who modeled camera motion blur by Gaussian PSF, locally deformed according to the direction and extent of blur. This model can be appropriate for small blurs. Relevant papers using variational techniques can be found also in the context of optical flow estimation, such as [22].

The idea behind variational methods [19], [20] is as follows. We know how to describe mathematically (that is how to simulate) the process of blurring according to (1). In this formula, the blur is characterized by a PSF that depends on camera settings, depth map of scene and camera motion and we assume that this relation as well as camera motion and camera settings are known. The algorithm is looking for such a sharp image and depth map that after the blurring of the image using the depth map, give images as similar as possible to the blurred images at the input of the algorithm. The “similarity” of images is expressed by a functional that should achieve as small a value as possible. Thus, the solution of the problem is equivalent to the minimization of the functional.

Unfortunately, it is by no means easy to apply this idea in practice. In the case of camera motion blur, we must solve the following issues.

- 1) How to express the PSF as a function of depth. For this purpose, we need either to get a description of camera motion and camera settings at the input or to be able to recover both from the blurred images themselves
- 2) Design an efficient algorithm to minimize nonconvex functional we derive
- 3) We need a reasonable initial estimate of the solution to prevent the algorithm from getting trapped in a local minimum

It turns out that it is especially difficult in the case of a more general camera motion. This is probably the reason why there are so few papers considering camera motion blur in this framework. In [20], the authors consider only very simple case of linear translation. In addition, this translation must be very short so that the true rectangular impulse could be modeled by the Gaussian function and the minimization does not get trapped in a local minimum. The use of the Gaussian is indispensable for their minimization method and we cannot see any straightforward way to extend it to the situations where the PSF cannot be

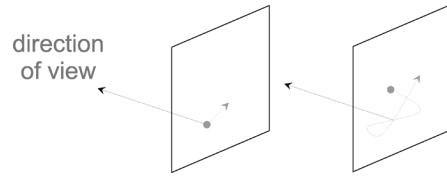


Fig. 1. Unlike existing methods (left), in the proposed algorithm, the camera can move along an arbitrary curve in one plane perpendicular to the optical axis (right).

modeled by the Gaussian; [21] is an extension of [20] to segmentation of moving objects but it keeps the limitations of the original paper concerning the shape of the PSF.

Our motivation was to enlarge the class of camera motions where the variational approach can be used. The results are outlined in Section II (Contributions).

Next, Section III explains how we model camera motion blur. Section IV introduces notation used for two important linear operations. Then, in several sections, we proceed to the description of the algorithm, including comments upon the practical issues associated with its implementation. In Section VI, we discuss the extension of the algorithm to general camera motion. Finally, we present three experiments with real images.

II. CONTRIBUTIONS

In this paper, we present a novel algorithm for restoration from multiple images blurred by camera motion.

We consider the special case when the camera moves along an arbitrary curve lying in one plane parallel to the image sensor without any rotations (Fig. 1). This plane must remain the same for all the input images, which ensures that also the depth map is common for all of them. On the other hand, the algorithm is able to deal with a change of camera position between images (unlike [20]).

We do not consider depth of field effects explicitly. Nevertheless, if the camera settings are the same for all the images, we can consider depth-of-field effects as part of the “original image” u . Then our algorithm removes the motion blur and preserves the depth of field. The only condition is that there exist a flat part of scene in focus (See Section V-A). We also neglect occlusions at depth discontinuities originating in nonzero (finite) aperture.

At the input, the algorithm needs the blurred images and selection of a window of approximately constant depth by user input. On the other hand, it needs to know neither how the camera moves nor camera parameters.

The proposed algorithm belongs to the group of variational methods, the idea of which was indicated at the end of the previous section. The algorithm does not rely on any parametric shape of the PSF and works for motions along an arbitrary curve parallel to the image sensor (figure-eight and V-shaped trajectories can be seen in the second and third experiments). This is a significant improvement over previous methods described at the end of the introduction.

To estimate the camera trajectory that corresponds to the shape of PSF, we apply the blind deconvolution algorithm [10] (Section V-A). We also address the problem that the functional

we minimize has many local minima and no gradient based algorithm can guarantee global convergence on its own. To solve this issue, we propose the use of an auxiliary algorithm described in Section V-A to get a reasonable initial estimate of the depth map.

The behavior of the algorithm is demonstrated by three experiments with real images.

III. MODELING OF CAMERA MOTION BLUR

In the proposed algorithm, we use the pinhole camera model mentioned in the introduction.

To model camera motion blur according to (1), we need to express the PSF as a function of camera motion and depth of scene. In case of general camera motion, it can be computed from the formula for *velocity field* [23], [20] that gives apparent velocity of the scene for the point (x, y) of the image at time instant τ as

$$\mathbf{v}(x, y, \tau) = \frac{1}{\mathbf{d}(x, y, \tau)} \begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \end{bmatrix} T(\tau) + \begin{bmatrix} xy & -1 - x^2 & y \\ 1 + y^2 & -xy & -x \end{bmatrix} \Omega(\tau) \quad (2)$$

where $\mathbf{d}(x, y, \tau)$ is the depth corresponding to point (x, y) and $\Omega(\tau)$ and $T(\tau) = [T_x(\tau), T_y(\tau), T_z(\tau)]^T$ are 3-D vectors of rotational and translational velocities of the camera at time τ . Both vectors are expressed with respect to the coordinate system originating in the optical center of the camera with axes parallel to x and y axes of the sensor and to the optical axis. All the quantities, except $\Omega(\tau)$, are in focal length units. The *depth* $\mathbf{d}(x, y, \tau)$ is measured along the optical axis, the third axis of the coordinate system. The function \mathbf{d} is called *depth map*.

The apparent curve $[\bar{x}(x, y, \tau), \bar{y}(x, y, \tau)]$ drawn by the given point (x, y) can be computed by the integration of the velocity field over the time when the shutter is open. Having the curves for all the points in the image, the 2-D space-variant PSF can be expressed as

$$\mathbf{h}(x, y; s, t) = \int \delta(s - \bar{x}(x, y, \tau), t - \bar{y}(x, y, \tau)) d\tau \quad (3)$$

where δ is the 2-D Dirac delta function.

We will show that in the considered special case, the PSF can be expressed explicitly using the knowledge of the PSF for one fixed depth of scene.

If the camera does not rotate, that is $\Omega = [0, 0, 0]^T$, and moves in only one plane perpendicular to the optical axis ($T_z(\tau) = 0$), (2) becomes

$$\mathbf{v}(x, y, \tau) = \frac{1}{\mathbf{d}(x, y, \tau)} \begin{bmatrix} -T_x(\tau) \\ -T_y(\tau) \end{bmatrix}. \quad (4)$$

In other words, the velocity field has the direction opposite to camera velocity vector and the magnitudes of velocity vectors are proportional to inverse depth. Moreover, depth for the given part of the scene does not change during such a motion (depth is measured along the optical axis and the camera moves perpendicularly to it), $\mathbf{d}(x, y, \tau)$ is a constant, and consequently the PSF simply follows the (mirrored because of the minus sign) curve drawn by the camera in image plane. The curve only changes its scale proportionally to the inverse depth.

The same is true for the corresponding PSFs we get according to relation (3). Let us denote the PSF corresponding to an object of the depth equal to the focal length as \mathbf{h}_0 . Note that this ‘‘prototype’’ PSF also corresponds to the path covered by the camera. Recall that the depth is given in focal length units. After linear substitution in the integral (3), we get

$$\mathbf{h}(x, y; s, t) = \mathbf{d}^2(x, y) \mathbf{h}_0(\mathbf{sd}(x, y), t\mathbf{d}(x, y)). \quad (5)$$

Equation (5) implies that, if we recover the PSF for an arbitrary fixed depth, we can compute it for any other depth by simple stretching in the ratio of the depths.

IV. NOTATION

In the case of the general space-variant linear degradation according to (1), we can look at the involved linear operation as convolution with PSF that changes with its position in the image and speak about *space-variant convolution*. For this operation, we introduce notation

$$\mathbf{u} *_v \mathbf{h}[x, y] = \int \mathbf{u}(x - s, y - t) \mathbf{h}(x - s, y - t; s, t) ds dt. \quad (6)$$

Note that we use subscript v to distinguish from ordinary space-invariant convolution often denoted by asterisk.

We will also need the operator adjoint to space-variant convolution that can be written as

$$\mathbf{u} \circledast_v \mathbf{h}[x, y] = \int \mathbf{u}(x - s, y - t) \mathbf{h}(x, y; -s, -t) ds dt. \quad (7)$$

V. ALGORITHM

In this and the following sections, we detail the proposed algorithm.

At the input, the algorithm requires multiple blurred images and specification of a region of approximately constant depth by user input. The user must also specify an upper-bound for the size of the PSF. The algorithm works in three phases.

- 1) PSF estimation at fixed depth using the blind deconvolution algorithm [10] (Section V-A).
- 2) Depth map estimation using the method described in Section V-B.
- 3) Variational minimization to get the sharp image and a more precise depth map (Sections V-C to V-G).

Recall that the camera does not rotate and moves in only one plane parallel to the image sensor. This plane is common for all the images, which ensures that the depth map is common, as well. We need to know neither how the camera moves nor camera parameters.

A. PSF Estimation at Fixed Depth

In the first phase of the proposed algorithm, the user chooses a section of the image where depth variations are negligible. It implies that the blur is approximately space-invariant within this section and can be modeled by convolution. We apply the blind deconvolution method [10] to estimate the PSF valid for the depth corresponding to this part of the scene. This method requires an upper-bound for the size of the PSF at the input.

Let us denote this depth as d_0 . Since we usually do not know its real value, the depth map estimate we are getting in the fol-

lowing phases of the algorithm is relative, that is correct up to a scale factor. This *relative inverse depth map* will be denoted as \mathbf{w} , with values

$$\mathbf{w}(x, y) = \frac{d_0}{\mathbf{d}(x, y)}.$$

In other words, its values give the ratio of the depth of the section the user chose and the true depth for the point (x, y) . We will see later that for restoration purposes the relative depth map is sufficient.

Suppose for the moment that we know a correct depth map \mathbf{w} . We can use the relation between PSFs for different depths (5) and get the space-variant PSF for the whole image. We denote this by the operator $h_p(\mathbf{w})$, where p is an image number. Thus, using notation (6), $\mathbf{u} *_v h_p(\mathbf{w})$ simulates the blurring of the image p .

To conclude the description of the first phase of the algorithm, we should remark that for numerical reasons, the user should choose a section that is as close to the camera as possible. Otherwise, the matrix representing PSF must be enlarged by a large factor for close objects and we loose precision. Another important consideration is that this section should be in focus, that is blurred only by camera motion. If it is simultaneously out-of-focus, the algorithm [10] does not work properly. The reason is that the PSFs have a common factor with respect to convolution and the deconvolution is ambiguous. Consequently, we are not able to separate motion and out-of-focus blur, which is crucial for our algorithm. On the other hand, the scene can have a small depth of field as illustrated in the third experiment. The only condition is that there must be a suitable flat part of the scene which is in focus.

B. Window-Based Estimation of the Depth Map

In the second phase of the algorithm, we compute an initial rough estimate of the depth map using a simple method based on the assumption that blur is space-invariant in a neighborhood of each pixel. In other words, the scene is assumed to be piecewise planar. The method we used can be described, as follows.

Let us denote the blurred images at the input as \mathbf{z}_p . Suppose we are able to compute $h_1(w)$ and $h_2(w)$ for an arbitrary inverse depth w . Now we take w as scalar value and h_i are space-invariant PSFs. Then, we compute

$$[\mathbf{z}_1 * h_2(w) - \mathbf{z}_2 * h_1(w)]^2 \quad (8)$$

for a sequence of values w covering the interval of possible inverse depths. Experiments have shown that it is sufficient to take the step corresponding to change in the support of the PSF of about 0.1 pixel. Obviously, if there is no noise and w is correct, the value of (8) should be zero because $\mathbf{z}_1 = \mathbf{u} * h_1(w)$, $\mathbf{z}_2 = \mathbf{u} * h_2(w)$ and convolution is commutative. In practice, for every pixel, the algorithm simply takes w minimizing the average value of (8) over some finite window. Note that the size of the window must be significantly larger than considered maximal support of PSF. In addition, if we have an estimate of noise levels σ_1 and σ_2 in images \mathbf{z}_1 and \mathbf{z}_2 , it proved beneficial to subtract $\sigma_2^2 \|h_1(w)\|^2 + \sigma_1^2 \|h_2(w)\|^2$ from (8), to compensate for the bias produced by the noise. Details can be found in [24].

We can ask whether it is possible to use the depth map we have got directly for restoration. The answer is that in most situations it results in unacceptable artifacts, as shown later in our experiments [Fig. 4(a)]. Nevertheless, this depth map can be used as a reasonable initial position for the minimization described in the following sections.

C. Variational Minimization

Recall that the process of blurring can be modeled using space-variant convolution, which can be written in simplified manner as (6). Then the third phase of the proposed algorithm can be described as minimization of cost functional

$$E(\mathbf{u}, \mathbf{w}) = \frac{1}{2} \sum_{p=1}^P \|\mathbf{u} *_v h_p(\mathbf{w}) - \mathbf{z}_p\|^2 + \lambda_u Q(\mathbf{u}) + \lambda_w R(\mathbf{w}) \quad (9)$$

with respect to sharp image \mathbf{u} and the relative inverse depth map \mathbf{w} defined in Section V-A. As we already mentioned, the depth map is the same for all the images, because of the constraints placed on the camera motion.

The first term of (9), called *error term* in the rest of this paper, is a measure of difference between the inputs, i.e., blurred images \mathbf{z}_p , and the image \mathbf{u} blurred according to blurring model using information about depth of scene \mathbf{w} . The size of the difference is measured by L_2 norm $\|\cdot\|$. The inner part of the error term

$$\mathbf{e}_p = \mathbf{u} *_v h_p(\mathbf{w}) - \mathbf{z}_p \quad (10)$$

is nothing else than the matrix of error at the individual points of the image p .

For image p , operator $h_p(\mathbf{w})$ gives the space-variant PSF corresponding to camera motion and to the depth map represented by \mathbf{w} as described in Section V-A. Note that h_p incorporates a possible shift of the camera between the images.

Regularization is a standard method to achieve satisfactory solution of problems involving inversion of ill-conditioned operators such as the convolution with space-variant PSF. The role of regularization terms is to make the problem well-posed and incorporate prior knowledge about the solution [25].

Thus, $Q(\mathbf{u})$ is an image regularization term which can be chosen to properly represent the expected character of the image function. For the majority of images a good choice is total variation $(TV) Q_{TV}(\mathbf{u}) = \int \|\nabla \mathbf{u}\|$, where $\nabla \mathbf{u}$ denotes the gradient of \mathbf{u} . The size of the gradient is integrated over the whole area of the image. Very good anisotropic denoising properties of the total variation were shown by Rudin *et al.* [2]. A reason why TV works so well for real images is that the total variation favors piecewise constant functions and in real images object edges create sharp steps that appear as discontinuities in the intensity function.

Tikhonov regularization term $Q_2(\mathbf{u}) = \int \|\nabla \mathbf{u}\|^2$ can be more appropriate for scenes without sharp edges. In turn, an issue with Tikhonov regularization is that it tends to oversmooth sharp edges because of the square of the image gradient that penalizes too much the gradients corresponding to edges. For more detailed discussion of image regularization, see [1], [26], and [27].


 (a) 870×580 image blurred by periodic horizontal motion

 (b) 870×580 image blurred by periodic vertical motion

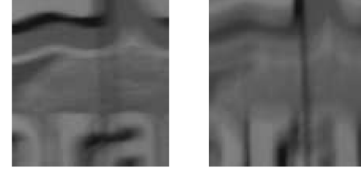

(c) ground truth image

Fig. 2. We took two images from the camera mounted on device vibrating in (a) horizontal and (b) vertical directions. For both images, the shutter speed was set to 5 s and aperture to $F/16$. For comparison, the third image was taken without vibrations serving as a “ground truth.”

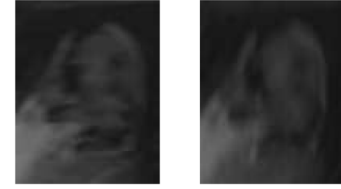
Similarly, we can choose a convenient depth map regularization term $R(\mathbf{w})$ according to the character of the depth map. Again TV and Tikhonov regularization are good candidates.

D. Gradient of the Cost Functional

In theory, to minimize the cost functional (9), we could apply simulated annealing [19], which guarantees global convergence. In practice, however, it would be prohibitively slow. For efficient minimization, we need to know at least the gradient (Fréchet



(a) sections of images Fig. 2(a) and (b) used for the estimate of PSFs were taken from areas at the juice box on the right (50×54 pixels, $5\times$ enlarged)


 (b) 11×11 PSFs computed from images (a)


(c) another section from the proximity of image center used for computation of PSFs (d) (46×59 pixels, $5\times$ enlarged)


 (d) 11×11 PSFs computed from images (c)

Fig. 3. Algorithm needs an estimate of PSFs for at least one distance from camera. For this purpose, we cropped a section from the right part of images Fig. 2(a) and (b), where the distance from camera was constant and computed PSFs (b) using blind space-invariant restoration method [10]. For comparison, we computed (d) PSFs from (c) sections taken from the image center. We can see that, in agreement with our model, (d) the PSFs are a scaled down version of (b) PSFs.

derivative) of the functional. It equals the sum of the gradients of individual terms. First, we cover the gradients of the regularization terms.

The gradient of any functional of form $\int \kappa(\|\nabla \mathbf{u}\|)$, where κ is an increasing smooth function, can be expressed [28] as

$$-\text{div} \left(\frac{\kappa'(\|\nabla \mathbf{u}\|)}{\|\nabla \mathbf{u}\|} \nabla \mathbf{u} \right). \quad (11)$$

In this equation, the function κ' is the derivative of κ . For Q_2 and Q_{TV} , we get

$$\frac{\partial Q_2}{\partial \mathbf{u}} = -\text{div} \nabla \mathbf{u} = -\nabla^2 \mathbf{u} \quad (12)$$

$$\frac{\partial Q_{TV}}{\partial \mathbf{u}} = -\text{div} \left(\frac{\nabla \mathbf{u}}{\|\nabla \mathbf{u}\|} \right) \quad (13)$$

where the symbol ∇^2 denotes Laplacian operator and div the divergence operator. The gradient of $R(\mathbf{w})$ we get by simply replacing \mathbf{u} with \mathbf{w} in (11)–(13).

To express gradients of the error term, now denoted as Φ , we take advantage of the notation (7). The gradients of the error



(a) direct restoration using depth map (b), TV image regularization, $\lambda_u = 10^{-4}$



(b) depth map got by auxiliary algorithm (8), error averaged by 7×7 window, depth map finally smoothed by 23×23 median filter

Fig. 4. Illustration of the fact that we cannot use simple depth recovery methods directly for restoration. We can see many visible artifacts in all parts of the image despite of the heavy smoothing of the depth map.

term Φ in subspaces corresponding to image \mathbf{u} and depth map represented by \mathbf{w} can be expressed as

$$\frac{\partial \Phi}{\partial \mathbf{u}} = \sum_{p=1}^P \mathbf{e}_p \otimes_v h_p(\mathbf{w}) \quad (14)$$

$$\frac{\partial \Phi}{\partial \mathbf{w}} = \mathbf{u} \sum_{p=1}^P \mathbf{e}_p \otimes_v \frac{\partial h_p(\mathbf{w})}{\partial \mathbf{w}} \quad (15)$$

where $(\partial h_p(\mathbf{w})/\partial \mathbf{w})[x, y, s, t]$ is the derivative of the PSF related to image point (x, y) with respect to the value of $\mathbf{w}(x, y)$. The proof of these formulas can be found in [24].

Notice that the computation of gradients (14) and (15) does not take much longer than the computation of the cost functional itself. They consist only of point-wise multiplication, point-wise subtraction and two types of linear matrix operations (6) and (7). These two linear operations themselves consist only of scalar products of vectors. All these operations can be highly parallelized since basically the value can be computed in each pixel separately.

We should mention the actual implementation of $h_p(\mathbf{w})$ and $\partial h_p(\mathbf{w})/\partial \mathbf{w}$ we used. For the considered special case of camera motion, space-variant PSF $h_p(\mathbf{w})$ consists of the

values of $h_p(w)$ standing for space-invariant PSF (mask) for given inverse depth w . These masks are precomputed for a sequence of values of w with constant step Δ_w , i.e., we store $h_p(k\Delta_w)$ for an interval of indices k . During the minimization, intermediate masks are computed by linear interpolation as

$$h_p(w) = \left(\left\lceil \frac{w}{\Delta_w} \right\rceil - \frac{w}{\Delta_w} \right) h_p \left(\left\lfloor \frac{w}{\Delta_w} \right\rfloor \Delta_w \right) + \left(\frac{w}{\Delta_w} - \left\lfloor \frac{w}{\Delta_w} \right\rfloor \right) h_p \left(\left\lceil \frac{w}{\Delta_w} \right\rceil \Delta_w \right). \quad (16)$$

Thanks to linearity, the computation of (6) and (7) for an arbitrary mask takes only about twice the amount of time than for a mask we have stored.

Similarly, $\partial h_p(\mathbf{w})/\partial \mathbf{w}$ is based on $\partial h_p(w)/\partial w$ which is computed from masks stored in another array generated from $h_p(k\Delta_w)$ by taking symmetrical differences of adjacent entries. Again, we use linear interpolation to get the derivatives that are not stored. With higher precision, we can get them directly by application of third-order polynomial fitting filters [29] on $h_p(k\Delta_w)$. Note that the derivatives could be computed analytically using (5) but the way we have just described turned out to be simpler to implement and faster.

Both types of arrays are precomputed for all the images.

E. Minimization of the Cost Functional

Finding the minimum of the cost functional is a high-dimensional nonlinear problem with a huge amount of local minima, especially in the subspace corresponding to variable \mathbf{w} .

Our experiments confirmed that a reasonable choice of initial depth map estimate is essential to prevent the algorithm from becoming trapped in a local minimum. For this purpose we used the algorithm described in Section V-B. Let us denote its result as \mathbf{w}^0 . Note that we also tested random initialization of the depth map but as a rule the minimization resulted in a number of artifacts. Constant initialization did not work at all.

Next, we make use of a sort of alternating minimization (AM) algorithm [13], which basically iterates through minimizations in subspaces corresponding to unknown matrices \mathbf{u} and \mathbf{w} . For reasons explained later, another minimization over the image subspace with different image regularization constant λ_u^f and a higher number of iterations follows at the end of the algorithm.

Algorithm

- 1) for $n = 1 : N_g$
 - 2) $\mathbf{u}^n = \arg \min_{\mathbf{u}} E(\mathbf{u}, \mathbf{w}^{n-1})$
 - 3) $\mathbf{w}^n = \arg \min_{\mathbf{w}} E(\mathbf{u}^{n-1}, \mathbf{w})$
 - 4) end for
 - 5) $\mathbf{u}^{N_g+1} = \arg \min_{\mathbf{u}} E(\mathbf{u}, \mathbf{w}^{N_g})$
-

Note that steps 2), 3), and 5) itself consist of a sequence of iterations.

In the following paragraph, we will discuss the minimization methods used in respective subspaces.

Minimization of functional (9) with respect to \mathbf{u} is the well known and well examined problem of nonblind restoration

[1], [13]. If the regularization term $Q(\mathbf{u})$ is quadratic as in the Q_2 case, the whole problem is linear and we use simple and relatively fast conjugate gradient method [gradients (12) and (14) are obviously linear with respect to \mathbf{u}]. In case of Q_{TV} , matters become more complicated. However, even for this case there are sufficiently efficient algorithms which usually reduce the problem to a sequence of linear problems. We have chosen the approach described in [3]. Note that the authors originally designed their algorithm for denoising and space-invariant restoration problems. Nevertheless, the space-invariant convolution is treated as an almost general linear operator and since the space-variant convolution satisfies assumptions of their method as well, all the arguments are valid and all the procedures can be modified to work with the space-variant case, as well. In a very simplified manner, the idea is as follows.

Let \mathbf{u}_m be the current estimate of the image minimizing the cost functional (9) for a fixed \mathbf{w}^{n-1} . We will replace the regularization term $Q = Q_{TV} = \int \|\nabla(\mathbf{u})\|$ by quadratic term

$$\frac{1}{2} \int \frac{1}{\|\nabla \mathbf{u}_m\|} \|\nabla \mathbf{u}\|^2 + \|\nabla \mathbf{u}_m\|. \quad (17)$$

Obviously, it has the same value as Q_{TV} in \mathbf{u}_m . The right term of (17) is constant for now and consequently it does not take part in actual minimization. We have got a “close” linear problem

$$\mathbf{u}_{m+1} = \arg \min_{\mathbf{u}} \frac{1}{2} \sum_{p=1}^P \|\mathbf{e}_p\|^2 + \lambda_u \int \frac{1}{2\|\nabla \mathbf{u}_m\|} \|\nabla \mathbf{u}\|^2 \quad (18)$$

which becomes a new estimate \mathbf{u}_{m+1} . It can be shown [3] that \mathbf{u}_m converges to the desired minimum for $m \rightarrow \infty$. For numerical reasons, we take $\max(\varepsilon, \|\nabla \mathbf{u}_m\|)$ in place of $\|\nabla \mathbf{u}_m\|$ in (18). The minimization is not very sensitive to the choice of ε and for common images with values in the interval $[0,1]$ can be set to something between 0.001 and 0.01.

Here, we should stress that the use of the conjugate gradient method is crucial for the success of the minimization.

In turn, in the subspace corresponding to depth map we can afford to apply simple steepest descent algorithm. The optimum step length in one direction can be found by the interval bisection method. In this subspace, the convergence turned out to be sufficient to get satisfactory results.

Note that, in both subspaces, we can use TV regularization with very little slowdown since the additional cost of the matrix norm computation is not high compared to space-variant convolution in each step of the minimization algorithm.

Finally, we should mention that we carried out experiments with both types of regularization (Tikhonov and TV) in both subspaces. The choice of image regularization term $Q(\mathbf{u})$ does not seem to have much influence on convergence properties of the minimization and we can freely choose the type that works better for our application. In turn, the use of TV regularization for depth map seems to slow down the convergence to some extent. In most cases we can recommend TV regularization for the image and Tikhonov regularization for the depth map.

F. Scheme of Iterations

First note that this section can be skipped in the first reading as it describes some details of our implementation.

The algorithm consists of three levels of iterations. Experiments showed that the result of minimization and the speed of convergence depends on the number and order of these iterations. To describe the whole sequence, we introduce notation for the number of iterations of particular subproblems.

The outermost level is given by the number of times, the algorithm alternates between the subspaces \mathbf{u} and \mathbf{w} . Recall that it is denoted as N_g in the description of the algorithm.

The minimization over the image \mathbf{u} depends on the type of regularization. In case of Tikhonov regularization, we apply the conjugate gradient method consisting of a certain number of iterations denoted as N_u . If we use TV regularization, the minimization consists of the sequence of linear subproblems (18) solved again by conjugate gradient method. Then, N_{TV} refers to the length of this sequence and N_u relates to the number of iterations of conjugate gradient method used for the minimization of the subproblems.

As regards the subspace corresponding to unknown \mathbf{w} , N_w stands for the number of direction changes of the steepest decent algorithm.

Finally, we can see that at the end of the algorithm (line 5) we repeat certain number of iterations over the image subspace, this time with the different value of image regularization constant λ_u^f . Analogously to line 2, we will denote the number of iterations as N_{TV}^f and N_u^f .

Put together, the whole sequence of iterations will be described as

$$N_g \times (N_{TV} \times N_u + N_w) + N_{TV}^f \times N_u^f.$$

We tested a large amount of possible combinations of these parameters and deduced several general rules. First, it is not efficient to simply minimize over image subspace as far as possible, then over depth map subspace, etc. It has turned out that the minimization is much faster if we make only some small number of iterations in each subspace. A good choice that worked for all our experiments (even for those not included in this paper) was $N_u = 8$ and $N_w = 10$. Interestingly, in the case of TV image regularization it is sufficient to set $N_{TV} = 1$.

The reason why we need the final minimization over the image subspace, is that the alternating minimization is faster if used with more image regularization. Therefore, we can use a larger value of λ_u , which naturally results in somewhat “softer” image and finally sharpen the image by running another minimization over the image subspace with less regularization λ_u^f and a higher number of iterations. We stress that this time it is necessary to repeat the minimization (18) several times to get what we want.

Thus, a typical description of iterations can look like $20 \times (1 \times 8 + 10) + 5 \times 25$. We use it in all the experiments presented in this paper.

G. Choice of Regularization Parameters

Unfortunately, it seems difficult to apply known approaches for estimation of regularization parameters [30] directly to our problem. Nevertheless, a possible direction of future research could be the application of *generalized cross-validation (GCV)* similarly to [31] and [32]. GCV is based on the “leave-one-

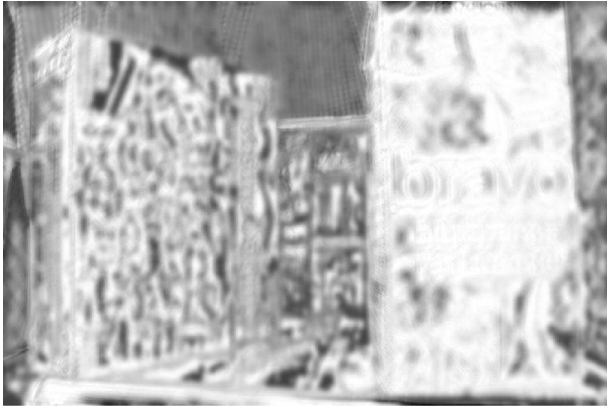
(a) $\lambda_w = 10^{-6}$ (b) $\lambda_w = 10^{-5}$

Fig. 5. Depth maps produced for two different levels of depth map regularization. In both cases, minimization started from depth map Fig. 4(b) with image regularization constant $\lambda_u = 10^{-4}$.

out” principle which basically takes the regularization parameter which is most successful in guessing adjacent points. The difficult part is the estimation of eigenvalues of the operator corresponding to space-variant convolution.

Selection of depth map regularization parameter seems to be even harder to solve due to the nonlinearity of the problem.

The papers working along similar lines [19]–[21] do not address the problem of regularization parameters at all.

In our implementation, we set the parameters by trial and error method as well. Fortunately, the algorithm is not very sensitive to the choice of these constants and if they work for one image with given noise level and amount of blur, it will probably work for other images in the same application, as well.

Another aspect of the issue with the regularization parameters is that we do not have just one correct answer as to what the best solution is. There is always a trade-off between sharpness of the image and noise reduction. We can choose sharper and more noisy (smaller values of λ_u) or softer and less noisy image (larger values of λ_u).

VI. EXTENSION TO GENERAL CAMERA MOTION

If the camera motion and camera parameters (focal length, resolution of the sensor) are known, the proposed algorithm



(a) restoration using depth map 5(a)



(b) restoration using depth map 5(b)

Fig. 6. Results of restoration. We can see that we can get good restoration for different degrees of depth map regularization. For comparison, see ground truth image Fig. 2(c). In both cases $\lambda_u^f = 10^{-4}$.

seems to be extendible to the case of general camera motion where it is necessary to consider all six possible degrees of freedom (DOF) [24]. As this case requires further investigation, we just summarize very briefly the main differences with respect to the special case we have described in this paper. We include it mainly to illustrate a possible direction of future research.

The functional remains the same, except the PSFs $h_p(\mathbf{w})$. The main issue arises from the fact that h_p is a function of not only depth but also of coordinates (x, y) . In other words, different points of the scene draw different apparent curves during the motion even if they are of the same depth. In addition, the depth map is no longer common for all the images and consequently, for $p > 1$, it must be transformed to the coordinate system of the image p before computing h_p using (2) and (3). The same is true in the auxiliary algorithm for the estimation of initial depth map, where the convolution becomes space-variant convolution.

The formulas (14) and (15) hold, in general cases, as well [24], and so the main issue remains how to compute h_p and its gradient for arbitrary (x, y) . Since we cannot store it for every possible (x, y) , a reasonable solution seems to store them only on a grid of positions and compute the rest by interpolation. The density of this grid will depend on application. However, the numerical integration of the velocity field can be quite time-consuming even for a moderate size set of coordinates.

(a) 800×500 image blurred by space-variant motion blur (first image)(b) 800×500 image blurred by space-variant motion blur (second image)

Fig. 7. We took two images from the camera mounted on the framework limiting camera motion to one vertical plane (red channel). For both images, the shutter speed was set to 1.3 s and aperture to $F/22$.

In turn, a nice property of this approach is that, once all the PSFs are precomputed, both the depth map estimate and minimization do not take much longer than in the case of the translational motion described in previous sections.

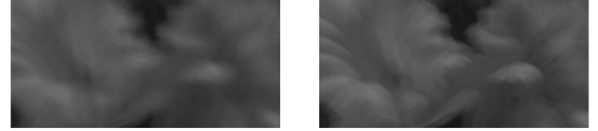
VII. EXPERIMENTS

We present three experiments with real images. All of them were taken as RAW files by a digital SLR camera mounted on a framework that limits motion or vibrations to one vertical plane. We work with red channel (the first experiment) or green channel (the second and third experiments), scaled down by a factor of four.

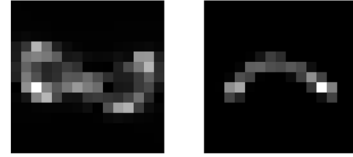
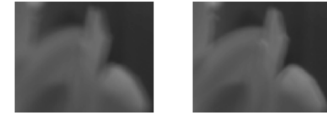
The first experiment documents behavior of our algorithm for images blurred by 1-D harmonic motion of the camera. The scene was chosen relatively simple but so that the extent of blur varies significantly throughout the image.

We took two images [Fig. 2(a) and (b)] (red channel) from a camera mounted on the device vibrating approximately in (a) horizontal and (b) vertical directions, both with shutter speed $T = 5$ s. Note that such 1-D motions were considered in [14] for flat scenes. To achieve a large depth-of-field, we set f-number to $F/16$. The third image [Fig. 2(c)] was taken without vibrations and we use it as ground truth.

In the first phase of the algorithm, we took two small sections [Fig. 3(a)] from the right part of the input images and com-



(a) sections of images Fig. 7(a) and (b) used for the estimate of PSFs taken from the foreground part of the image (353×167 pixels, $2 \times$ enlarged)

(b) 15×15 PSFs computed from images (a)

(c) another section from the upper-right corner of the LCD screen in the background (67×54 pixels, $3 \times$ enlarged)

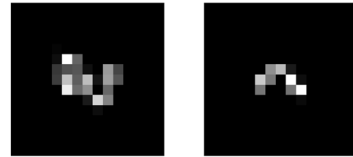
(d) 15×15 PSFs computed from image sections (c)

Fig. 8. Algorithm needs an estimate of PSF for at least one distance from camera. We took a central part of the images in Fig. 7(a) and (b) where the degree of blur was approximately constant and computed PSFs (b) using space-invariant blind restoration method [10]. For comparison, we computed (d) PSFs from (c) background sections. We can see that in agreement with our model, (d) the PSFs are a scaled down version of (b) PSFs.

puted PSFs [Fig. 3(b)] using space-invariant blind restoration algorithm [10] (with parameters $\lambda = 1000$, $\varepsilon = 0.1$, $\gamma = 10$, support of both PSFs was set to 11×11 pixels). Then, we pre-computed PSFs using relation (5) for $w = 0.1$ with step $1/500$, that is we store altogether 501 PSFs.

To demonstrate the space-variance of the blur in our images we took another section [Fig. 3(c)] from the image center (bear in waterfall) and computed PSFs [Fig. 3(d)], again using the method [10]. We can see that the extent of the blur is about half compared to the PSFs [Fig. 3(b)] which is in agreement with our model.

In the second phase, we got an initial estimate of depth map [Fig. 4(b)]. In the algorithm, the error was averaged by a 7×7 window. Afterward, the result was smoothed by 23×23 median filter. Direct restoration using this depth map results in strong artifacts in the whole area of the image, as shown in Fig. 4(a).

Next, we applied the minimization procedure from Section V-E. Figs. 5 and 6 show depth maps and restored images for two different levels of depth map regularization. In both cases, we used the same image regularization constant $\lambda_u = 10^{-3}$ for the alternating minimization and $\lambda_u^f = 10^{-4}$ for final restoration. The image regularization constant does not have much influence on the produced depth map. The influence on the restored image is well described in literature [1]. We have visually almost undistinguishable results for different depth maps. In the following experiment, we will show that in case of more com-

(a) $\lambda_w = 10^{-6}$ (b) $\lambda_w = 5 \times 10^{-6}$

Fig. 9. Depth maps computed from the blurred images of Fig. 7(a) and (b) for two different levels of Tikhonov depth map regularization.

plex scene we must choose the depth map regularization constant more carefully.

The second experiment was set up to show limitations of the proposed algorithm. The scene is much more complex with a lot of small details and there are many depth discontinuities. Also the camera motion is more complex. It consists of a flower placed in front of an LCD. There is a white wall with almost no texture in the background. The structure of the experiment is similar to the previous one.

Images [Fig. 7(a) and (b)] (green channel) were taken from the same device as in the previous experiment. This time the shutter speed was set $T = 1.3$ s and f-number to $F/22$. We made the framework quiver by a random impulse of hand and took two images in a rapid sequence. Again, the third image [Fig. 10(c)] was taken by motionless camera and we use it as ground truth.

The same way as in the previous experiment, we computed the PSFs for one distance from camera using algorithm [10] (with parameters $\lambda = 1000$, $\varepsilon = 0.1$ and $\gamma = 10$ for the larger mask of size 15×15 and $\lambda = 10^4$, $\varepsilon = 0.1$ and $\gamma = 10$ for the smaller mask of size 11×11). For this purpose, we chose the area close to the image center with the most blurred blossoms Fig. 8(a). Resulting masks are in Fig. 8(b). For comparison, we cropped sections Fig. 8(c) and computed masks Fig. 8(d) corresponding to the upper-right corner of the LCD screen in the background part of the image. Again, we can see that (5) approximately holds.



(a) restoration using depth map 9(a)



(b) restoration using depth map 9(b)



(c) ground truth image

Fig. 10. Results of restoration. We can see that (a) lesser depth map regularization may result in artifacts in the areas of weak texture (wall in the background). A higher degree of (b) regularization caused more visible artifacts on the edges (edge between blossoms near the right edge of the LCD screen). For comparison, the third image was taken at rest serving as “ground truth.”

Then we applied (8) to get an initial estimate of depth map for the minimization. Figs. 9 and 10 give results of the whole algorithm for two degrees of depth map regularization. Again, in both cases, we used the same image regularization constant $\lambda_u = 10^{-3}$ for the alternating minimization and $\lambda_u^f = 10^{-4}$ for final nonblind restoration.

We can see that if we use less regularization, there are visible wave-like artifacts on the wall in the background. On the other hand, if we use more regularization, it pronounces visible ringing effects in the places where the distance from the camera

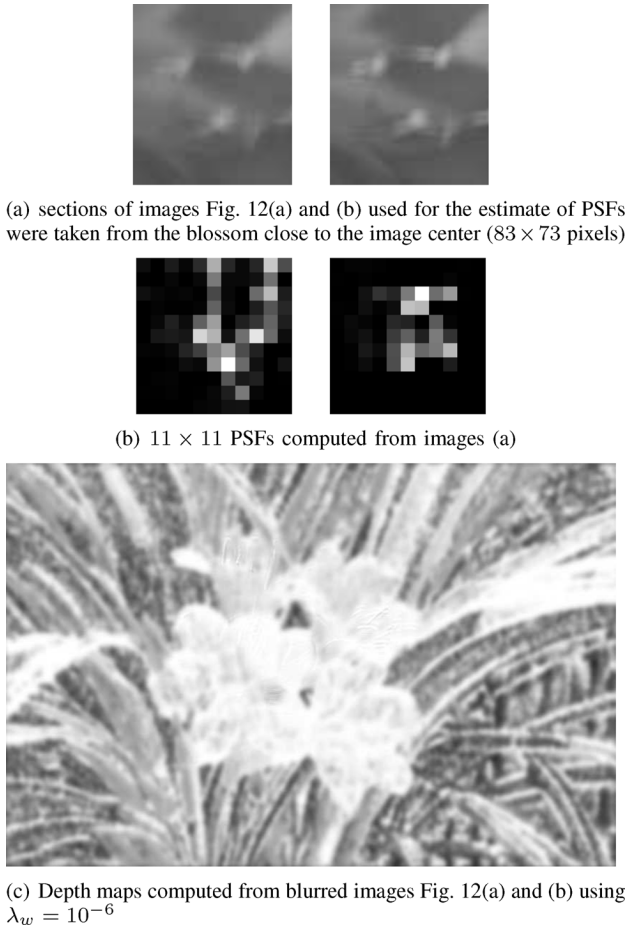


Fig. 11. PSFs computed in the first phase of the algorithm from sections of images Fig. 12(a) and (b) using the blind space-invariant restoration method [10].

suddenly changes. Sometimes we must make a compromise according to the situation. Note that some error cannot be avoided in principle without taking into account occlusions present on the edges of objects [33], [34]. It is probably one of reasons why there are some visible artifacts close to the edges.

We should also remark that the depth map estimate is not very good in this case. The main reason is the complexity of the scene that results in poor performance of the auxiliary algorithm for initial depth map estimation. Fortunately, in most cases, it does not affect restoration seriously.

In the third experiment, we consider the interesting case of large aperture ($F/4$), which results in small depth-of-field. The scene, as well as camera motion, are again quite complex. We work with green channels, reduced in half to 870×580 pixels. The scene contains blossoms in focus and some leaves around it that are out-of-focus. It is a setup typical for macro photography. Our goal is to remove camera shake but to preserve the depth of field.

The experiment was carried out the same way as the previous one. The PSF is recovered from the blossom area [Fig. 11(a)]. Remark that this is the only area we can use since this is the only area in focus as discussed in Section V-A. Here, we present the result of minimization with $\lambda_u = 10^{-3}$ and $\lambda_w = 10^{-6}$.

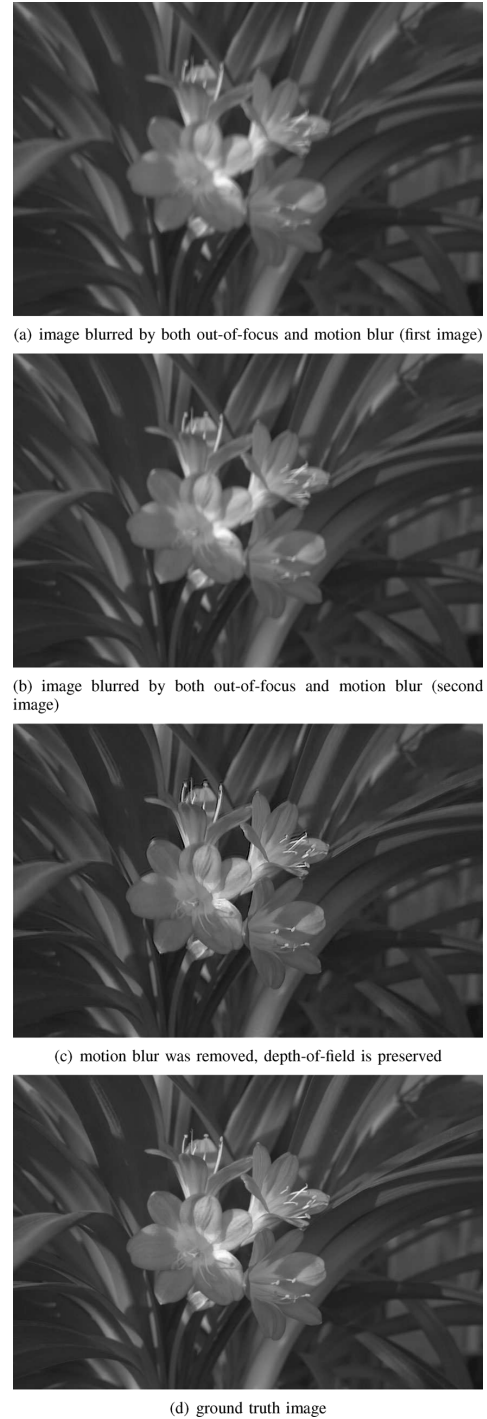


Fig. 12. We took images (a) and (b) with the aperture set to $F/4$ to achieve small depth-of-focus. (c) The algorithm removed the motion blur. The image (d) was taken without vibrations.

An important conclusion is that the depth map estimate is sufficiently precise even in the out-of-focus areas. It may be less precise than in the areas in focus but the human eye is less sensitive to small deviations in the blurred parts of the image.

VIII. CONCLUSION

We have presented an algorithm for image restoration and simultaneous estimation of depth map from multiple images of

the same scene blurred by camera motion. We considered the special case where the camera moves in one plane perpendicular to the optical axis. This type of motion is significantly more complex than those considered in previous literature. We have shown that the algorithm works well with real images.

There are two important directions of future research.

First, we have not considered finite-aperture effects in a full scale. As discussed in Section V-A, in the case of limited depth of field, the algorithm works only if there is a convenient flat part of the scene in focus. In this case, the algorithm removes the camera shake but preserves the depth of field. A possible direction of future research would be to remove the out-of-focus blur as well.

Another interesting problem is the extension to general camera motion. Some ideas on this subject were discussed in Section VI.

REFERENCES

- [1] M. R. Banham and A. K. Katsaggelos, "Digital image restoration," *IEEE Signal Process. Mag.*, vol. 14, no. 2, pp. 24–41, Mar. 1997.
- [2] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D*, vol. 60, pp. 259–268, 1992.
- [3] A. Chambolle and P. Lions, "Image recovery via total variation minimization and related problems," *Numer. Math.*, vol. 76, no. 2, pp. 167–188, Apr. 1997.
- [4] M. Ben-Ezra and S. K. Nayar, "Motion-based motion deblurring," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 6, pp. 689–698, Jun. 2004.
- [5] D. Kundur and D. Hatzinakos, "Blind image deconvolution," *IEEE Signal Process. Mag.*, vol. 13, no. 3, pp. 43–64, May 1996.
- [6] M. Ng, R. Plemmons, and S. Qiao, "Regularization of RIF blind image deconvolution," *IEEE Trans. Image Process.*, vol. 9, no. 6, pp. 1130–1138, Jun. 2000.
- [7] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, "Removing camera shake from a single photograph," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 787–794, 2006.
- [8] G. Harikumar and Y. Bresler, "Efficient algorithms for the blind recovery of images blurred by multiple filters," in *Proc. IEEE Int. Conf. Image Processing*, Lausanne, Switzerland, 1996, vol. 3, pp. 97–100.
- [9] G. Harikumar and Y. Bresler, "Perfect blind restoration of images blurred by multiple filters: Theory and efficient algorithms," *IEEE Trans. Image Process.*, vol. 8, no. 2, pp. 202–219, Feb. 1999.
- [10] F. Šroubek and J. Flusser, "Multichannel blind iterative image restoration," *IEEE Trans. Image Process.*, vol. 12, no. 9, pp. 1094–1106, Sep. 2003.
- [11] R. L. Lagendijk and J. Biemond, "Block-adaptive image identification and restoration," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, May 1991, pp. 2497–2500.
- [12] M. K. Ozkan, A. M. Tekalp, and M. I. Sezan, "Identification of a class of space-variant image blurs," in *Proc. SPIE Conf. Image Processing Algorithms and Techniques II*, San Jose, CA, Jun. 1991, pp. 146–156.
- [13] Y.-L. You and M. Kaveh, "Blind image restoration by anisotropic regularization," *IEEE Trans. Image Process.*, vol. 8, no. 3, pp. 396–407, Mar. 1999.
- [14] A. Rav-Acha and S. Peleg, "Restoration of multiple images with motion blur in different directions," in *Proc. 5th IEEE Workshop on Applications of Computer Vision*, Dec. 2000, pp. 22–28.
- [15] X. Liu and A. E. Gamal, *Simultaneous Image Formation and Motion Blur Restoration via Multiple Capture*, vol. 3, pp. 1841–1844, May 2001.
- [16] Y. F. Wang and P. Liang, "3D shape and motion analysis from image blur and smear: A unified approach," in *Proc. IEEE Int. Conf. Computer Vision*, 1998, pp. 1029–1034.
- [17] I. M. Rekleitis, "Optical flow recognition from the power spectrum of a single blurred image," in *Proc. Int. Conf. Image Processing*, Sep. 1996, vol. 3, pp. 791–794.
- [18] W. G. Chen, N. Nandhakumar, and W. N. Martin, "Image motion estimation from motion smear: A new computational model," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 18, no. 4, pp. 412–412, Apr. 1996.
- [19] A. N. Rajagopalan and S. Chaudhuri, "An MRF model-based approach to simultaneous recovery of depth and restoration from defocused images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 7, pp. 577–589, Jul. 1999.
- [20] P. Favaro, M. Burger, and S. Soatto, "Scene and motion reconstruction from defocus and motion-blurred images via anisotropic diffusion," in *Proc. ECCV*, vol. 3021, pp. 257–269.
- [21] P. Favaro and S. Soatto, "A variational approach to scene reconstruction and image segmentation from motion-blur cues," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2004, vol. 1, pp. 631–637.
- [22] D. L. Tull and A. K. Katsaggelos, "Regularized blur-assisted displacement field estimation," in *Proc. Int. Conf. Image Processing*, Sep. 1996, vol. 3, pp. 85–88.
- [23] D. J. Heeger and A. D. Jepson, "Subspace methods for recovering rigid motion," *Int. J. Comput. Vis.*, vol. 7, no. 2, pp. 95–117, 1992.
- [24] M. Šorel, "Multichannel blind restoration of images with space-variant degradations," Ph.D. dissertation, Charles Univ., Prague, Czech Republic, 2007.
- [25] A. Tikhonov and V. Arsenin, *Solution of Ill-Posed Problems*. New York: Wiley, 1977.
- [26] D. Tschumperlé and R. Deriche, "Vector-valued image regularization with pdes: A common framework for different applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 4, pp. 506–517, Apr. 2005.
- [27] F. Šroubek and J. Flusser, "Multichannel blind deconvolution of spatially misaligned images," *IEEE Trans. Image Process.*, vol. 14, no. 7, pp. 874–883, Jul. 2005.
- [28] D. Tschumperlé and R. Deriche, "Diffusion PDE's on vector-valued images," *IEEE Signal Process. Mag.*, vol. 19, no. 5, pp. 16–25, Sep. 2002.
- [29] P. Meer and I. Weiss, "Smoothed differentiation filters for images," *J. Vis. Commun. Image Represent.*, vol. 3, pp. 58–72, 1992.
- [30] N. P. Galatsanos and A. K. Katsaggelos, "Methods for choosing the regularization parameter and estimating the noise variance in image restoration and their relation," *IEEE Trans. Image Process.*, vol. 1, no. 3, pp. 322–336, Jul. 1992.
- [31] G. H. Golub, M. Heath, and G. Wahba, "Generalized cross-validation as a method for choosing a good ridge parameter," *Technometrics*, no. 21, pp. 215–223, 1979.
- [32] N. Nguyen, P. Milanfar, and G. Golub, "Efficient generalized cross-validation with applications to parametric image restoration and resolution enhancement," *IEEE Trans. Image Process.*, vol. 10, no. 9, pp. 1299–1308, Sep. 2001.
- [33] S. Bhasin and S. Chaudhuri, "Depth from defocus in presence of partial self occlusion," in *Proc. IEEE Int. Conf. Computer Vision*, Jul. 2001, vol. 1, pp. 488–493.
- [34] P. Favaro and S. Soatto, "Seeing beyond occlusions (and other marvels of a finite lens aperture)," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2003, vol. 2, pp. 579–586.



Michal Šorel received the M.Sc. and Ph.D. degrees in computer science from the Charles University, Prague, Czech Republic, in 1999 and 2007, respectively.

He is currently with the Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic.



Jan Flusser (M'93–SM'03) received the M.Sc. degree in mathematical engineering from the Czech Technical University, Prague, Czech Republic, in 1985, the Ph.D. degree in computer science from the Czechoslovak Academy of Sciences in 1990, and the D.Sc. degree in technical cybernetics in 2001.

Since 1985, he has been with the Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, Prague. From 1995 to 2006, he held the position of Head of the Department of Image Processing. In 2007, he was appointed Director of the Institute. Since 1991, he has also been affiliated with the Charles University, Prague, and the Czech Technical University, where he gives courses on digital image processing and pattern recognition. He has been a Full Professor since 2004. His current research interests include all aspects of digital image processing and pattern recognition, namely 2-D object recognition, moment invariants, blind deconvolution, image registration, and image fusion. He has authored and coauthored more than 150 research publications in these areas.