Charles University in Prague
Faculty of Mathematics and Physics

Multichannel Blind Restoration of Images with Space-Variant Degradations

Ph.D. Thesis

Michal Šorel

March 2007

Department of Software Engineering
Faculty of Mathematics and Physics
Charles University in Prague

Supervisor: Prof. Ing. Jan Flusser, DrSc.
Institute of Information Theory and Automation
Academy of Sciences of the Czech Republic

Declaration

This thesis is submitted for the degree of Doctor of Philosophy at Charles University in Prague. The research described herein was conducted under the supervision of Professor Jan Flusser in the Department of Image Processing, Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic.

Except where explicit reference is made, this thesis is entirely the outcome of my own work and includes nothing done in collaboration. No part of this work has been submitted for a degree or diploma at any other university. Some of the work contained in this thesis has been published.

Michal Šorel
March 2007

Acknowledgments

This would not have been possible without the help and support of my advisor Professor Jan Flusser. His guidance and assistance are deeply appreciated. Many thanks to my colleague Filip Šroubek for valuable discussions and helpful feedback. Finally, I would like to thank family and friends for their support.

Research has been supported by the Czech Ministry of Education, Youth, and Sports under the project 1M0572 (Research Center DAR) and by the Grant Agency of the Czech Republic under the project 102/04/0155.

Abstract

In this thesis, we cover the related problems of image restoration and depth map estimation from two or more space-variantly blurred images of the same scene in situations, where the extent of blur depends on the distance of scene from camera. This includes out-of-focus blur and the blur caused by camera motion. The latter is typical when photographing in low-light conditions.

Both out-of-focus blur and camera motion blur can be modeled by convolution with a spatially varying point spread function (PSF). There exist many methods for restoration with known PSF. In our case, the PSF is unknown as it depends on depth map of the scene and camera motion. Such a problem is ill-posed if only one degraded image is available. We consider multichannel case, when at least two images of the same scene are available, which gives us additional information that makes the problem tractable.

The main contribution of this thesis, Algorithm I, belongs to the group of variational methods that estimate simultaneously sharp image and depth map, based on the minimization of a cost functional. Compared to other existing methods, it works for much broader class of PSFs.

In case of out-of-focus blur, the algorithm is able to consider optical aberrations.

As for camera motion blur, we are concerned mainly with the special case when the camera moves in one plane perpendicular to the optical axis without any rotations. In this case the algorithm needs to know neither camera motion nor camera parameters. This model can be valid in industrial applications with camera mounted on vibrating or moving devices. In addition, we discuss the possibility to extend the described algorithm to general camera motion. In this case, the knowledge of camera motion is indispensable. In practice, information about the motion could be provided by inertial sensors mounted on the camera.

Besides, we present two filter-based methods for depth map estimation based on the measurement of the local level of blur. Algorithm II is a fast method working for arbitrary sufficiently symmetrical blurs using only two convolutions. Algorithm III places no constraints on the shape of PSF at the expense of higher time requirements.

Finally, we propose an extension of Algorithms I and III to color images.

Contents

List of figures

Chapter 1

Introduction

Subject to physical and technical limitations, the output of digital imaging devices, such as cameras, microscopes and astronomical telescopes, is not perfect and substantial part of image processing research focuses on removing of various types of image degradations.

1.1 Out-of-focus and camera motion blur

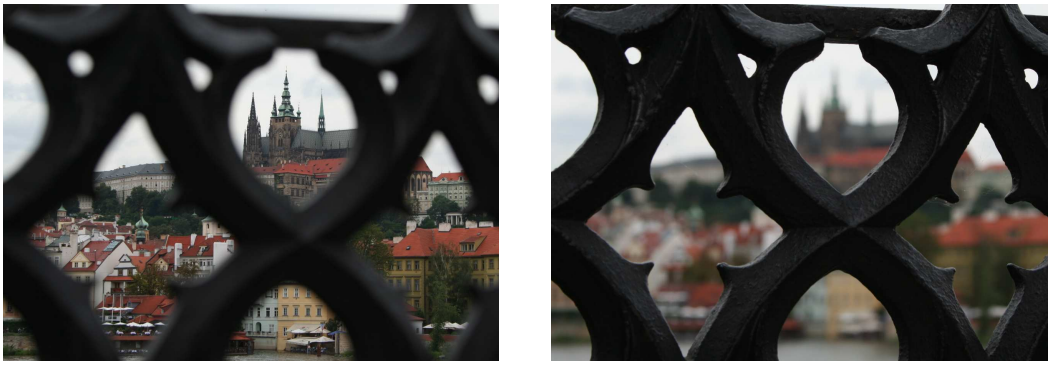
The most frequent degradations are perceived by humans as blur and noise. They can be usually modeled with reasonable precision by linear relation

$$\mathbf{z}(x,y) = \int_{\Omega} \mathbf{u}(x-s, y-t)\mathbf{h}(x-s, y-t; s, t)\, dsdt + \mathbf{n}(x,y), \qquad (1.1)$$

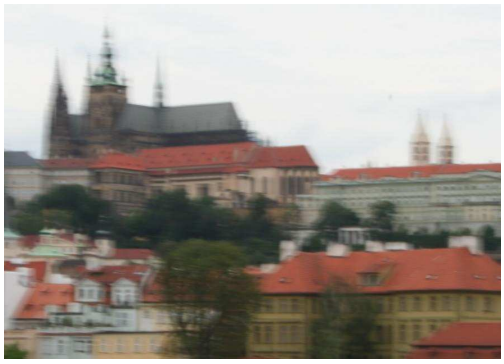
where $\mathbf{u}$ is an *ideal image*[1], $\mathbf{h}$ is called *point-spread function* (*PSF*), $\mathbf{n}(x,y)$ is additive signal independent noise[2] and $\mathbf{z}$ the blurred and noisy image. The integral term of (1.1) can be viewed as smearing of each point (x,y) of the image $\mathbf{u}$ into a blob of the shape given by $\mathbf{h}(x,y;s,t)$. 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 matrix is called *convolution mask* or simply mask. We will use this term in general *space-variant* case as well in the sense that the mask is considered for each image pixel separately.

[1] We can also encounter expressions *scene radiance*, *sharp image* or *original image*. Alternatively we could speak about the image we would get by hypothetical camera with infinitely small aperture and free of diffraction effects. This so called *pinhole camera* model is often used in stereo applications.

[2] The most widespread image sensors based on CCD and CMOS technologies are subject to multiplicative (speckle) noise as well. For the purposes of this work, this phenomenon can be neglected.



(a) real digital camera has a finite depth of focus



(b) typical image blurred by camera shake, shutter speed $1/15\,s$

Figure 1.1: Digital images are often subject to out-of-focus or motion blur.

While space-invariant case has been extensively studied, in more difficult space-variant case there are much more open problems to resolve. The latter case is the subject of this thesis.

We are interested in two important types of space-variant blur, namely *out-of-focus blur* (*defocus*) and *camera motion blur*. Both types of blur have common property that the extent of blur depends on the distance of objects from camera.

Figure 1.1(a) illustrates the fact that real cameras have a finite depth of focus and the whole image can be perfectly in focus only if the whole scene is in the same distance from camera. Figure 1.1(b) is an example of image blurred by camera shake which happens when we take photographs from hand at long shutter speeds. It is typically unavoidable in low-light conditions.

Now, we briefly characterize the PSF corresponding to the blurs we are discussing. They are treated in detail in Chapters 4 and 5.

In case of defocus, if we assume simple Gaussian optics and circular aperture, the graph of PSF has a cylindrical shape usually called *pillbox* in literature. It's radius r is a linear function of the reciprocal of the distance l from camera, namely

$$r = \frac{1}{l}\rho\zeta + \rho\zeta\left(\frac{1}{\zeta} - \frac{1}{f}\right). \tag{1.2}$$

Here f stands for focal length, ρ is aperture radius and ζ the distance of the image plane from the optical center. Note that the distance l is measured along the optical axis and often is referred to as *depth*. When we describe appearance of this PSF in an image or a photograph, we speak about *blur circle* or *circle of confusion*. In many cases, the PSF can be better approximated by two-dimensional Gaussian function with variance again related to the object distance. As a rule, these models work well for high quality optics. Otherwise, even for objects of the same distance, PSF changes as a function of where the camera is focused and also of the coordinates (x, y) themselves. For details see Chapter 4.

The second considered type of blur is the motion blur due to camera motion. If we assume planar scene perpendicular to the optical axis and steady motion of the pinhole camera[1] in a plane parallel to the scene, it is well known that the PSF is space-invariant one-dimensional rectangular impulse in the direction of camera motion. The length of the impulse is inversely proportional to the distance from camera. This situation can be extended to the case when the camera moves, as in the steady case, in one plane perpendicular to the optical axis without any rotations but can change its speed and motion direction. Then, the size of PSF

$$\frac{l^2}{\zeta^2}\mathbf{h}_0(\frac{l}{\zeta}s, \frac{l}{\zeta}t) \tag{1.3}$$

is again inversely proportional to the distance l from camera. Function $\mathbf{h}_0(s, t)$ corresponds to the path covered by the camera during the time the shutter is open. This model can be valid for example for cameras mounted on vibrating or moving devices. For distant objects or scenes taken with a longer focal length the dominant camera motion is rotation. Then PSF does not depend on the distance from camera and the problem can be converted to simpler space-invariant case not treated in this work. In general case, the PSF can be very complex depending on the camera motion, depth of scene and parameters of the optical system. For details see Chapter 5.

1.2 Terminology of related image processing techniques

There are several frequently used terms referring to the image processing techniques related to the presence of blur in the images.

The problem to find the sharp image **u** when we know the blurred image **z** and the degradation **h** is called *restoration*, *deblurring* or, especially if **h** is space-invariant, *deconvolution*. If even the PSF **h** is not known, we speak about *blind* restoration or deconvolution. The problem of blind restoration from one image is ill-posed. However, if we have at least two observations of the same scene taken with different camera settings, it gives us additional information that makes the task tractable. This situation is referred to as *multichannel* (*MC*) restoration.

The complementary problem to recover the blur **h** is an integral part of many blind restoration algorithms but can be interesting in itself. We can take advantage of the fact, that the amount of blur is a function of distance and take its inverse to recover the three-dimensional structure of the scene. This structure is usually represented by *depth map*, i. e. the matrix of the same size as the image, where each element gives the depth of the part of the scene imaged to the corresponding pixel of the image.

Depth from defocus (*DFD*) can be defined as the task to recover depth map if we know a small set (usually two or three) of blurred images taken from the same place with different camera settings. DFD as approach to passive ranging developed as an alternative to *depth from focus* (*DFF*) methods. The idea behind DFF is that we successively focus at all the distances potentially occurring in the scene and determine the distance related to certain pixel by choosing the image that is least out-of-focus in its neighborhood [3]. An important application area of both DFD and DFF approaches is microscopy. In turn, for large-scale scenes it is often better to use stereo techniques [4], which are more precise thanks to larger physical size of stereo base compared to aperture diameter [5], and work even for fast-moving scenes.

The main drawback of DFF approach is that it involves lengthy focusing motion of camera lens over the large range of positions, while DFD needs just two or three positions or it is even possible to eliminate focusing completely by changing of the aperture instead of the distance, where the camera is focused. Thus, for example in microscopy, DFD could be a useful alternative to DFF, especially when the observed specimen moves. We can imagine a large-scale application of DFD as well if the precision of depth measurements is of no concern. An example of such an application is rough estimation of depth map necessary for initialization of variational restoration methods. Com-

pared to stereo methods, DFD does not suffer from correspondence problems and occlusions happen only at object edges and can be mostly neglected.

Motion blur can be used in a way similar to DFD [6]. We have not found any generally accepted name for this group of techniques, so we will call it simply depth estimation based on motion blur or, in short, *depth from motion blur*. Besides, by the extraction of *optical flow (OF)* we mean the recovery of the direction and the extent of apparent motion corresponding to the given part of the image. Some OF algorithms use motion blur to recover OF and since the extent and direction of blur correspond to local optical flow, they can be used to recover depth maps as well. Similarly to DFD, these methods can be used as part of restoration algorithms.

1.3 Problem statement

The topic of this thesis is restoration of images blurred by space-variant blur with the property that the extent of blur is a function of the distance from camera. This includes out-of-focus blur and the blur caused by camera motion.

Both out-of-focus and camera motion blur can be modeled by convolution with a spatially varying PSF. There exist many techniques for restoration with known PSF. In our case, the PSF is unknown as it depends on camera motion and depth map of the scene. Such a problem is ill-posed if only one degraded image is available. We consider multichannel case, when at least two images of the same scene are available, which gives us additional information that makes the problem tractable.

Most of existing algorithms for space-variant restoration are based on the assumption that the character of blur does not change in a sufficiently large neighborhood of each image pixel, which simplifies solution of the problem.

For space-variant blur caused by camera motion or defocus 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 [7]. This approach was adopted by Favaro et al. [8, 9] who modeled camera motion blur by Gaussian PSF, locally deformed according to the direction and extent of blur. This method can be appropriate for small blurs.

The idea behind variational methods is as follows. Assume that we are able to describe mathematically the process of blurring, in our case using linear relation (1.1) and knowledge of the relation between the PSF and the depth of the scene for given camera parameters. Algorithm is looking for

such a (sharp) image and depth map that, after 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 value as possible. Thus, solution of the problem is equivalent to the minimization of the functional. Algorithms can differ in the precise shape of the resulting functional and methods used for its minimization.

All previously published variational algorithms suffer from weaknesses that limit their use in practical applications. They are outlined in the rest of this section.

First of all, the existing variational algorithms work with Gaussian PSF. As regards out-of-focus, the PSF of a real lens system can significantly differ from Gaussian function and this limits precision of the algorithm. Modelling of motion blur by Gaussian PSF is impossible in all non-trivial cases, except of very slight blurs.

Another issue with variational methods in general is that they are based on the minimization of complex functionals, which can be very time-consuming in the space-variant case. It is probably the reason, why these methods did not appear until recently. One way around it is parallelization for which, at least in principle, this approach is well suited. Unfortunately, for the previously published algorithms, possible level of parallelization is limited because each of parallel units has to be able to compute rather complicated Gaussian function.

The final difficulty with variational approach we should mention is that the corresponding functional has many local minima and consequently it can be hard to guarantee location of the correct global minimum. In theory, we could apply simulated annealing [7], which guarantees global convergence, but it is too slow to be used in practice.

1.4 Goals

The main goal of this thesis is to develop new methods for restoration of images with space-variant degradations with accent on out-of-focus and camera motion blur.

In particular, an algorithm or algorithms should be developed that overcome weaknesses of published methods mentioned in the previous section. They should work

1. with only two input images (from one image the problem is not well posed),

2. without any restrictions on scene such as a small number of parallel planes perpendicular to the optical axis (unlike for example [10, 11]) or condition that every part of the image is sharp in at least one of the input images (unlike image fusion methods such as [12]),

3. with PSF that cannot be well approximated using simple models such as Gaussian or pillbox,

4. with motion blurred images, which is not well treated in literature. Investigate non-trivial types of camera motion with potential applications in the reduction of camera shake.

5. If possible, algorithms should be easily implementable, operations should be as simple as possible to facilitate hardware implementation.

1.5 Contributions

This section gives the overview of the key results presented in this thesis. In Section 1.5.3, we list the publications of the author.

1.5.1 Algorithm I

The main contribution of this thesis, Algorithm I, belongs to the group of variational methods estimating the sharp image from two or more space-variantly blurred images of the same scene [7, 8, 9].

Algorithm I was designed to overcome the weaknesses of existing variational methods described in the problem statement. For out-of-focus blur, it assumes two or more images of the same scene taken from the same place with different camera parameters. In turn, for the case of camera motion, the camera parameters are supposed to be the same and the camera motion is different. In the basic version of the algorithm, the camera motion is limited to one plane perpendicular to the optical axis and this limitation includes the change of camera position between the images. In this special case the algorithm needs to know neither camera motion nor camera parameters. The algorithm can be modified to work with color images and we discuss the possibility of extension to general camera motion as well.

Now we indicate the ways, Algorithm I deals with the issues outlined in the problem statement (Section 1.3).

Unlike the existing methods, our algorithm works independently of a particular shape of PSF. The idea is to approximate the relation between distance and PSF by a finite number of masks stored in memory and compute

intermediate masks by polynomial interpolation. The interpolation makes it possible to work with ordinary minimization algorithms.

This approach is especially useful in situations when PSF is not given analytically. For out-of-focus blur, in case of significant optical aberrations, it is easy to get the PSF of a particular lens system by a raytracing algorithm or by a measurement but difficult to express it explicitly by an equation. This approach can be naturally applied to motion blur as well. Indeed, to the best of our knowledge, it is the first time, any space-variant restoration algorithm works for a complex type of camera motion.

The second advantage of this approach is that in the course of minimization it uses only elementary point-wise matrix operations, vector dot products and two linear operations that can be seen as extensions of convolution and correlation to space-variant case—we refer to them as "*space-variant convolution*" (3.1) and "*space-variant correlation*" (3.2). Besides being faster in itself, we believe that this approach can simplify construction of multipurpose parallel hardware working for both out-of-focus and motion blur with other potential applications in image and video processing.

To avoid the problem with the existence of many local minima, [7] used method [1] for initial estimate of depth map. Algorithm I keeps this idea, but since we work with more general class of blurs, we extended method [1] to work with more general class of symmetrical PSFs, resulting in Algorithm II. Unfortunately, there are important applications, such as reduction of camera shake, where PSFs are not symmetrical. For this case we developed a new filter-based depth estimation method described in this thesis as Algorithm III.

The basic assumption of the used approach is the knowledge of the relation between the PSF and the depth of the scene. As mentioned above, if we know the arrangement of lenses, the PSF of an optical system can be computed by a raytracing algorithm. Another possibility is taking a picture of a grid of point sources, which gives directly PSFs for the whole field of view. Of course, it must be done for all combinations of possible camera parameters and possible depths of scene.

As for the blur caused by camera motion, besides somewhat impractical hybrid systems [13], the relation between PSF and distance can be computed from data gather by inertial sensors trekking motion of the camera. However, if the camera is constrained to move only in one plane perpendicular to the optical axis without any rotations, we can apply a blind space-invariant restoration method on a flat part of the scene to get the mask for one distance from camera. Then it is possible to compute masks for arbitrary distance. We already mentioned that this limitation is assumed in the basic version of

I and was also used in our experiments.

1.5.2 Algorithms II and III

Both algorithms were developed as auxiliary methods used for initial depth map estimates for Algorithm I. However, especially Algorithm III turned out to be interesting on its own.

Algorithm II is a modification of filter based DFD method [1] to work with arbitrary sufficiently symmetrical PSF for both out-of-focus and motion blur. Its primary merit is speed, main weakness its sensitivity to noise and limited precision, especially in the areas of weak texture. Besides, it requires careful calibration to provide applicable results.

Algorithm III is another filter based depth recovery method, which works for arbitrary type of PSF at the expense of higher time consumption. Compared to Algorithm II, it is more stable in the presence of noise and is also less sensitive to the precise knowledge of the PSF. Since it places no requirements on the symmetry of the PSF, Algorithm III can be applied on images blurred by camera motion blur, where we meet very irregular PSFs. This algorithm, the same way as Algorithm I, has potential to be extended to general camera motion.

1.5.3 Publications of the author

Preliminary versions of Algorithm I were published as [14, 15, 16]

- M. Šorel and J. Flusser, "Blind restoration of images blurred by complex camera motion and simultaneous recovery of 3D scene structure," in *Proceedings of the Fifth IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), Athens*, Dec. 2005, pp. 737–742.

- M. Šorel and J. Flusser, "Simultaneous recovery of scene structure and blind restoration of defocused images," in *Proceedings of the Computer Vision Winter Workshop 2006. CVWW'06.*, O. Chum and V. Franc, Eds. Czech Society for Cybernetics and Informatics, Prague, 2006, pp. 40–45.

- M. Šorel, "Multichannel blind restoration of images with space-variant degradations," Research Center DAR, Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, Prague, Tech. Rep. 2006/28, 2006.

Complete version, covering Chapters 5, 6 and 8 and part of Chapter 11, was submitted as [17]

- M. Šorel and J. Flusser, "Space-variant restoration of images degraded by camera motion blur," *IEEE Trans. Image Processing*, 2007, submitted.

Color extension of the algorithm was submitted as [18]

- M. Šorel and J. Flusser, "Restoration of color images degraded by space-variant motion blur," in *Proc. Int. Conf. on Computer Analysis of Images and Patterns*, 2007, submitted.

A paper covering space-variant restoration of out-of-focus images (Chapters 4, 6 and 8) with applications in microscopy is being prepared for publication as [19]

- M. Šorel and J. Flusser, "Restoration of out-of-focus images with applications in microscopy," *J. Opt. Soc. Am. A*, work-in-progress.

Out of the scope of this thesis, the author published [20, 21]

- M. Šorel and J. Šíma, "Robust implementation of finite automata by recurrent RBF networks," in *Proceedings of the SOFSEM, Seminar on Current Trends in Theory and Practice of Informatics, Milovy, Czech Republic*. Berlin: Springer-Verlag, LNCS 1963, 2000, pp. 431–439.

- M. Šorel and J. Šíma, "Robust RBF finite automata," *Neurocomputing*, vol. 62, pp. 93–110, 2004.

1.6 Outline of the thesis

The thesis goes further with the survey of literature (Chapter 2). In Chapter 3 we overview used notation and explain important concepts of space-variant convolution and correlation.

Chapter 4 gives basic facts about optics and models we use to describe out-of-focus blur. Similarly, Chapter 5 deals with basic facts about models describing camera motion blur.

The main result of the thesis, Algorithm I, including comments upon the practical issues associated with its implementation, is presented in Chapter 6. Two auxiliary algorithms for estimation of depth maps are described in Chapters 7 and 8.

Short Chapter 9 discusses principal limitations of the precision of depth measurements we can achieve.

To give the full picture of the behavior of the proposed algorithms, we present two groups of experiments. Chapter 10 tests numerical behavior of the algorithms under different levels of noise using simulated experiments. Experiments on real images including color images are presented in Chapter 11.

Conclusion (Chapter 12) summarizes results presented in this thesis, describes their strengths and weaknesses with respect to existing methods and indicates directions of future research and possible applications.

Finally, Appendices A and B detail proofs of mathematical propositions necessary in Algorithms I and II respectively.

Chapter 2

Literature survey

The algorithms proposed in this work fall to the categories of depth from defocus, depth from motion blur and image restoration. All these categories are covered in the following survey. The algorithms that do both restoration and depth recovery simultaneously are treated at the end of the section on image restoration. Abbreviations used in this chapter were explained in Section 1.2.

2.1 Depth from defocus

Among the first DFD results we can mention Pentland [22, 23], who used two images of a scene, only one of them out-of-focus. Ens and Lawrence [24] iteratively estimated local convolution matrix that, convolved with one of the images, produces the other image. The resulting matrix can be mapped to depth estimates.

Subbarao and Surya [1] assumed the Gaussian mask shape, approximated image function by third-order polynomial and derived an elegant expression for relative blur

$$\sigma_2^2 - \sigma_1^2 = 2\frac{\mathbf{z}_2 - \mathbf{z}_1}{\nabla^2\left(\frac{\mathbf{z}_1+\mathbf{z}_2}{2}\right)}, \tag{2.1}$$

which can be used to estimate distance. Here z_1, z_2 are near and far focused images, σ_1^2, σ_2^2 denote variances of mask shapes taken as distributions of two-dimensional random quantities and ∇^2 is the symbol for Laplacian. This method also requires knowledge of two constants α and β, describing relation between the mask variances for pairs of corresponding points in $\mathbf{z}_1$ and $\mathbf{z}_2$ by linear relation $\sigma_2 = \alpha\sigma_1 + \beta$, where α and β can be computed from camera settings. Note that this is assumed to hold analogously to the same relation between radii of blur circles (4.8), which is true according to Gaussian optics

model. See Chapter 4 for details. Note that assuming Gaussian masks we can recover the relative blur $\sigma_2^2 - \sigma_1^2$ but it is principally impossible to recover the variances absolutely if we do not known the camera settings for both images. In this context the requirement of Pentland that one of the images must be in focus can be understood as the prior knowledge that $\sigma_1 = 0$. We anticipate that in our algorithm a generalization of this extremely fast method serves as an alternative for reasonable initial estimate of the depth map.

All the early methods are based on the assumption that the amount of defocus is constant over some fixed neighborhood of each image point. The choice of window has naturally a profound impact on results. Xiong and Schafer [25] addressed the problem of analyzing and eliminating the influence of finite-width windows using moment and hypergeometric filters. Their method requires a large number of filters to cover enough frequency bands and as a consequence can be markedly more time-consuming then [1]. Note that application of a single filter is a synonym for making convolution in this context.

Watanabe and Nayar [26] proposed another filter-based method but unlike [25], they used a small number of broadband operators, resulting in much faster (probably less precise but still much more precise then [1]) algorithm. Nonlinear optimization is used to compute the filter kernels. As a byproduct, they get a depth confidence measure. Their method assumes pillbox mask shape.

Deschênes et al. [27] derived a filter for simultaneous estimation of defocus and shift (disparity).

2.2 Depth from motion blur

Compared to defocus, there is markedly less literature related to to recovery of depth from motion blur. In a very simple form, this idea appears in [28] for space-invariant case, assuming two images, only one of them blurred. Besides, we can mention several papers on the extraction of OF information using motion blur, either from just one image [29, 30] or from more images of the same scene taken with different camera parameters [28, 31]. Wang and Liang [32] proposed a method to recover depth from both motion blur and defocus. Again, all these methods are based on the assumption that for each point of the image there exist a neighborhood of fixed size, where the character of blur remains approximately constant.

2.3 Image restoration

Now, we move our attention to the restoration of blurred images. First we mention non-blind methods, which are simpler and more straightforward than the blind ones. Then, we will focus on blind methods, many of which incorporate some non-blind methods as well as DFD algorithms and algorithms for estimation of depth from motion blur as their part.

There exist many methods for restoration of single image degraded by known space-invariant blur, so called space-invariant *single channel* (*SC*) *non-blind* restoration techniques. A good survey paper is [33]. Many of them are formulated as linear problems that can be efficiently solved by elementary numerical algorithms, some others including important *anisotropic regularization* techniques [34, 35, 36] can be reduced to a sequence of linear problems. Extension of these methods to MC case is straightforward and many of them can be used for space-variant restoration as well because they treat convolution as linear operator that is sufficiently general to include space-variant PSF. Note that in this case the corresponding matrix is no longer block-Toeplitz and we cannot take advantage of fast Fourier transform to speed up the computation. One exception is the case, when we know the PSF on a grid of image positions and the PSF is computed by linear interpolation in the rest of the image [37]. An application of non-blind restoration in conjunction with the extraction of OF for motion deblurring can be found in [13].

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

Although a number of SC blind deconvolution algorithms were proposed [38, 39, 40, 41, 42, 43] their use is very limited even in space-invariant case because of a severe lack of the information contained in just one image. They work only in some special cases when it is possible to incorporate some prior knowledge about the original image, such as uniformly illuminated background in case of astronomical images. Recently, a promising approach appeared employing statistics of the distribution of gradients in natural images [44].

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 and indeed, there exist a number of methods successfully solving this issue [45, 46, 47, 2]. Note here that we use method [2] as part of the proposed Algorithm I.

In connection with our algorithms we are interested mainly in the space-variant case, when the PSF can change from point to point.

If there are no constraints on the shape of 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. Within this group, the method [48] is based on Kalman filtering, [49] on a variant of expectation maximization (EM) algorithm and [50] on regular null patterns in image spectrum. Note that all these methods are of very limited application and we can expect that they fail whenever a depth discontinuity appears. Unfortunately, it is typically the case of both out-of-focus and camera motion blur.

If we know the type of space-variant blur, as in the case of motion blur or defocus, the number of unknowns is significantly reduced. The vast majority of algorithms still assumes that the PSF is locally space-invariant [51].

In the introduction we said that there are two important types of blur we are concerned with, camera motion blur and defocus, with the property that the PSF does not change arbitrarily but is a function of depth. If we have two images of the same scene taken with different camera settings it gives us additional information that makes the problem of space-variant restoration tractable. We have seen that there exist a couple of DFD, OF and depth from motion algorithms. A natural approach is to take the depth map or OF information and use it together with the knowledge of the relation between depth/OF and PSF for non-blind restoration [37]. In this way restoration is closely related to the depth recovery algorithms.

An alternative approach is to do both depth recovery and restoration simultaneously, using variational methods.

For defocus, Rajagopalan and Chaudhuri [7] proposed a variational method based on Markov random fields, assuming two images and Gaussian PSF. To minimize the corresponding cost functional they used simulated annealing which has a nice property of global convergence, but is too slow to be used in practice. To initialize the minimization, they used the filter-based depth estimation method [1]. Later, they extended the algorithm for combination of defocus and stereo [52].

Another view of the same minimization problem was given by Favaro et al. [8] who modeled defocusation as anisotropic diffusion process and solved the corresponding partial differential equation. In [9] they incorporated motion blur into the model as well. The motion blur was modeled by Gaussian PSF, which was locally deformed according to the direction and the extent of blur. This approach can be adequate for small blurs.

To bypass the deblurring phase of minimization, Favaro and Soatto [6] derived projection operators that yield directly the minimum value of the cost functional for given depth map. On terms of local invariance of the blur and finite set of possible depths, they got an algorithm that can be used for arbitrary known PSF. If the PSF is not know, the method is able to derive filters from a set of sample images. Unlike the filter-based DFD algorithms described in Section 2.1, it requires computation of a convolution for each considered depth of scene.

Chapter 3

Notation

This chapter is a short review of notation used in the thesis. We start with two operators that can be seen as generalization of convolution and correlation to space-variant situations. Then we explain conventions used to name variables and in the end we give a table of used variables and mathematical expressions with a concise description of their meaning.

Convolutions have a prominent role in image processing as they are able to model most space-invariant image degradations, including out-of-focus and motion blur. Moreover, convolution satisfies well known convolution theorem that often makes computations faster.

Convolution can be viewed as *spreading* (distribution, diffusion) of energy of each pixel *over* the neighboring points with weights given by the *convolution mask*[1]. It explains why the continuous counterpart of the convolution mask is called *point spread function* (*PSF*).

In case of general space-variant linear degradation according to (1.1), 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*. Precisely, we can define it as

$$\mathbf{u} *_v \mathbf{h}\ [x, y] \quad = \quad \int_\Omega \mathbf{u}(x-s, y-t)\mathbf{h}(x-s, y-t; s, t)\ dsdt. \qquad (3.1)$$

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

[1] In image processing convolution is often (in somewhat confusing way) described as *gathering* of energy *from* neighboring pixels with weights given by the convolution mask turned around its center, i. e. computing of dot product. The rotation of mask is necessary to get this correlation-like description in agreement with the "natural" definition.

Similarly, with a slight abuse of terminology, we can define *space-variant correlation* as

$$\mathbf{u} \circledast_v \mathbf{h}\ [x,y] \quad = \quad \int_\Omega \mathbf{u}(x-s, y-t)\mathbf{h}(x,y;-s,-t)\ dsdt. \qquad (3.2)$$

We can imagine this operator as putting space-varying PSF to all positions in the image and computing dot product. It can be shown that for real $\mathbf{h}$ space-variant correlation is the adjoint operator to space-variant convolution with the same PSF[2].

Note that in the space-invariant case, when $\mathbf{h}(x,y;s,t) = \mathbf{h}(s,t)$, the space-variant convolution gives exactly the standard convolution and the space-variant correlation gives standard correlation without normalization (which is again conjugate transpose to convolution with the same mask).

As we will see later, both definitions are useful and the introduced notation results in surprisingly neat expressions for the gradient of the used cost functional. In the following chapter, we will show how space-variant convolution can be naturally used to describe space-variant degradations produced by camera lenses.

In the description of the algorithms and in all mathematical formulas we use continuous (functional) notation. It means that images and depth maps are treated as two-dimensional functions and convolutions are expressed using integrals. The conversion to finite-dimensional form used in actual implementation is nevertheless straightforward. Functions and integrals correspond to matrices and finite sums of matrix elements respectively. L_2 norm turns into Frobenius matrix norm and derivatives become symmetrical differences in the common way.

We should also mention the notation used in integral limits. As a rule we integrate over some finite subset of $\mathbf{R}^2$. To distinguish between two most frequent cases at the first sight, we use $\mathcal{D}$ for integration over the whole image and Ω for integration over some finite neighborhood corresponding to PSF support.

Bold letters will denote functions (matrices), for example $\mathbf{r}(x,y)$ denotes radius of blur circle r corresponding to point (x,y).

[2]It should be no surprise as columns of the matrix corresponding to convolution operator with a mask tell us where the corresponding points spread and rows from which points information for the given point comes. We work with real numbers so the adjoint operator corresponds to simple transposition of the matrix.

P	the number of blurred images we process
$\mathbf{z}_1, \ldots, \mathbf{z}_P$	blurred images we get at the input of our algorithms
$\mathbf{u}$	ideal (sharp) image we wish to compute
$\mathbf{w}$	depth map or some convenient representation of the depth map we wish to compute
$h_p(w)$	operator giving the space-invariant PSF corresponding to the distance represented by scalar w (for input image p)
$h_p(\mathbf{w})$	operator returning space-variant PSF $\mathbf{h}(x, y; s, t) = h_p(\mathbf{w}(x, y))[s, t]$
$\frac{\partial h_p(w)}{\partial w}$	derivative of PSF with respect to the value of the depth representation
$\frac{\partial h_p(\mathbf{w})}{\partial \mathbf{w}}$	analogously to $h_p(\mathbf{w})$, gives space-variant PSF $\mathbf{h}(x, y; s, t) = \frac{\partial h_p(\mathbf{w}(x,y))}{\partial w}[s, t]$
$*_v$	space-variant convolution (subscript v means variant to distinguish from ordinary convolution)
$\circledast_v$	space-variant correlation (adjoint operator to space-variant convolution with the same PSF)
$\|.\|$	L_2 norm for functions or corresponding Frobenius norm for matrices
$\int_{\mathcal{D}}$	integration over the whole image
$\int_{\Omega}$	integration over some finite neighborhood, usually corresponding to the support of a PSF

Chapter 4

Out-of-focus blur

This chapter is primarily concerned with description of degradations produced by optical lens systems and relation of the involved PSF to three-dimensional structure of observed scene, position of the object in the field of view and to camera settings. We begin by description of Gaussian model of optical systems (Fig. 4.1) and corresponding PSFs, then proceed to more realistic models and end up with the case of general axially-symmetric optical system.

4.1 Gaussian optics

Image processing applications widely use a simple model based on *Gaussian (paraxial) optics* which follows the laws of ideal image formation[1] described in the next paragraph. The name paraxial suggests that in reality it is valid only in a region close to the optical axis.

Note that we will refer to *image space* and *object space* meaning the space behind and in front of the lens, respectively. The basic postulate of *ideal image formation* is that all rays through any point P in object space must pass through one point P' in image space and the coordinates (x, y) of P are proportional to the coordinates (x', y') of P'. In other words, any figure on a plane perpendicular to the optical axis is perfectly imaged as a geometrically similar figure on some plane in image space that is also perpendicular to the optical axis. The properties of the ideal optical system are completely fixed by four *cardinal points*—two *principal points* and two *foci*. In other words, we can use these four points to find the position and size of the image of any object. The basic equation connecting the distance l of an object from

[1] Concept formalized by James Clerk Maxwell (1856) without invoking any physical image-forming mechanism [53].

the *front principal plane*, i. e. the plane perpendicular to the optical axis at the front principal point, and the distance l' of its image from the *rear principal plane*, i. e. the plane perpendicular to the axis passing through the rear principal point, is

$$\frac{1}{f} = \frac{1}{l} + \frac{1}{l'}, \tag{4.1}$$

where f is *focal length*, i. e. the distance of the focus from the principal point. In theory there are two focal lengths, front and rear, but if media in front of and behind the lens have the same index of refraction, as is usually true, the lengths are the same [53].

Moreover, the principal planes (and so principal points) are usually assumed to coincide, implying that *depth* (distance along the optical axis) in object and image spaces is measured from the same plane and the whole system is given by just two points.

In real optical systems, there is also a roughly circular *aperture*, the hole formed by the blades that limit the pencils of rays propagating through the lens (rays emanate within solid angle subtended by the aperture). Its size is usually specified by *f-number*

$$f_{\#} = \frac{f}{2\rho}, \tag{4.2}$$

where ρ is radius of aperture hole. A nice property of f-number is that it describes illumination of film or image sensor independently of focal length. Besides it controls depth of field. The aperture is usually assumed to be placed at the principal plane, i. e. somewhere inside the lens. It should be noted that this arrangement has an unpleasant property that magnification varies with focus settings. If we work with more images of the same scene focused at different distances, it results in more complicated algorithms with precision deteriorated either by misregistration of corresponding points or by errors introduced by resampling and interpolation[2]. Note that Algorithms I and III solve this issue to some extent, but at the cost of higher memory requirements.

[2]These problems can be eliminated using so called front telecentric optics, i. e. optics with aperture placed at the front focal plane. Then all principal rays (rays through principal point) become parallel to the optical axis behind the lens and consequently magnification remains constant as the sensor plane is displaced [26]. Unfortunately most conventional lenses are not telecentric.

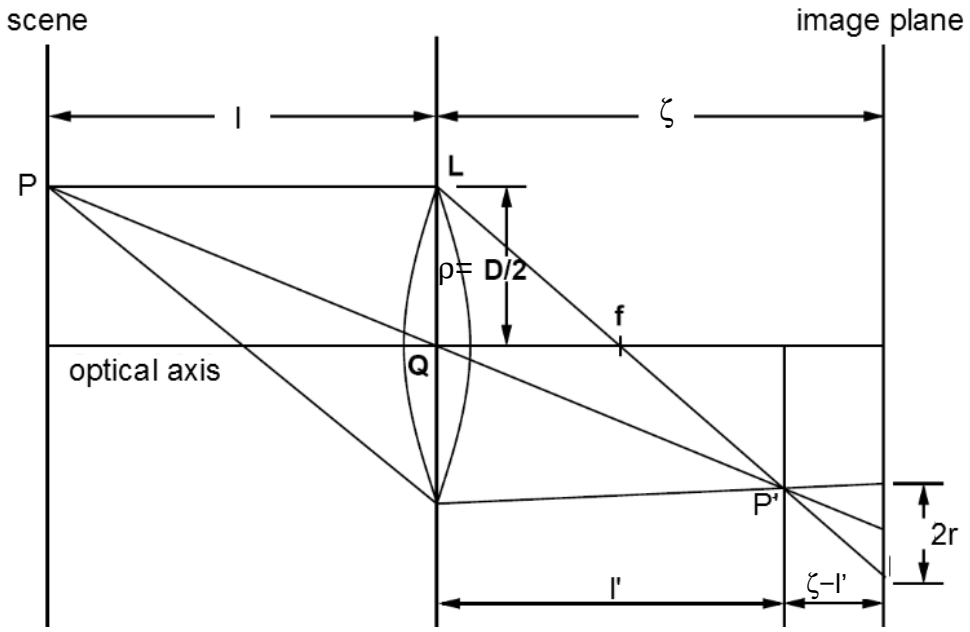


Figure 4.1: Lens system and formation of blur circle (modified from [1]).

In the introduction we mentioned that the degradation produced by an optical system can be described by linear relation (1.1). Using the notation for space-variant convolution (3.1) we can write (1.1) as

$$\mathbf{z} = \mathbf{u} *_v \mathbf{h} + \mathbf{n}. \tag{4.3}$$

In the following sections we show several models that can be used for the PSF $\mathbf{h}$ and its relation to the distance of objects from camera.

4.2 PSF in case of Gaussian optics

We consider the Gaussian optics model described in the previous paragraphs. If the aperture is assumed to be circular, graph of PSF has a cylindrical shape usually called *pillbox* in literature. When we describe the appearance of the PSF in the image (or photograph), we speak about *blur circle* or *circle of confusion*. It can be easily seen from similarity of triangles (see Fig. 4.1)

that its radius for arbitrary point in the distance l

$$
\begin{aligned}
r &= \rho\frac{l'-\zeta}{l'} = \rho\zeta\left(\frac{1}{\zeta}+\frac{1}{l}-\frac{1}{f}\right) && (4.4)\\
&= \rho\zeta\left(\frac{1}{l}-\frac{1}{l_s}\right) && (4.5)\\
&= \frac{1}{l}\rho\zeta+\rho\zeta\left(\frac{1}{\zeta}-\frac{1}{f}\right), && (4.6)
\end{aligned}
$$

where ρ is the aperture radius, ζ is the distance of the image plane from the lens and l_s distance of the plane of focus (where objects are sharp) that can be computed from ζ using (4.1).

Notice the importance of inverse distances in these expressions. The expression (4.5) tells us that radius r of blur circle grows proportionally to the difference between inverse distances of the object and of the plane of focus[3]. Expression (4.6) can be restated that r is a linear function of the inverse of the distance l. Other quantities ρ, ζ and f depend only on the camera settings and are constant for one image.

Thus, PSF can be written as

$$
\mathbf{h}(x,y;s,t) = \begin{cases} \frac{1}{\pi\mathbf{r}^2(x,y)}, & \text{for } s^2+t^2 \le \mathbf{r}^2(x,y),\\ 0, & \text{otherwise,}\end{cases} \qquad (4.7)
$$

where $\mathbf{r}(x,y)$ denotes the radius r of the blur circle corresponding to the distance of point (x,y) given by relations (4.4)-(4.6). Given camera parameters f, ζ and ρ, matrix $\mathbf{r}$ is readily only alternative representation of depth map.

Now, suppose we have another image of the same scene, registered with the first image and taken with different camera settings. As the distance is the same for all pairs of points corresponding to the same part of the scene, inverse distance $1/l$ can be eliminated from (4.6) and we get linear relation between the radii of blur circles in the first and the second image

$$
\begin{aligned}
\mathbf{r}_2(x,y) &= \alpha\mathbf{r}_1(x,y)+\beta, \text{ where} && (4.8)\\
\alpha &= \frac{\rho_2}{\rho_1}\frac{\zeta_2}{\zeta_1}, && (4.9)\\
\beta &= \rho_2\zeta_2\left(\frac{1}{\zeta_2}-\frac{1}{\zeta_1}+\frac{1}{f_1}-\frac{1}{f_2}\right). && (4.10)
\end{aligned}
$$

[3]An obvious consequence is a photographic rule to focus on harmonic average of the distances of the nearest and farthest object we want to have in focus. As it does not sound very practical, textbooks give a rule of thumb to focus to one-third of the distance. Actually it holds only if the farthest object is twice as far as the nearest one.

The proposed algorithm assumes α and β are known. Obviously, if we take both images with the same camera settings except of aperture, i. e. $f_1 = f_2$ and $\zeta_1 = \zeta_2$, we get $\beta = 0$ and α equal to the ratio of f-numbers defined by (4.2).

In reality the aperture is not a circle but shape (often polygon) with as many sides as there are blades. Note that at full aperture, where blades are completely released, the diaphragm plays no part and the support of the PSF is really circular. Still assuming Gaussian optics, the aperture projects on the image plane according to Fig. 4.1, changing its scale the same way as for circular aperture, i. e. in the ratio

$$w = \frac{l' - \zeta}{l'} = \zeta \left(\frac{1}{l} - \frac{1}{l_s} \right) = \frac{1}{l}\zeta + \zeta \left(\frac{1}{\zeta} - \frac{1}{f} \right), \tag{4.11}$$

with a consequence that

$$\mathbf{h}(x, y; s, t) = \frac{1}{\mathbf{w}^2(x, y)} \mathbf{h}(\frac{s}{\mathbf{w}(x, y)}, \frac{t}{\mathbf{w}(x, y)}), \tag{4.12}$$

where $\mathbf{h}(s, t)$ is the shape of the aperture. The mask keeps the unit sum of $\mathbf{h}$ thanks to the normalization factor $1/\mathbf{w}^2$. Comparing (4.11) with (4.4)-(4.6) it can be easily seen that blur circle (4.7) is a special case of (4.12) for $\mathbf{w}(x, y) = \mathbf{r}(x, y)/\rho$ and

$$\mathbf{h}(s, t) = \begin{cases} \frac{1}{\pi\rho^2}, & \text{for } s^2 + t^2 \leq \rho^2, \\ 0, & \text{otherwise.} \end{cases} \tag{4.13}$$

On the other hand, using (4.11) for two images yields

$$\mathbf{w}_2(x, y) = \alpha' \mathbf{w}_1(x, y) + \beta', \text{ where} \tag{4.14}$$

$$\alpha' = \frac{\zeta_2}{\zeta_1}, \tag{4.15}$$

$$\beta' = \zeta_2 (\frac{1}{\zeta_2} - \frac{1}{\zeta_1} + \frac{1}{f_1} - \frac{1}{f_2}). \tag{4.16}$$

Notice that if the two images differ only in the aperture, then $\mathbf{w}_2 = \mathbf{w}_1$.

4.3 Approximation of PSF by two-dimensional Gaussian function

In practice, due to lens aberrations and diffraction effects, PSF will be a roughly circular blob, with brightness falling off gradually rather than

sharply. Therefore, most algorithms use two-dimensional Gaussian function

$$\frac{1}{2\pi\sigma^2}e^{-\frac{s^2+t^2}{2\sigma^2}} \tag{4.17}$$

instead of pure pillbox shape. Notice that it can be written in the form of (4.12) for

$$\mathbf{h}(s,t) = \frac{1}{2\pi}e^{-\frac{s^2+t^2}{2}}$$

with $w = \sigma$ as well. To map the variance σ to real depth, [1] propose to use relation $\sigma = r/\sqrt{2}$ together with (4.4) with the exception of very small radii. Our experiments showed that it is often more precise to state the relation between σ and r more generally as $\sigma = kr$, where k is a constant found by camera calibration (for the lenses and settings we tested k varied around 1.2). Then analogously to (4.8) and (4.14)

$$\sigma_2 = \alpha'\sigma_1 + \beta', \quad \alpha', \beta' \in \mathbf{R}, \tag{4.18}$$

where $\alpha' = \alpha$, $\beta' = k\beta$. Again, if we change only the aperture then $\beta' = 0$ and α' equals the ratio of f-numbers.

Corresponding PSF can be written as

$$\mathbf{h}(x,y;s,t) = \frac{1}{2\pi k^2\mathbf{r}^2(x,y)}e^{-\frac{s^2+t^2}{2k^2\mathbf{r}^2(x,y)}} \ . \tag{4.19}$$

If possible we can calibrate the whole (as a rule monotonous) relation between σ and distance (or its representation) and consequently between σ_1 and σ_2.

In all cases, to use Gaussian efficiently, we need a reasonable size of its support. Fortunately Gaussian falls off quite quickly to zero and it is usually sufficient to truncate it by a circular window of radius 3σ or 4σ. Moreover, any real out-of-focus PSF has finite support anyway.

4.4 General form of PSF for axially-symmetric optical systems

In case of high-quality optics, pillbox and Gaussian shapes can give satisfactory results as the model fits well with reality. For less well corrected optical systems rays can be aberrated from their ideal paths to such an extent that it results in very irregular PSFs. In general, aberrations depend on the distance of the scene from camera, position in the image and on the camera settings f, ζ and ρ. As a rule, the lenses are well corrected in the image center, but

towards the edges of the image PSF may become completely asymmetrical and look for example like in Fig. 10.2(a).

Common lenses are usually axially-symmetric. For such a system, since it must behave independently of its rotation about the optical axis, it is easily seen that

1. in the image center, PSF is radially symmetric,
2. for the other points, PSF is bilaterally symmetric about the line passing through the center of the image and the respective point,
3. for points of the same distance from the image center and corresponding to objects of the same depth, PSFs have the same shape, but they are rotated about the angle given by angular difference of their position with respect to the image center.

The second and third points can be written as

$$\mathbf{h}(x, y; s, t) = \mathbf{h}\left(0, |(x, y)|; \frac{|(-t, s)(x, y)^T|}{|(x, y)|}, \frac{(s, t)(x, y)^T}{|(x, y)|}\right). \tag{4.20}$$

The dot products are simply sine and cosine of the angle of rotation according to the third point and the absolute value in the numerator of the third term is the half of PSF which is sufficient to specify thanks to the bilateral symmetry.

In most cases, it is impossible to derive an explicit expression for PSF of given optical system. On the other hand, it is relatively easy to get it by a raytracing algorithm. Above mentioned properties of axially-symmetric optical system can be used to save memory as we need not to store PSFs for all image coordinates but only for every distance from the image center. Naturally, it makes the algorithms more time consuming as we need to rotate the PSFs every time they are used.

Finally, we should mention the existence of other optical phenomenons that to some extent influence the real PSF but that can be neglected for the purpose of this work.

Diffraction is a wave phenomenon which makes a beam of parallel light passing through a circular aperture spread out a little. The smaller the aperture, the more the spreading. Since we are interested in situations of small depth of focus, diffraction takes no much effect and we can neglect it.

It is well known that the refractive index varies with wavelength or frequency of light. This so called dispersion is a source of chromatic aberrations in optical systems [53]. However, for algorithms working with intensity images it is probably impossible to take them into account because we have no

information about spectral content of the images and in addition their influence is rather limited as the spectral sensitivity of one channel is narrow. Color images are treated only marginally in this work.

4.5 Summary

In this chapter, we described several shapes of PSF that can be used to model out-of-focus blur. Gaussian and pillbox shapes are adequate for good quality lenses or in the proximity of the image center, where the optical aberrations are usually well corrected. A more precise approach is to consider optical aberrations. However, an issue arises in this case that aberrations must be described for the whole range of possible focal lengths, apertures and planes of focus.

Chapter 5

Camera motion blur

In the previous chapters we have already mentioned that camera motion blur can be modeled by convolution with a space-variant PSF. To use this model in the proposed algorithms, we need to express the PSF as a function of the camera motion and the depth of the scene.

Note that we follow the convention that the z-axis coincides with the optical axis and the x and y axes are considered parallel to horizontal and vertical axes of the image sensor. The origin of the coordinate system is placed at the front principal point of the optical system, which corresponds to the optical center of the pinhole camera.

5.1 General camera motion

In the general case, the PSF can be computed from the formula for *velocity field* [54, 8] that gives apparent velocity of the scene for the point (x, y) of the image at time instant τ as

$$\begin{aligned}\mathbf{v}(x,y,\tau) = \frac{1}{\mathbf{l}(x,y,\tau)} \begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \end{bmatrix} T(\tau) + \\ \begin{bmatrix} xy & -1-x^2 & y \\ 1+y^2 & -xy & -x \end{bmatrix} \Omega(\tau),\end{aligned} \tag{5.1}$$

where $\mathbf{l}(x, y, \tau)$ is the depth corresponding to point (x, y) and $\Omega(\tau)$ and $T(\tau)$ are three-dimensional 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 of $\Omega(\tau)$, are in focal length units.

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 two-dimensional space-variant PSF can be expressed as

$$\mathbf{h}(x, y; s, t) = \int \delta(s - \bar{x}(x, y, \tau), t - \bar{y}(x, y, \tau))d\tau, \tag{5.2}$$

where δ is two-dimensional Dirac delta function.

In the case of general camera motion, the solution of the restoration problem can be difficult, as discussed in Section 6.7. Therefore, it may be reasonable to consider some limited class of motions, where the PSF can be expressed explicitly.

Arbitrary camera motion can be decomposed into two types of translations and two types of rotations. In the following sections we discuss the influence of these motion components on the PSF they produce.

For the purposes of this thesis, the most important case is translation in one plane perpendicular to the optical axis, which will be treated in detail in Section 5.3. Rotations (Section 5.2) and translations in the direction of the optical axis (Section 5.4) will be described briefly without explicit formulas for the corresponding PSF.

5.2 Rotation

First we describe the rotational movements, which are simpler in the sense that the blur they produce does not depend on the distance of the scene from camera. Therefore, if we track rotational camera motion by an inertial sensor, we are able to assign a PSF to each image pixel and restore the sharp image from just one single image by one of non-blind restoration methods. It is well known that any three-dimensional rotation can be decomposed into rotations about three independent axes going through the center of rotation—in our case, without loss of generality, about the axes of the coordinate system.

Rotation of the camera about the optical axis (rolling) makes the points in the image move along concentric circles centered in the center of the image. Consequently, the PSF is uniquely determined by the course of angular velocity of the camera and the image coordinates (x, y). The extent of the blur increases linearly with the distance from the image center.

The blur caused by the rotation about any axis lying in the front principal plane and going through the optical center (panning, tilting) is influenced by perspective distortion. In the proximity of the image center the PSF

is almost space-invariant but as we move away from the image center, we must compensate for the dilation/contraction in the direction of the axis of rotation.

The PSF for combination of rotation (angular motion) with defocus, including optical aberrations, was described recently in [55].

5.3 Translation in one plane perpendicular to the optical axis

Now, we proceed to the translational motion, which depends on the distance of the scene from the camera. Again, it can be decomposed into translations in the directions of the three axes.

If the camera moves in one plane perpendicular to the optical axis without any rotations ($\Omega = (0, 0, 0)$, $T(3) = 0$), which is the case assumed in the basic version of Algorithms I and III, then the magnitude of the velocity vector is proportional to the inverse depth. Moreover, depth for the given part of the scene does not change during such a motion and consequently the PSF simply decreases its scale proportionally to the depth, namely

$$\mathbf{h}(x, y; s, t) = \mathbf{l}^2(x, y)\mathbf{h}_0(s\mathbf{l}(x, y), t\mathbf{l}(x, y)), \tag{5.3}$$

where "prototype" PSF $\mathbf{h}_0(s, t)$ corresponds to the path covered by the camera during the time when the shutter is open. Depth is again given in focal length units.

Equation (5.3) implies that if we know PSF for an arbitrary fixed distance from camera, we can compute it for any other distance by simple stretching in the ratio of the distances.

Interestingly, PSF (5.3) is the same formula that holds for most models of out-of-focus blur described in Chapter 4 with $\mathbf{w}$ being inverse depth

$$\mathbf{w}(x, y) = 1/\mathbf{l}(x, y). \tag{5.4}$$

The only difference is the shape of the "prototype" mask $\mathbf{h}_0$.

We should mention a special case, steady motion of the camera in a direction perpendicular to the optical axis. Then, it is well known that the PSF is space-invariant one-dimensional rectangular impulse in the direction of camera motion and its length

$$\mathbf{d}(x, y) = \frac{b}{\mathbf{l}(x, y)}, \tag{5.5}$$

where b is the path covered by camera during the capture process. If we realize that $\mathbf{l}$ is given in focal length units, it is not surprising that equation (5.5) is exactly the formula for stereo disparity, where b is the length of the the baseline.

5.4 Translation in the direction of optical axis

Finally, we come to the translational motion in the direction of the optical axis. It it the most complicated motion component in the sense that the PSF depends on both the distance from the camera and position in the field of view. As the camera moves towards the scene, the image increases its scale but the extent of this scale change depends on the distance from camera. In other words, image points move outwards/inwards along lines emanating from the image center but the speed of their motion depends on the depth.

5.5 Summary

In this chapter, we discussed relation between PSF and several types of camera motions.

For our purposes, we need mainly Section 5.3, describing translational motion in one plane perpendicular to the optical axis. It is exactly the model with which the basic versions of Algorithms I and III work. The principal advantage of this assumption is that the corresponding PSF is a function of only depth and not of the position in the field of view. This model can be valid in industrial applications with camera mounted on vibrating or moving objects.

Possibility of restoration in the case of completely general camera motion will be discussed in Section 6.7.

Chapter 6

Restoration of space-variantly blurred images (Algorithm I)

In this chapter we describe the main result presented in this thesis—an algorithm for restoration of images blurred by space-variant out-of-focus or camera motion blur.

Let us denote the blurred images at the input as $\mathbf{z}_p$. For out-of-focus blur, the images must be taken from the same place with different camera parameters. In case of camera motion blur, the camera parameters are supposed to be the same and the camera motion differs. In the following description of the algorithm, the camera motion is limited to translational motion in one plane perpendicular to the optical axis. This limitation includes not only the camera motion during the capture of one image but also the change of camera position between the images, which ensures that the depth map is common for all the images. We should stress that in this case we need to know neither how the camera moves nor camera parameters. The extension to general camera motion is discussed in Section 6.7. Finally, we assume known relation between distance and PSF according to models from Chapter 4 for out-of-focus blur and from Chapter 5 for motion blur.

Recall that the process of blurring can be modeled using space-variant convolution (1.1), which can be written in simplified form as (4.3) using notation (3.1). The proposed algorithm can be described as minimization of cost functional

$$E(\mathbf{u}, \mathbf{w}) = \frac{1}{2} \sum_{p=1}^{P} \|\mathbf{u} *_v h_p(\mathbf{w}) - \mathbf{z}_p\|^2 + \lambda_u Q(\mathbf{u}) + \lambda_w R(\mathbf{w}) \qquad (6.1)$$

with respect to sharp image $\mathbf{u}$ and depth map represented by $\mathbf{w}$. The value of $\mathbf{w}(x, y)$ does not give directly the distance related to pixel (x, y) in the

common way but it is a convenient linear function of the reciprocal of the distance from reasons explained later in this chapter. As will be discussed later, a good choice is *inverse depth* $\mathbf{w}(x,y) = \frac{1}{\mathbf{l}(x,y)}$. Recall that the depth map is common for all the images in the cases we consider now.

The first term of (6.1), called *error term* in the rest of this thesis, is a measure of difference between the inputs, i. e. blurred images $\mathbf{z}_p$, and the image $\mathbf{u}$ blurred according to chosen blurring model using information about depth of scene $\mathbf{w}$. The size of the difference is measured by L_2 norm $\|.\|$, which corresponds to Frobenius matrix norm in the actual implementation. The inner part of the error term,

$$\mathbf{e}_p = \mathbf{u} *_v h_p(\mathbf{w}) - \mathbf{z}_p, \tag{6.2}$$

is nothing else than the matrix of error at the individual points of the image p. The error term can be written as $\Phi = \sum_{p=1}^{P} \Phi_p$, where $\Phi_p = \frac{1}{2}\|\mathbf{e}_p\|^2 = \frac{1}{2}\int_{\mathcal{D}} \mathbf{e}_p^2(x,y)$.

For image p, the operator $h_p(\mathbf{w})$ gives space-variant PSF corresponding to depth map represented by $\mathbf{w}$ according to chosen blurring model. Its space-variant convolution with the sharp image $\mathbf{u}$ models the process of blurring.

In case of defocus, h_p is unambiguously given as a function (pillbox or Gaussian) of depth and camera parameters, with the exception of aberrated optics, where the PSF must be stored in a way for all combinations of camera parameters, depths of the scene and positions in the field of view.

In the considered case of camera motion in one plane perpendicular to the optical axis, relation (5.3) implies that it is sufficient to known the PSF for one fixed depth and h_p can be computed for an arbitrary depth using this relation. For this purpose, we can apply space-invariant blind restoration method [2] on a flat part of the scene, where the blur is approximately space-invariant. Besides the restored sections, this method provides also an estimate of masks (PSFs). As we usually do not know the real depth for this part of the scene, the depth map we compute is correct only up to a scale factor. This is however sufficient, since our primary goal is restoration. Note that the masks incorporate the relative shift of the cameras between the images.

Regularization is a popular method to achieve satisfactory solution of problems involving inversion of ill-conditioned operators such as the convolution with space-variant mask. The role of regularization terms is to achieve well-posedness of the problem and incorporate prior knowledge about the solution [56, 57].

Thus, $Q(\mathbf{u})$ is an image regularization term which can be chosen to represent properly the expected character of the image function. For the majority

of images a good choice is total variation $Q_{TV}(\mathbf{u}) = \int |\nabla \mathbf{u}|$, proposed by Rudin et al. [34]. Tikhonov regularization term $Q(\mathbf{u}) = \int |\nabla \mathbf{u}|^2$ can be more appropriate for scenes without sharp edges, where TV regularization often results in a "blocky" look of the image. In turn, an issue with Tikhonov regularization is that it tends to smooth sharp edges. For more detailed discussion of image regularization, see [58, 33].

Similarly, we can choose convenient depth map regularization term $R(\mathbf{w})$. Contrasting the image regularization, paradoxically, the best choice for depth map is usually Tikhonov regularization. The reason is that TV regularization may cause convergence problems at steep depth edges as demonstrated in simulated experiments.

6.1 Choice of depth map representation

Now, we will discuss why we do not work directly with depth and outline more convenient depth map representations suitable for different models of blur. We have already mentioned that a good choice is an arbitrary linear function of inverse depth. We will show that in a sense all such representations are equivalent. Note that the algorithm can be implemented independently of any particular representation.

In theory, we could always use directly the real depth. However, it has several major drawbacks. First, we need to know exactly all the camera settings (f, ζ, ρ). We will show that it is not always necessary using other representations if our goal is mainly restoration of the sharp image. Another issue with the direct use of distance is that it tends to regularize the depth map too heavily at the edges between near and distant objects which can result in slight defocus of distant objects. Finally, non-linear dependence on distance results in more complicated formulas for derivatives of functional (6.1).

If we look at the considered models of out-of-focus and camera motion blur, we can see that in all the cases PSF scales linearly with the inverse of the distance. Note that while it is an inherent property of out-of-focus blur, for motion blur it holds only when the camera motion is limited to translation. If we take more images of the same scene, it holds for all of them and therefore at corresponding image locations the size of PSF in one image is a linear function of the size of PSF in another image. In other words, choosing any representation linear with respect to inverse depth, that is $w = \gamma/l + \delta$, PSF in arbitrary channel scales linearly with this representation. We can also imagine that PSF is now given as a function of the size of its support.

Using this type of representation, depth map regularization terms will reg-

ularize a quantity proportional to the extent of blur. If we consider Tikhonov and TV regularization terms, all the representations are equivalent with respect to the regularization up to a multiplicative constant. Indeed, if we change representation, it is sufficient to multiply λ_w by the ratio of γ's for R_{TV} and squared ratio of γ's for R_2 to get the same results.

In case of pillbox out-of-focus blur a natural choice of depth map representation is the radius of blur circle according to (4.4)-(4.6) for one of the images. Without loss of generality, let it be the first image. We get linear relation (4.8) that links PSF in the other images to the PSF in the first image. If we take the images with the same camera settings except for the aperture, i. e. $\beta = 0$, we need to know just one parameter α equal to the ratio of f-numbers. It can help us in situations when we have only Exif[1] data produced by present-day digital cameras that usually contain only f-numbers and rather unprecise focal lengths but no information where the camera was focused[2]. Thus, the algorithm actually minimizes over the extent of blur instead of over the distance and the regularization is also applied at this quantity.

Interestingly, we can use similar representation even if we do not limit ourselves to the pillbox PSF. If we consider non-circular aperture according to (4.12) or Gaussian function (4.19), we can represent distance by the ratio $\mathbf{w}$ given by (4.11). Again we have a linear relation between representations (4.14) or (4.18) respectively.

In case of blur due to the translational camera motion in one plane perpendicular to the optical axis, the depth is naturally represented by the ratio of the depth of the part of the scene where the PSF is known and the real depth as mentioned in the description of h_p above. The PSF for arbitrary depth is then computed using (5.3).

If we consider both out-of-focus blur and camera motion blur simultaneously, we can represent distance by $1/l$. In this mixed case we need all three camera parameters.

[1] *Exchangeable image file format* is a specification for the image file format used by digital cameras. The specification uses existing file formats with the addition of specific metadata tags (see http://en.wikipedia.org/wiki/Exif).

[2] One exception are professional Canon cameras with some newer lenses providing focusing information necessary for ETTL-II flash systems. Still, however, precision of provided depth information is principally limited by relations discussed in Chapter 9.

6.2 Gradient of the cost functional

In theory, to minimize the cost functional (6.1), we could apply simulated annealing [7], which guarantees global convergence. In practice however it would be prohibitively slow. For efficient minimization, we need to know at least the gradient[3] of the functional. Readily 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 [59] as

$$-\operatorname{div}\left(\frac{\kappa'(|\nabla \mathbf{u}|)}{|\nabla \mathbf{u}|}\nabla \mathbf{u}\right), \tag{6.3}$$

which for Q_2 and Q_{TV} gives

$$\frac{\partial Q_2}{\partial \mathbf{u}} = -\operatorname{div}\nabla \mathbf{u} = -\nabla^2 \mathbf{u}, \tag{6.4}$$

$$\frac{\partial Q_{TV}}{\partial \mathbf{u}} = -\operatorname{div}\left(\frac{\nabla \mathbf{u}}{|\nabla \mathbf{u}|}\right), \tag{6.5}$$

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 (6.3)-(6.5).

Gradients of the error term in image and depth map subspaces are a bit more complicated. We take advantage of the notation for space-variant correlation and get surprisingly elegant formulas.

Proposition 1. *Gradients of the error term Φ in subspaces corresponding to image $\mathbf{u}$ and depth map represented by $\mathbf{w}$ can be expressed as*

$$\frac{\partial \Phi}{\partial \mathbf{u}} = \sum_{p=1}^{P} \mathbf{e}_p \circledast_v h_p(\mathbf{w}) = \sum_{p=1}^{P} \mathbf{u} *_v h_p(\mathbf{w}) \circledast_v h_p(\mathbf{w}) - \mathbf{z}_p \circledast_v h_p(\mathbf{w}), \tag{6.6}$$

$$\frac{\partial \Phi}{\partial \mathbf{w}} = \mathbf{u} \sum_{p=1}^{P} \mathbf{e}_p \circledast_v \frac{\partial h_p(\mathbf{w})}{\partial \mathbf{w}}, \tag{6.7}$$

where $\frac{\partial h_p(\mathbf{w})}{\partial \mathbf{w}}[x, y; s, t]$ is the derivative of the mask related to image point (x, y) with respect to the value of $\mathbf{w}(x, y)$.

[3] Rigorously, if we use functional notation we should speak about Fréchet derivative instead of gradient.

Note that the formulas hold even if $h_p(\mathbf{w})$ and consequently $\frac{\partial h_p(\mathbf{w})}{\partial \mathbf{w}}$ depends also on coordinates (x, y). The proof of Proposition 1 can be found in Appendix A. Notice that the computation of gradients (6.6) and (6.7) does not take much longer than computation of the cost functional itself. They consist of only four types of matrix operations: space-variant convolution, space-variant correlation, point-wise multiplication and point-wise subtraction. The two space-variant operations itself consist of multiplications and additions. All these operations can be highly parallelized since basically the value can be computed separately in each pixel.

Here we should mention the actual implementation of $h_p(\mathbf{w})$ and $\frac{\partial h_p(\mathbf{w})}{\partial \mathbf{w}}$ we used. For defocus and the considered type of motion blur, the mask is unambiguously determined by depth, that is the space-variant PSF $h_p(\mathbf{w})$ consists of the values of $h_p(w)$ that stand for the space-invariant PSF (mask) for given w. These masks are precomputed for a sequence of values of w with constant step $\triangle_w$, i. e. we store $h_p(k\ \triangle_w)$ for an interval of indices k. During the minimization, intermediate masks are computed by linear interpolation as

$$h_p(w) = (\lceil \frac{w}{\triangle_w} \rceil - \frac{w}{\triangle_w}) h_p(\lfloor \frac{w}{\triangle_w} \rfloor\ \triangle_w) + (\frac{w}{\triangle_w} - \lfloor \frac{w}{\triangle_w} \rfloor) h_p(\lceil \frac{w}{\triangle_w} \rceil\ \triangle_w). \quad (6.8)$$

Thanks to linearity of these operations, the computation of space-variant convolution and correlation with an arbitrary mask takes only about twice more time than in the case of masks we stored.

Similarly $\frac{\partial h_p(\mathbf{w})}{\partial \mathbf{w}}$ is based on $\frac{\partial h_p(w)}{\partial w}$ which is computed from masks stored in another array generated from $h_p(k\ \triangle_w)$ 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 [60] on $h_p(w)$. Note that the derivatives could be computed analytically using (5.3) 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.

We should remark that in general, it is not evident how such an interpolation influences the convergence properties of continuous gradient-based minimization. In our experiments it has turned to be of no concern. But still if necessary, we could use interpolation of a higher order as well.

6.3 Minimization algorithm

How to find the minimum of the cost functional if we know its gradient? It is high-dimensional nonlinear problem with a huge amount of local minima, especially in the subspace corresponding to variable $\mathbf{w}$. Experiments confirmed

that the right choice of initial depth map estimate is essential to prevent the algorithm from getting trapped in a local minimum. We tested random initialization of the depth map but as a rule the minimization resulted in a number of artifacts. Constant initial choice did not work at all.

An approach that proved effective was to compute the initial estimates of the depth map using one of simpler methods based on the assumption that blur is space-invariant in a neighborhood of each pixel.

If the main requirement is speed, we can use the method presented in Chapter 7 which is a generalization of already mentioned DFD method of Subbarao and Surya [1]. It can be described by simple expressions (7.1), (7.2), (7.4) and (7.5) and can be implemented by just two convolutions, which is negligible in comparison with the time required by the following minimization. It provides noisy and inaccurate depth estimates but often proved sufficient to prevent the algorithm from getting stuck in a local minimum and it also speeds up the minimization considerably. Notice that it also does not estimate distance directly but instead it estimates convenient representation—variance of the PSF.

The necessary condition of this method is central symmetry of PSF. It implies that under certain circumstances we can use it even for strongly aberrated optics since, as we mentioned in Chapter 4 (the first point on p. 29), arbitrary axially-symmetric optical system has a rotationally symmetric PSF in the area around the image center. Of course pillbox PSF is a special case. We should remark that this method must be carefully calibrated to give reasonable results. It works when there is no much noise in the image and texture is of sufficient contrast.

Unfortunately, if the condition of symmetry is not satisfied, results can be seriously distorted. For this reason this method is unsuitable for less well corrected optics in the areas near the image border and for more complex motion blurs. For these cases, we developed another simple method described in Chapter 8 which is more general but slower. It proved to be more stable with respect to noise as well. Both methods provide either noisy and inaccurate estimates or (after smoothing) estimates with lower spatial resolution resulting in artifacts at the edges.

Let us denote the initial depth map estimate as $\mathbf{w}^0$. Now, we could use the steepest gradient method but it is well known that it suffers from slow convergence. Instead, we make use of a sort of alternating minimization (AM) algorithm [42], which basically iterates through minimizations in subspaces corresponding to unknown matrices $\mathbf{u}$ and $\mathbf{w}$. From reasons explained later, at the end of the algorithm there is another minimization over the image

subspace with different image regularization constant λ_u^f and higher number of iterations.

Algorithm I

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

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

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

Minimization of E with respect to $\mathbf{u}$ is the well known and well examined problem of non-blind restoration [33, 42]. 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 gradients method (gradients (6.4) and (6.6) are obviously linear with respect to $\mathbf{u}$). In case of Q_{TV}, matters become more complicated. However, even for this case there exist sufficiently efficient algorithms, which usually reduce the problem to a sequence of linear problems. We have chosen the approach described in [36]. Note that the authors originally designed their algorithm for denoising and space-invariant restoration problems. Nevertheless, the space-invariant convolution is treated as sufficiently general linear operator there 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 (6.1) for a fixed $\mathbf{w}^{n-1}$. We will replace the regularization term $Q = Q_{TV} = \int |\nabla(\mathbf{u})|$ by quadratic term

$$\frac{1}{2} \int_{\mathcal{D}} \frac{1}{|\nabla \mathbf{u}_m|} |\nabla \mathbf{u}|^2 + |\nabla \mathbf{u}_m|. \tag{6.9}$$

Obviously, it has the same value as Q_{TV} in $\mathbf{u}_m$. The right term of (6.9) is constant for now and consequently it does not take part in actual minimization. We have got a "close" linear problem

$$\mathbf{u}_{m+1} = \arg\min_{\mathbf{u}} \frac{1}{2} \sum_{p=1}^{P} \|\mathbf{e}_p\|^2 + \lambda_u \int_{\mathcal{D}} \frac{1}{2|\nabla \mathbf{u}_m|} |\nabla \mathbf{u}|^2, \tag{6.10}$$

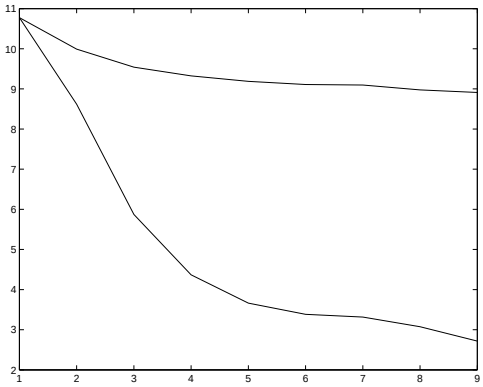


Figure 6.1: Error after the nth iteration of steepest descent (upper curve) and conjugate gradient (lower curve) methods.

solution of which becomes a new estimate $\mathbf{u}_{m+1}$. It can be shown [36] that $\mathbf{u}_m$ converges to the desired minimum for $m \to \infty$. For numerical reasons we take $\max(\varepsilon, |\nabla \mathbf{u}_m|)$ in place of $|\nabla \mathbf{u}_m|$ in (6.10). 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 gradients method (or some other method such as GMRES [61]) is crucial for the success of the minimization. Figure 6.1 shows a simulation when we know the correct depth map and minimization is run just over the image subspace. We can see that in case of steepest descent it may look like converging but it is still very far from minimum which is zero in this case.

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 interval bisection method. In this subspace the convergence turned out 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 $Q(\mathbf{u})$ seems to have no 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 may cause convergence problems at places, where the depth rapidly changes. In most cases we recommend TV regularization for the

image and Tikhonov regularization for the depth map.

6.4 Scheme of iterations

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

Experiments showed that the result of minimization and the speed of convergence depends on the number and order of iterations. In this section we will explain notation used to describe it. The Algorithm I consists of three levels of iterations. To describe the whole sequence of iterations, we need to 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 N_g in the description of the algorithm (p. 41).

The minimization over the image $\mathbf{u}$ depends on the type of regularization. In case of Tikhonov regularization, we apply the conjugate gradients methods 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 (6.10) solved again by conjugate gradients method. Then, N_{TV} refers to the length of this sequence and N_u relates to the number of iterations of conjugate gradients 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. Note that this time with 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 was $N_u = 8$ and $N_w = 10$. Interestingly, in 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 another rule states that the alternating minimization is faster if used with more image regularization. Therefore, we can use 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 higher number of iterations. We stress that this time it is necessary to repeat several times the minimization (6.10) to get what we want.

Thus, a typical description of iterations can look like $50 \times (8+10)+5 \times 25$. Note that we leave out the N_{TV} since it is equal to one.

6.5 Choice of regularization parameters

We already mentioned that regularization is an effective way to get reasonable solutions to problems that involve inversion of ill-conditioned operators [56]. For the first time, the choice of regularization constants in image restoration problems was addressed by Hunt [62]. An overview of methods can be found in [57].

Unfortunately, it seems difficult to apply known approaches directly to our problem. Nevertheless, a promising direction of future research could be the application of *generalized cross-validation* (GCV) for estimation of the regularization parameters similarly to [61, 63]. GCV is based on the idea of the "leave-one-out" principle which basically takes 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 non-linearity of the problem. The papers working along similar lines [7, 8, 9] do not address this problem 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 given amount of blur, it will probably work for other images in the same application as well.

Another aspect of the issue with the estimation of regularization parameters is that we do not have just one correct definition, 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).

6.6 Extension to color images

The algorithm can be extended to color images in a straightforward manner.

The error term of the functional (6.1) is summed over all three color channels. Similarly, image regularization term can be implemented as the sum of regularization terms for individual channels. Alternatively, better results can be achieved when TV is applied on multivalue images [59] using regularization term

$$\int_{\mathcal{D}} \sqrt{|\nabla \mathbf{u}_r|^2 + |\nabla \mathbf{u}_g|^2 + |\nabla \mathbf{u}_b|^2}, \tag{6.11}$$

which suppresses noise more effectively. Another advantage of this approach is that it prevents color artifacts at the edges. We used this approach in the experiments with color images presented in this thesis.

Depth map is common for all the channels, which brings additional resistance to noise.

6.7 Extension to general camera motion

If the camera motion and camera parameters (focal length, resolution of the sensor) are known, the proposed algorithm can be, at least in theory, extended to the case of general camera motion. As this topic deserves further investigation, we just summarize very briefly the main differences with respect to the special case we have detailed above.

The functional remains the same, except of 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, depth map is no longer common for all the images and consequently, for $p > 1$, the depth map must be transformed to the coordinate system of the image p before computing h_p using (5.1) and (5.2). The same is true in the auxiliary algorithm for the estimation of initial depth map, where the convolution becomes space-variant convolution.

The formulas in Proposition 1 hold in general case as well (see the proof) 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 necessary density of this grid depends on application. However, the numerical integration of the velocity field can be quite time-consuming even for a moderate size set of coordinates.

In turn, a nice property of this approach is that once all the masks 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.

6.8 Summary

In Chapter 6, we have presented the main contribution of this thesis, a multi-channel variational method for restoration of images blurred by space-variant out-of-focus blur, camera motion blur or both simultaneously.

The algorithm works independently of a particular shape of PSF, which allows to use more precise models of blur than previously published methods. For out-of-focus blur, it includes optical aberrations, for motion blur, translational motion in one plane perpendicular to the optical axis. In the latter case, the algorithm needs to know neither camera motion nor camera parameters. Besides, if the camera motion is known, the algorithm seems to be extensible to general camera motion. This case needs further investigation.

The algorithm is based on the minimization of a complex functional with many local minima. To solve the problem how to localize the right minimum, Algorithm I uses an initial estimate of depth map provided by one of simpler methods described in the following two chapters.

The main weakness is high time consumption. However, this issue can be alleviated by the fact that the algorithm uses only simple linear operations, which could facilitate potential hardware implementation.

Chapter 7

Depth from symmetrical blur (Algorithm II)

Basically there are two groups of multichannel algorithms that recover three-dimensional scene structure (depth map) based on the measurement of the amount of blur. The first and historically older group of algorithms is based on the assumption that the amount of blur does not change in a sufficiently large neighborhood of each image pixel. They often suffer from noise and poor accuracy.

Algorithms of the second group, variational methods, take an image formation model and look for the solution that minimizes its error with respect to the input images (see description of Algorithm I). Unfortunately, the minimization of the resulting cost functional is a nonlinear problem of very high dimension, its minimization takes a lot of time and tends to trap in one of many local minima. Nevertheless, if the minimum is localized correctly, the result is relatively precise. To avoid the problem with local minima we can naturally use an algorithm from the first group as an initial estimate. In this way we use the method presented in this chapter. For an overview of related literature see Sections 2.1 (Depth from defocus) and 2.2 (Depth from motion blur) in the literature survey.

Subbarao and Surya [1] proposed a filter based method, already mentioned in the overview of relevant literature, which gives an estimate of depth from two out-of-focus images assuming Gaussian PSF. It can be implemented by just two convolutions with small masks as can be seen from expression (2.1). In this chapter we show that their method can be modified to work with arbitrary sufficiently symmetrical PSF.

Resulting expressions (7.1)-(7.5) are formally very similar to that in [1]. We stress that it is not the best existing filter-based method but it is the simplest one and it was intended mainly as an initial estimate for variational

Algorithm I.

Notation will be the same as in Algorithm I. We work with two blurred images $\mathbf{z}_1$ and $\mathbf{z}_2$, supposed to be registered [64] at the input. This method assumes that the amount of blur is approximately constant within a sufficiently large window in the neighborhood of each image pixel which allows to model the blur locally by convolution with a mask.

7.1 Filters for estimation of relative blur

The whole algorithm is based on the following statements describing relative blur between images $\mathbf{z}_1$ and $\mathbf{z}_2$ expressed as the difference between the second moments of masks $\mathbf{h}_i$. Of course, if we want to use the relative blur to recover the depth, it must be an invertible function of the depth, at least for the interval of considered depths. For many real cases, it is satisfied.

Propositions assume apparently very limiting condition that the sharp image $\mathbf{u}$ is third-order polynomial within a local window. Later we will show that this condition can be approximately met using a simple trick.

Proposition 2. *Let $\mathbf{u}(x, y)$ be a third-order polynomial*[1] *of two variables and $\mathbf{z}_i = \mathbf{u} * \mathbf{h}_i$, $i = 1, 2$, where $\mathbf{h}_i$ are energy preserving ($\int \mathbf{h} = 1$) PSFs symmetric about axes x, y and both axes of quadrants. Then*

$$\sigma_2^2 - \sigma_1^2 = 2\frac{\mathbf{z}_2 - \mathbf{z}_1}{\nabla^2\left(\frac{\mathbf{z}_1+\mathbf{z}_2}{2}\right)}, \tag{7.1}$$

where σ_1^2, σ_2^2 are the second moments[2] *of $\mathbf{h}_1$ and $\mathbf{h}_2$ and ∇^2 is the symbol for Laplacian.*

Proof can be found in Appendix B. Note that the condition of symmetry in Proposition 2 holds for all circularly symmetric masks. We mentioned in Chapter 4 that it is a property of any axially symmetric lens with arbitrarily strong optical aberrations in the proximity of image center. Of course, relation between σ_1 and σ_2 must be carefully calibrated in this case.

In case of pillbox PSF, we can use relation between radius r of blur circle and its second moment $r = 2\sigma$ to get

[1] Two-dimensional third-order polynomial is a polynomial $P(x, y) = \sum_{m=0}^{3}\sum_{n=0}^{3-m} a_{m,n}x^m y^n$.

[2] If we take mask as distribution of a random quantity, the second moment or variance is usually denoted as σ^2. For two-dimensional functions there are actually three second-order moments but here $\sigma^2 = \mathbf{h}_{2,0} = \mathbf{h}_{0,2}$ and mixed second-order moment $\mathbf{h}_{1,1}$ is zero, both thanks to the symmetry.

Corollary 1. *Let* $\mathbf{u}(x, y)$ *be a third-order polynomial of two variables and* $\mathbf{z}_i = \mathbf{u} * \mathbf{h}_i$*,* $i = 1, 2$*, where* $\mathbf{h}_i$ *are energy-preserving pillbox PSFs of radii* r_i*. Then*

$$r_2^2 - r_1^2 = 8\frac{\mathbf{z}_2 - \mathbf{z}_1}{\nabla^2\left(\frac{\mathbf{z}_1+\mathbf{z}_2}{2}\right)}. \tag{7.2}$$

If we know camera parameters, we can use linear relation (4.8) to get r_1 and r_2 and equation (4.4) to estimate real depth. In the special case of $\beta = 0$ we get

$$r_1 = \frac{1}{\sqrt{1-\alpha^2}}\sqrt{r_2^2 - r_1^2}, \tag{7.3}$$

which is useful even if we do not know α to get at least a scaled version of the depth map.

Similar proposition holds for $\mathbf{h}_i$ being one-dimensional even PSF, which can happen in case of motion blur.

Proposition 3. *Let* $\mathbf{u}(x, y)$ *be a third-order polynomial*[1] *of two variables and* $\mathbf{z}_i = \mathbf{u} * \mathbf{h}_i$*,* $i = 1, 2$*, where* $\mathbf{h}_i$ *are energy preserving (*$\int \mathbf{h} = 1$*) one-dimensional even PSFs oriented in the direction of the* x*-axis. Then*

$$\sigma_2^2 - \sigma_1^2 = 2\frac{\mathbf{z}_2 - \mathbf{z}_1}{\frac{\partial^2}{\partial x^2}\left(\frac{\mathbf{z}_1+\mathbf{z}_2}{2}\right)}, \tag{7.4}$$

where σ_1^2, σ_2^2 *are the second moments of* $\mathbf{h}_1$ *and* $\mathbf{h}_2$*.*

In case of one-dimensional rectangular impulse corresponding to steady motion in the direction of the x-axis we get

Corollary 2. *Let* $\mathbf{u}(x, y)$ *be a third-order polynomial of two variables and* $\mathbf{z}_i = \mathbf{u} * \mathbf{h}_i$*,* $i = 1, 2$*, where* $\mathbf{h}_i$ *are energy-preserving rectangular impulses of length* d_i *oriented in the direction of the* x*-axis. Then*

$$d_2^2 - d_1^2 = 24\frac{\mathbf{z}_2 - \mathbf{z}_1}{\frac{\partial^2}{\partial x^2}\left(\frac{\mathbf{z}_1+\mathbf{z}_2}{2}\right)}. \tag{7.5}$$

Proofs can be found in Appendix B.

If the above mentioned motion blur originated in steady motion of the camera in a direction perpendicular to the optical axis, according to (5.5), the extent d of the motion blur depends linearly on the inverse distance of the scene from camera. If we take two images from cameras of the same

velocity with shutter speeds T_1 and T_2, d_2/d_1 is equal to the ratio $\alpha = T_2/T_1$ and

$$d_1 = \frac{1}{\sqrt{\alpha^2 - 1}} \sqrt{d_2^2 - d_1^2}. \tag{7.6}$$

To get the actual depth map, we can use equation (5.5). Again, even if we do not know α, we can omit the constant term and (7.6) gives us a useful representation of the scene as the actual distance is just its multiple given by camera parameters.

However, there are not many practical situations, when camera moves in this simple manner. First, camera rarely moves at constant speed. One exception is a camera pointing out of the window of a moving vehicle. In this situation speed remains approximately constant as the shutter time is relatively short. Another issue is that it is quite difficult to get "coordinated" measurements so as the position of the camera in the middle of the interval of open shutter agrees. It requires a special hardware, which further limits applicability of Algorithm II on motion blur. One possibility is to attach two cameras to the same lens using semi-transparent mirror [26] and synchronize shutters appropriately. At least in theory, a similar result could be achieved using two stereo cameras rigidly attached above each other with respect to the direction of side-motion if the disparity due to their relative position can be neglected.

7.2 Polynomial fitting filters

Subbarao and Surya [1] also noticed that the assumption on $\mathbf{u}$ to be a third-order polynomial can be approximately satisfied by fitting third-order polynomials to the blurred images. It is not difficult to see that polynomial fitting in the least square sense can be done by convolution with a filter, say $\mathbf{p}$. If $\mathbf{z}_i = \mathbf{u} * \mathbf{h}_i$ then the commutativity of convolution implies $\mathbf{z}_i * \mathbf{p} = \mathbf{u} * \mathbf{p} * \mathbf{h}_i$ for an arbitrary mask $\mathbf{h}_i$. Now, if $\mathbf{p}$ fits polynomial to $\mathbf{u}$, $\mathbf{u} * \mathbf{p}$ is smooth enough to be close to a third-order polynomial and we can use Propositions 2 and 3 with $\mathbf{z}_i * \mathbf{p}$ to get the relative blur $\sigma_2^2 - \sigma_1^2$. Notice that there is a trade-off between precision of depth estimates (which needs large support of $\mathbf{p}$) and precision of localization since large support of $\mathbf{p}$ needs larger area of space-invariant blur.

Polynomial smoothing filters corresponding to different window sizes can be described by surprisingly simple explicit expressions given by Meer and Weiss [60]. Thus the one-dimensional third-degree polynomial can be fitted

by convolution with quadratic function

$$L_0(n) = -\frac{3\left(5n^2 - (3N^2 + 3N - 1)\right)}{(2N-1)(2N+1)(2N+3)}, \tag{7.7}$$

where the support of the filter is $n = -N, -(N-1), \ldots, 0, \ldots, N-1, N$.

Similarly the second derivative of the fitted third-degree polynomial can be directly expressed as convolution of the image with

$$L_2(n) = -\frac{30\left(3n^2 - N(N+1)\right)}{(N(N+1)(2N-1)(2N+1)(2N+3)}. \tag{7.8}$$

Corresponding two-dimensional filters fitting two-dimensional third-degree[3] polynomials are separable, i. e. they can be expressed by convolution of corresponding one-dimensional filters as $L_0(n)^T * L_0(n)$ for the smoothing filter and $L_0(n)^T * L_2(n)$ for the second partial derivative.

If we need the result to be invariant with respect to the rotation of the image, we can use circular instead of rectangular window. The only drawback is that the filter is not separable and consequently takes a bit more time to compute.

7.3 Summary

In this chapter, we described an extension of filter-based DFD method [1] to arbitrary blur with centrally-symmetrical PSF. The method is extremely fast, requires only two convolutions.

The main application area of this algorithm is defocus. Both Gaussian and pillbox PSFs satisfy assumptions of this algorithm and even if we consider optical aberrations, the PSF is approximately symmetrical at least in the proximity of the image center. In this case, the method requires careful calibration.

As for motion blur, there are not many practical situations that fulfill requirements of this model. They are met only in the case of simple steady motion in a direction perpendicular to the optical axis or in the case of harmonic vibration that is symmetric about its center.

In the following chapter, we present more precise algorithm working with an arbitrary type of blur, which turned out to be more suitable for the needs of Algorithm I.

[3]Here we use term third-degree polynomials for two-dimensional polynomials with terms $a_{m,n}x^m y^n$, $m <= 3, n <= 3$ opposed to third-order polynomials, where $m + n \leq 3$.

Chapter 8

Depth from blur (Algorithm III)

In this chapter we present another algorithm for estimation of depth map from two or more blurred images of the same scene originally meant as auxiliary procedure to initialize depth map in Algorithm I. Compared to Algorithm II and most of the methods published in literature, it places no requirements on the form of the PSF, is less sensitive to the precise knowledge of the PSF and more stable in the presence of noise. On the other hand, it is more time-consuming. Compared to [6], which also works for almost arbitrary PSF and has similar time requirements, is much simpler to implement.

The algorithm works for the same class of problems as Algorithm I. In the basic version, described in Section 8.1, it includes the case of translational motion of the camera in one plane perpendicular to the optical axis and Gaussian or pillbox PSF for out-of-focus blur. The algorithm can be easily extended to the case of significant optical aberrations. The extension to general camera motion is possible in principal, as discussed in Algorithm I, but requires further investigation.

8.1 Description of the algorithm

Suppose that the blurred images $\mathbf{z}_1$ and $\mathbf{z}_2$ are registered and we know their camera parameters and noise levels. Similarly to the other presented algorithms, we must know the relation between PSF and depth for both images. This relation is assumed to be given in discrete steps as masks $h_i(k\ \triangle_w)$. For reasons detailed in Chapter 6 we store masks in equal steps of inverse distance, which corresponds to equal steps in the size of the PSF.

The algorithm assumes that the blur is approximately invariant in a neighborhood of each image pixel. For each pixel it computes minimum

$$\min_k \Bigg| \mathbf{m} * \Big[\big(\mathbf{z}_1 * h_2(k \bigtriangleup_w) - \mathbf{z}_2 * h_1(k \bigtriangleup_w) \big)^2 - \big(\sigma_2^2 \| h_1(k \bigtriangleup_w) \|^2 + \sigma_1^2 \| h_2(k \bigtriangleup_w) \|^2 \big) \Big] \Bigg| \tag{8.1}$$

over all entries k covering the interval of possible depths. It is usually sufficient, if the step $\bigtriangleup_w$ corresponds to about 1/10 of pixel in the size of PSF. For details see the description of the PSF implementation on p. 40. Mask $\mathbf{m}$ is a convenient averaging window (rectangular or circular). Parameters σ_1 and σ_2 are variances of additive noise present in $\mathbf{z}_1$ and $\mathbf{z}_2$ respectively. Thanks to commutativity of convolution, if there were no noise in $\mathbf{z}_1$ and $\mathbf{z}_2$, the left term of (8.1) would be zero for the correct level of blur. In reality, the right term becomes important. It equals the expected value of the first term for correct masks given noise levels in $\mathbf{z}_1$ and $\mathbf{z}_2$. Without this term, the algorithm prefers masks with small norms (that is large blurs) that remove noise almost completely.

8.2 Time complexity

In the actual implementation we compute convolution of the whole images $\mathbf{z}_1$ and $\mathbf{z}_2$ with all the masks (or a subset) stored in the arrays corresponding to h_1 and h_2 respectively and for each pixel we choose the entry with the minimal value. It means that the algorithm computes twice more convolutions than the number of considered blur levels. To suppress noise and avoid problems in the areas of weak texture, we average the error over a window of fixed size. The time of averaging can be neglected as it can be done in $O(1)$ time per pixel. For square window, simple separable algorithm needs four additions per pixel. Altogether, if we use the above mentioned step of 1/10 of pixel in the diameter of the support of PSF, the number of convolutions the algorithm takes is $2 \times 10\times$ the diameter of maximal blur in pixels.

8.3 Noise sensitivity

The quality of result naturally depends on the level of noise present in the input images. Compared to other filter-based methods, this algorithm proved to be relatively robust with respect to noise. Moreover, if the noise level is

too high, we simply use larger window for error averaging. Doubling the size of the window decreases the mean error in (8.1) approximately by the factor of four. The price we pay for this improvement is that we effectively half the resolution of the image in the neighborhood of the edges. In other words, we will get less noisy depth map of lower spatial resolution.

If we do not know the actual noise variance, we can set $\sigma_1 = \sigma_2 = 0$ and for moderate noise levels and a reasonable upper estimate of the mask support it will often give satisfactory results.

8.4 Possible extensions

If we have more than two images, we sum the value of (8.1) over all pairs of images. A similar strategy can be used with RGB images. The error is simply computed as the sum of errors in individual channels. If the level of noise is low, it usually brings no much improvement because of strong correlation between channels. In the opposite case, the improvement can be significant.

We can use the algorithm even if the PSF is a function of not only distance but also of the position in the field of view. It includes optical aberrations or zooming motion. The only difference is that we replace convolution by its space-variant counterpart. For details of the difficult case of general camera motion see the discussion in Section 6.7.

Chapter 9

Precision of depth estimates

How precise are depth estimates produced by the proposed algorithms? Our experiments and analysis of published methods indicate that it is not possible to estimate the local extent of blur with precision higher than some constant fraction of one pixel. Applying relation between the precision of distance measurements and precision of detected support of PSF, we obtain an upper limit for the precision of depth estimates we can expect from methods using the amount of blur to measure distance.

We begin by recalling the linear dependence of the size of the blur circle on the inverse of the distance from camera (4.6). By differentiating with respect to the distance l we get

$$\frac{\partial r}{\partial l} = -\frac{\rho\zeta}{l^2} \,. \tag{9.1}$$

One consequence of (9.1) is an intuitive fact that small depth of field is essential for the precision of DFD methods as the error is proportional to the reciprocal of the aperture radius ρ. Second, assuming a constant error in detected blur size, the absolute error of the distance measurements increases quadratically with the distance from camera and the relative (percentage) error increases linearly.

Obviously, the same is true for all blurs depending linearly on the inverse distance $1/l$. We have shown that this is a property of several other types of blur considered in this thesis. Moreover, exactly the same is well known to be true in stereo, where distance is proportional to the reciprocal of pixel disparity [4]. It should come as no surprise as disparity is nothing other than the length of motion smear in the case of motion along stereo baseline.

We believe that this is a principal limitation of all ranging methods based on image pixel measurements, including stereo, DFF, DFD and depth from

motion blur, which is in agreement with arguments of Schechner and Kiryati [5] that DFD and stereo are not principally different.

Chapter 10

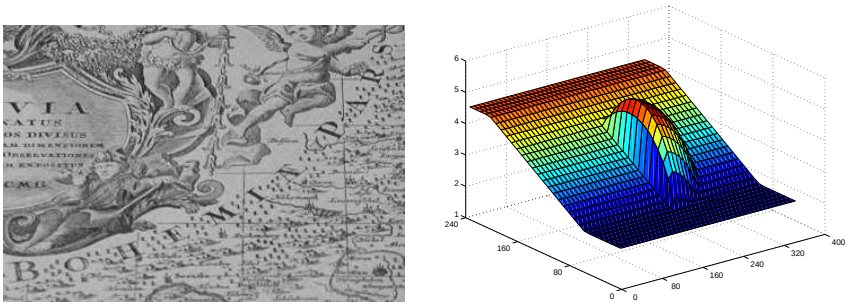
Experiments on synthetic data

To give the full picture of the properties of the proposed algorithms we present two groups of experiments. Experiments on synthetic data (simulated experiments) assume that the image formation model is correct and test numerical behavior of the presented algorithms in presence of different amounts of noise using the knowledge of ground truth. Experiments working with real data, on the other hand, are intended to validate the model we used and assess its applicability. We start with the experiments on synthetic data. Real experiments are presented in the next chapter.

First, let us look at the figure of historical map Fig. 10.1(a) used as the original image for the simulated experiments. It contains areas of very complex texture but we can also find places of almost constant image function. Since proposed algorithms behave locally in the sense that the solution depends mainly on points in close neighborhood of the given point (one step of minimization depends only on the neighborhood of size corresponding to blur mask support), it suggests a lot about the behavior of the algorithms on different types of scenes.

To produce the artificial depth map representation we used data from Fig. 10.1(b) for both out-of-focus and motion blur experiments. In case of motion blur the graph gives the half length of the motion smear. In case of out-of-focus, the data correspond to the radius of the PSF support. Again, the scene was designed to show behavior of the algorithms on various types of surfaces—there are areas of constant depth (lower and upper parts of the image), slanted plane, steep edge and curved smooth surface. The central part of the depth map was generated as the maximum value of the slanted plane and a quarter-sphere.

All the experiments were carried out at four different levels of noise—zero (SNR = ∞), low (40dB), moderate (20dB) and heavy (10dB). As a rule, results are arranged in two column tables with each line corresponding



(a) original image, 245×356 pixels (b) depth map

Figure 10.1: Original image, artificial depth map and prototype mask used for simulated experiments. Z-coordinate of the depth map indicates half of the PSF size. Note that the "rear" part of the depth map corresponds to the most blurred lower part of images Fig. 10.2 and Fig. 10.8.

to certain noise level (zero noise in the first line, low in the second, etc.).

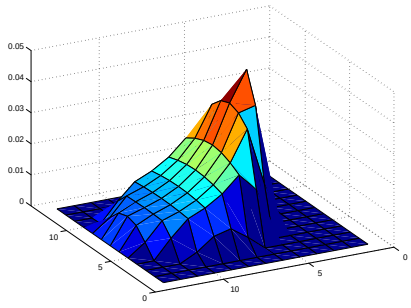
All experiments were run several times for different instances of noise and we give the average MSE. The restored images were almost visually undistinguishable and therefore images to present were chosen randomly. We used two channels, additional channels bring improvement approximately corresponding to decrease in noise variance we would obtain by averaging of measurements if we had more images taken with the same camera settings.

Since we know the corresponding ground truth Fig. 10.1, all the figures of restored images and depth maps contain the related value of mean square error (MSE). For images it is given in grey levels per pixel from 256 possible values. As follows from the discussion in Chapter 9, it has no much meaning to measure directly the error of depth since it depends on camera parameters and distance of the scene. Instead, we give the error of depth map as the error in blur radius or in the size of PSF support which is measured in pixels.

10.1 Out-of-focus blur

The first set of simulated experiments tests simultaneously Algorithms I and II for the case of out-of-focus blur.

To simulate how the PSF changes with the distance of corresponding object, we assumed that it keeps its shape and stretches analogously to models (4.12) and (4.19) to have the same support it would have if it was the pillbox of radius (4.4). It enables us to generate masks of arbitrary size from the prototype Fig. 10.2(a). The mask shape was chosen to imitate real PSF of a



(a) prototype mask, 13×13

(b) MSE = 17.21 levels

(c) MSE = 19.19 levels

Figure 10.2: To simulate out-of-focus blur, we blurred image Fig. 10.1(a) using blur map Fig. 10.1(b) and the PSF generated from prototype Fig. 10.2(a). The largest PSF support (in the lower part of the left image) is about 11×11 pixels. Amount of blur in the second (right) image is 1.2 times larger than in the first image (left), i. e. $\alpha_2 = 1.2$.

lens system with strong *coma* and *spherical aberration*[1] [53] in the area near the border of the field of view. We generated two channel (images) from Fig. 10.1(a) using depth map Fig. 10.1(b) assuming they had been captured with the same camera settings except of the aperture, which was considered 1.2 times larger in the second image, i. e. $\alpha_2 = 1.2$ and $\beta_2 = 0$. Finally, we added the above mentioned four levels of noise. Fig. 10.2 shows the result.

If we know the correct values of the depth map, it is not difficult to compute the image minimizing the cost functional using the first of two alternating phases of Algorithm I. Fig. 10.3 shows the result of such non-blind restoration using 100 iterations of Tikhonov regularization with $\lambda_u =$

[1]Optical aberrations are deviations from Gaussian optics. Both, coma and spherical aberration, appear when inner and outer parts of a lens have different focal lengths. Whereas spherical aberration does not change through the field of view, coma increases linearly with the distance from the image center and causes comet like effects at the periphery of the view field.

5×10^{-3}. We tested also total variation (TV) regularization but for this image the result turned out to look too blocky. Because of the guaranteed convergence of such minimization, it is the optimal result we can expect from any algorithm minimizing the cost functional over both unknowns. We will show that it is possible to achieve almost the same quality of restoration even if the actual depth map is not known. Notice that even in zero noise case, the mean square error of the result is about 5 levels. One could suspect it is caused by the influence of finite number of iterations, but it is negligible in this case and the actual reason is the regularization which makes the result somewhat smoother than it should be.

For comparison, in the right column we can see the result of the same restoration using Gaussian mask. It indicates the quality of the result we can expect if a method is limited to Gaussian mask and the mask significantly differs. Notice that the mean square error of the restored image is the same or even worse than of blurred images Fig. 10.2. It indicates that if we use wrong mask, we cannot hope for any reasonable result—at least in the sense of MSE. It is interesting that the result undoubtedly looks markedly sharper than the blurred images, which demonstrates the well known fact that the mean square error does not express exactly the human perception of image quality. Anyway, even from the human point of view, the results in the left column are much better and we will show that Algorithm I can achieve almost the same quality of restoration.

For the initial blur map estimate we use Algorithm II, covered in detail in Chapter 7. Note that the model of PSF we use does not satisfy requirements of Algorithm II, nevertheless the error is not as large. The first column of Fig. 10.4 shows the result for different amounts of noise. Obviously, we can use it directly for restoration. The second column shows the result of such a restoration, again using CG method with Tikhonov regularization and still the same $\lambda_u = 5 \times 10^{-3}$. The result looks relatively good, which is not very surprising since the MSE of the blur map is quite low, only 0.25 pixels. In reality, the error of this method can be much worse and even here, the error is still almost two times larger than that from Fig. 10.3 we want to approach.

Now, we will show that Algorithm I can provide results comparable with those in the left column of Fig. 10.3. We used TV regularization for the depth map and Tikhonov regularization for the image. Iteration scheme was $50 \times (8 + 10)$.

Fig. 10.6 gives resulting depth maps for Gaussian mask in the left column and the correct mask in the right column. The error with the correct mask is only about one-eight of a pixel, one half of the error achieved by direct restoration using the depth map produced by Algorithm II. Notice the blocky look of the top part of the quarter-sphere, which is a well known effect of

TV regularization. Corresponding restored images are presented in Fig. 10.7 and we can see that up to moderate noise level the result of Algorithm I is very satisfying. The MSE almost achieved the optimal values from Fig. 10.3 and with the exception of the depth discontinuity in the proximity of the image center, the image is visually undistinguishable from the original image Fig. 10.1(a). The issue at the discontinuity is very illustrative. Experiments showed that, at least in our implementation, using TV regularization for depth map often gave rise to convergence problems at places like that. In real experiments we will demonstrate, that it is often better to use Tikhonov regularization, which leads to somewhat oversmoothed depth map, but better image restoration. In this case, the problem is worsened by a shift of the edge position due to the unprecise localization of the edge typical for Algorithm II and all other algorithms based on the assumption of local space-invariance of the blur. Then, because of the problem with many local minima of the cost functional, the minimization algorithm is not able to "push" the edge back to the right position.

(a) SNR = ∞, MSE = 4.79 levels

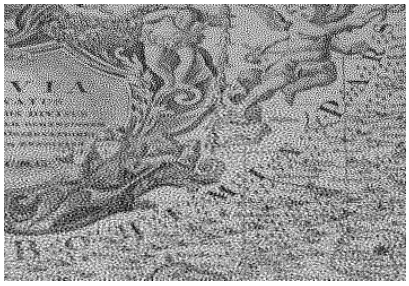
(b) SNR = ∞, MSE = 17.68 levels

(c) SNR = 40dB, MSE = 5.22 levels

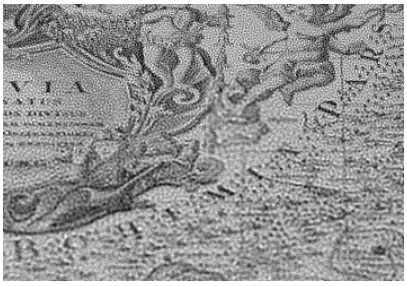
(d) SNR = 40dB, MSE = 17.75 levels

(e) SNR = 20dB, MSE = 16.39 levels

(f) SNR = 20dB, MSE = 18.48 levels

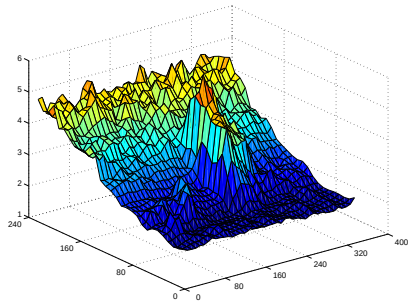


(g) SNR = 10dB, MSE = 52.22 levels



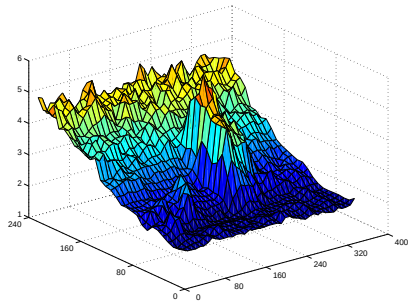
(h) SNR = 10dB, MSE = 29.01 levels

Figure 10.3: Result of restoration of images from Fig. 10.2 using known blur map 10.1(b) and prototype mask 10.2(a), 100 iterations of CG method, Tikhonov regularization with $\lambda_u = 5 \times 10^{-3}$. The best result we can expect from any algorithm minimizing the cost functional. In the right column the same reconstruction using Gaussian mask, the result we can expect from methods that assume fixed Gaussian PSF if it does not correspond to reality.



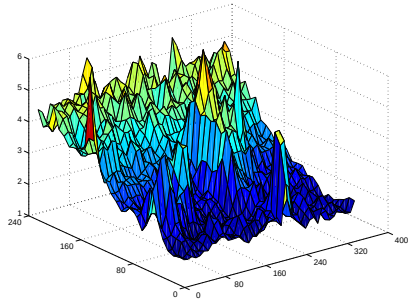
(a) SNR = ∞, MSE = 0.25 pixels

(b) SNR = ∞, MSE = 10.87 levels

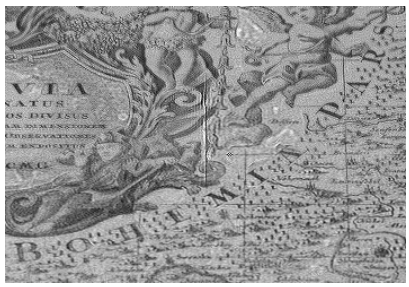


(c) SNR = 40dB, MSE = 0.26 pixels

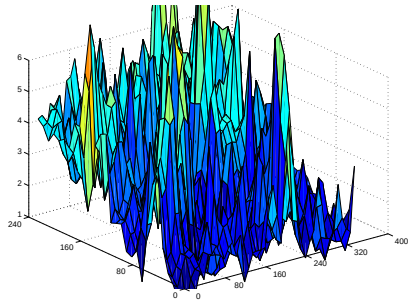
(d) SNR = 40dB, MSE = 11.39 levels



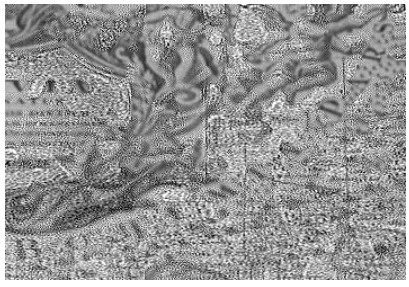
(e) SNR = 20dB, MSE = 0.51 pixels



(f) SNR = 20dB, MSE = 17.63 levels

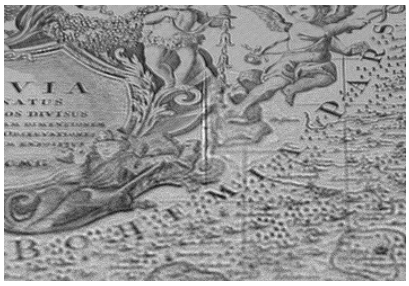


(g) SNR = 10dB, MSE = 1.42 pixels

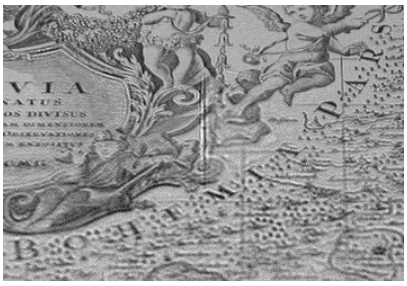


(h) SNR = 10dB, MSE = 45.70 levels

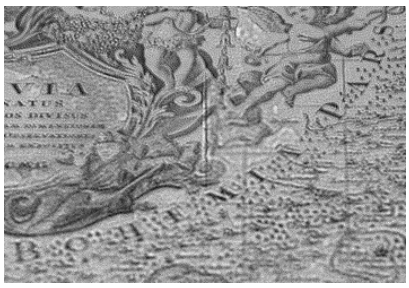
Figure 10.4: Depth maps recovered directly using filter based Algorithm II (smoothed by 11×11 median filter) and corresponding restorations.



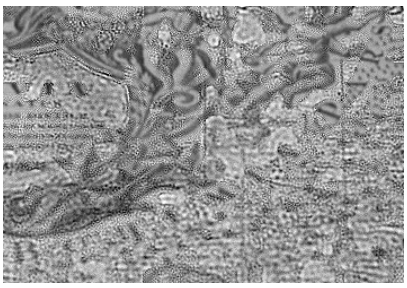
(a) SNR = ∞, MSE = 18.97 levels



(b) SNR = 40dB, MSE = 19.02 levels

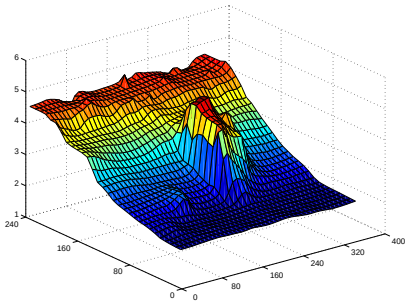


(c) SNR = 20dB, MSE = 20.62 levels

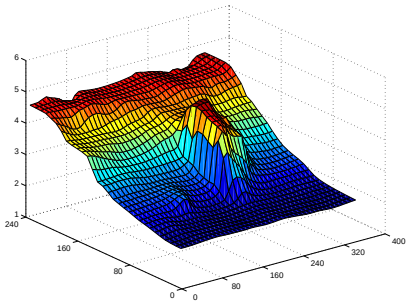


(d) SNR = 10dB, MSE = 32.31 levels

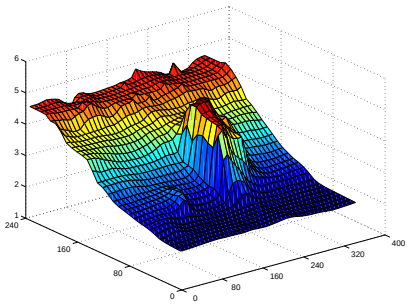
Figure 10.5: Restorations with Gaussian PSF using depth maps from the left column of Fig. 10.4.



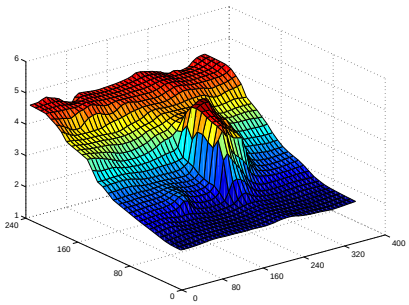
(a) SNR = ∞, MSE = 0.170 pixels



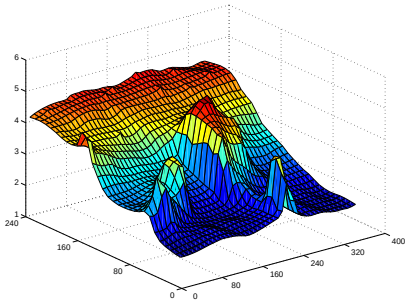
(b) SNR = ∞, MSE = 0.152 pixels



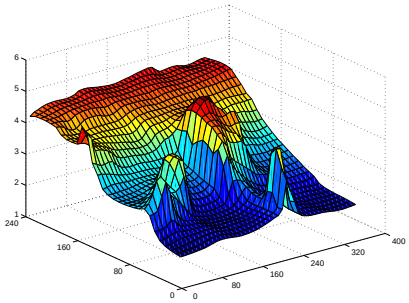
(c) SNR = 40dB, MSE = 0.172 pixels



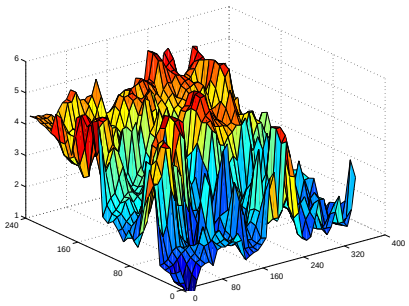
(d) SNR = 40dB, MSE = 0.151 pixels



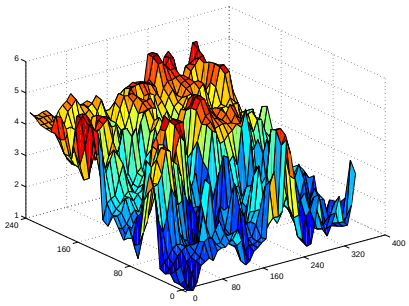
(e) SNR = 20dB, MSE = 0.309 pixels



(f) SNR = 20dB, MSE = 0.307 pixels



(g) SNR = 10dB, MSE = 0.895 pixels



(h) SNR = 10dB, MSE = 0.889 pixels

Figure 10.6: Depth map estimate we got from Algorithm I. In the first column using (wrong) Gaussian mask, in the second column using the correct mask. Iteration scheme $50 \times (8 + 10) + 100$. Interestingly, the depth map got by Gaussian mask is not much worse than using correct mask.

(a) SNR = ∞, MSE = 16.43 levels

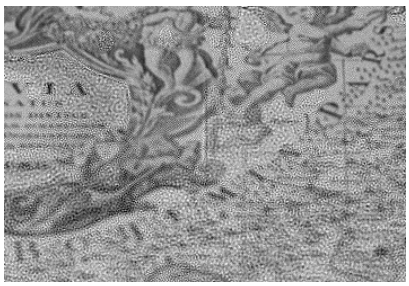
(b) SNR = ∞, MSE = 6.12 levels

(c) SNR = 40dB, MSE = 16.47 levels

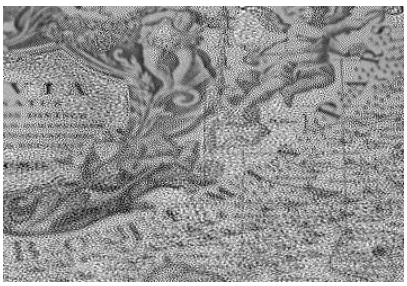
(d) SNR = 40dB, MSE = 6.42 levels

(e) SNR = 20dB, MSE = 18.72 levels

(f) SNR = 20dB, MSE = 15.42 levels



(g) SNR = 10dB, MSE = 31.57 levels



(h) SNR = 10dB, MSE = 42.89 levels

Figure 10.7: Restored images corresponding to Fig. 10.6, i. e. using Gausian PSF (left column) and correct PSF Fig. 10.2(a) (right column). In both cases iteration scheme $50 \times (8 + 10) + 100$.

(a) l_{max} = 8.25 pixels, MSE = 17.39 levels

(b) l_{max} = 9.90 pixels, MSE = 18.97 levels

Figure 10.8: To simulate motion blur, we blurred Fig. 10.1(a) using depth map Fig. 10.1(b). The extent of motion blur in second image (right) is 1.2 times larger than in the first (left) image, i. e. $\alpha_2 = 1.2$. Quantity l_{max} denotes maximal blur extent, we can see in the lower part of the images.

10.2 Motion blur

The second set of simulated experiments illustrates behavior of Algorithms I and II in case of motion blur. Its primary goal is to show limits of Algorithm I concerning the amount of noise and its sensitivity to the quality of initial depth map estimate.

This experiment has the same structure as the simulated experiment with out-of-focus blur. We used simple model of motion blur in the direction of x-axis, where the length of the motion smear is proportional to the inverse distance from camera. Recall that it is one of two simple types of motion blur the Algorithm II works with. In the next chapter, we present real experiments that work with more complex motion of the camera and require Algorithm III to get the initial estimate of the depth map.

Again, we used the original image Fig. 10.1(a) and depth map Fig. 10.1(b), blurred the original image in accordance with the model and added four different amounts of noise. The extent of motion blur in right image is 1.2 times larger than in the left image, that is $\alpha_2 = 1.2$.

The left column of Fig. 10.9 shows the depth map estimate computed by Algorithm II. We used it as initial estimate for Algorithm I and the result after 50 iterations can be seen in the right column of the same figure. The MSE clearly decreased by about one-third. Again, Fig. 10.9(f) is a nice illustration of the problem with local minima. Weak texture in the upper-left part of the images leads to wrong initial depth estimate and this propagates through the whole minimization resulting in the peaks in the lower-left corner of the depth map. Note that they developed primarily as a result of the noise

sensitivity of Algorithm II, not of the Algorithm I.

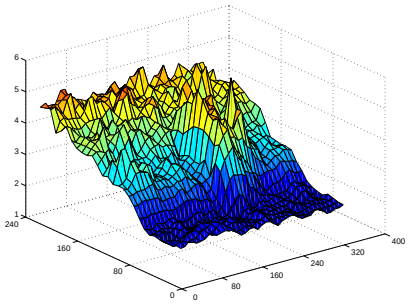
Fig. 10.10 allows to compare the result of corresponding restorations. We can see that in the zero noise case the result of minimization is almost visually undistinguishable from the ideal image Fig. 10.1(a), again with the exception of steep depth change in the central part of the image. Also the direct restoration using depth map computed by filter-based Algorithm II gives satisfactory result but the improvement of Algorithm I is clearly visible. Again, notice the depth edge in the image center and convergence problems in its neighborhood from reasons mentioned in the previous experiment. Similarly to the experiment with out-of-focus blur, real experiments will demonstrate that it is often better to use Tikhonov regularization.

10.3 Summary

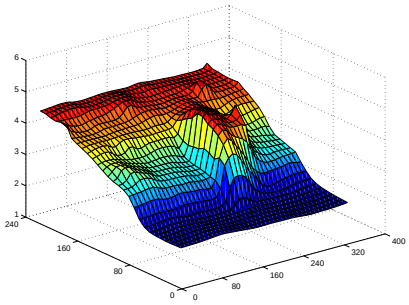
In this chapter, we have presented simulated experiments that demonstrated behavior of the proposed algorithm in the presence of four different levels of noise.

The scene for the experiments was chosen to represent various types of textures and the depth map was generated so as to cover several types of surfaces.

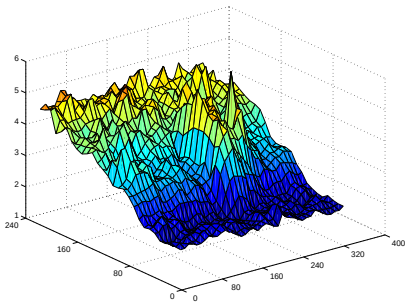
We demonstrated that Algorithm I works well up to about 20dB but is dependent to a large extent on good initial estimate of the depth map. The artifacts on the depth discontinuity (Fig. 10.7 and Fig. 10.10) were caused by the unprecise localization of the edge by Algorithm II typical for most of the algorithms based on the assumption of local blur space-invariance. We have seen as well that Algorithm II gives quite noisy results even for ideal input lacking any noise.



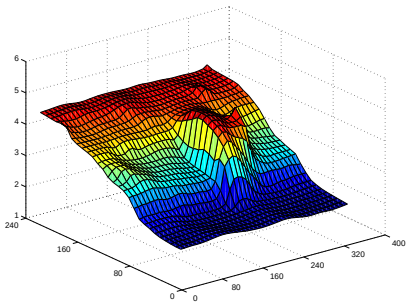
(a) SNR $= \infty$, MSE $= 0.31$ pixels



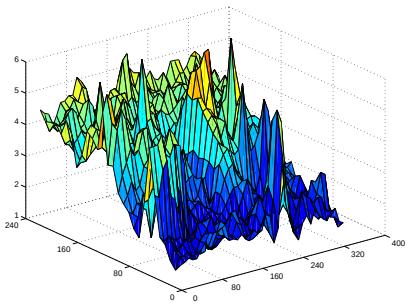
(b) SNR $= \infty$, MSE $= 0.21$ pixels



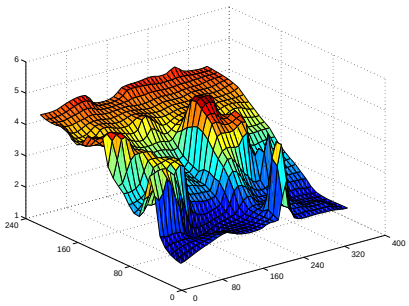
(c) SNR $= 40$dB, MSE $= 0.32$ pixels



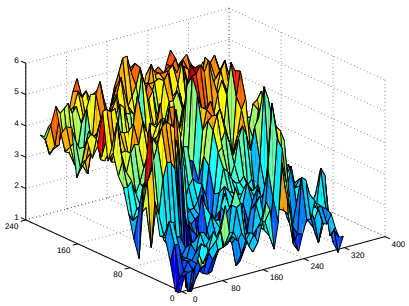
(d) SNR $= 40$dB, MSE $= 0.20$ pixels



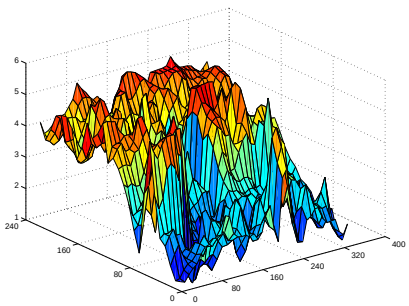
(e) SNR $= 20$dB, MSE $= 0.72$ pixels



(f) SNR $= 20$dB, MSE $= 0.44$ pixels



(g) SNR $= 10$dB, MSE $= 0.97$ pixels

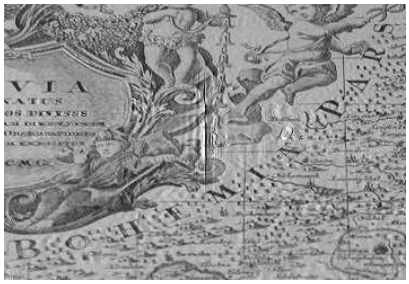


(h) SNR $= 10$dB, MSE $= 0.82$ pixels

Figure 10.9: Comparison of depth map estimation using Algorithm II (left column) and the result of Algorithm I (right column). We used Tikhonov regularization with $\lambda_u = 5 \times 10^{-3}$ and as the initial estimate we took the left column. Iteration scheme $50 \times (8 + 10)$.

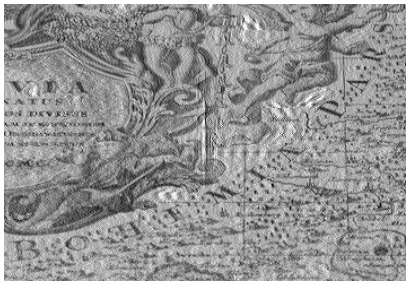
(a) SNR = ∞, MSE = 10.48 levels

(b) SNR = ∞, MSE = 8.09 levels

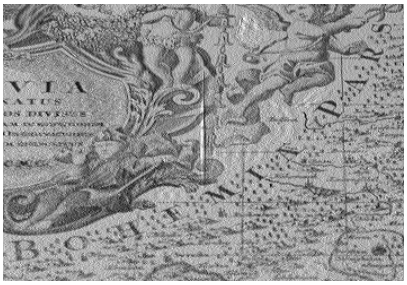


(c) SNR = 40dB, MSE = 12.59 levels

(d) SNR = 40dB, MSE = 10.89 levels



(e) SNR = 20dB, MSE = 24.99 levels



(f) SNR = 20dB, MSE = 21.91 levels

(g) SNR = 10dB, MSE = 59.93 levels

(h) SNR = 10dB, MSE = 51.44 levels

Figure 10.10: Comparison of restored images corresponding to Fig. 10.9. Results of filter-based Algorithm II (left column) and subsequent minimization using Algorithm I (right column). Iteration scheme $50 \times (8 + 10) + 100$.

Chapter 11

Experiments on real data

To document behavior of the proposed algorithms on real images we present three experiments, one for space-variant out-of-focus blur and two for the space-variant blur caused by camera motion. Algorithms II and III are not presented separately but they are discussed as part of Algorithm I.

In all cases we used digital SLR camera Canon 350D with set lens Canon EF-S 18–55mm II. For experiments with intensity (monochromatic) images we use red channel for the first and second experiments and green channel for the third experiment.

11.1 Out-of-focus blur

We focused the camera in front of the scene and took two images Fig. 11.7(a) and 11.7(b) from tripod using the same camera settings with the exception of the aperture. We chose f-numbers $F/5.0$ and $F/6.3$, which is the worst case in the sense that close apertures result in very similar blurred images and consequently bring least information about depth. To compare with reality, we took another image Fig. 11.7(c) with aperture $F/16$ to achieve large depth of focus. The basic version of the proposed algorithms works with intensity (monochromatic) images. In this experiment we consider red channel Fig. 11.1.

To show the difficulties arising from space-variance of the blur in the input images we took three small sections of approximately constant blur and computed corresponding PSFs using space-invariant blind restoration algorithm [2] (with parameters $\lambda = 1000$, $\varepsilon = 0.1$, $\gamma = 10$, support of both PSFs was set to 17×17 pixels). Fig. 11.2 shows results of restoration of the whole image using the computed PSFs (using least squares method with TV regularization which is a special case of the first part of Algorithm I). It can

be readily seen that in all the cases the images contain many artifacts in the areas where the degree of defocus differs significantly from the right value. Thus Fig. 11.2(a), deconvolved by PSFs valid in the lower part of the images, is completely out-of-focus in the parts further from camera. Fig. 11.2(b), on the other hand, results from PSFs valid on the wall in the upper right corner of the images and we can see strong ringing effects in the lower part of the image. Fig. 11.2(c) corresponds to the PSF valid at the front part of the flowerpot and is somewhat out-of-focus at the back and there are also artifacts around edges in the front (lower) part of the image. To overcome the principal limitations of space-invariant methods we must consider space-varying PSF which is the case of the algorithms proposed in this work.

An assumption of Algorithm I is that we know the relation between PSF and distance from camera (or a convenient representation of the distance). In this experiment we assume pillbox model of PSF which fits the real PSF quite well as can be seen from the results that follow. Restoration would not be much better even if we knew the right PSF precisely. Moreover, paradoxically, the pillbox is a good PSF shape for testing of algorithms because of the difficulties arising from its non-continuous derivatives with respect to depth.

Now, we will show the outcomes of Algorithm I, which is the main result presented in this thesis.

First, the algorithm needs a reasonable initial estimate of depth map. For this purpose, we used Algorithm III and got depth map Fig. 11.3(a) with brighter areas corresponding to further objects. Unfortunately, this depth map cannot be used directly for restoration. Indeed, even if we smooth the depth map to a large extent (here we used 7×7 window for error averaging and the result of the algorithm was smoothed by additional median filtering by 23×23 window), it still produces many artifacts, especially in the areas of weak texture. We illustrate this fact in Fig. 11.3(b)-11.3(d), where we can see images restored using the depth map from Fig. 11.3(a) for three different levels of image regularization. Notice the areas on the floor where low contrast, implying very high SNR, results in poor depth estimates which again results in artifacts in the restored image.

Fig. 11.4 shows depth maps produced by $20 \times (8 + 10)$ iterations of Algorithm I for combinations of three different depth regularization constants λ_w and two different image regularization constants λ_u. Note that all of them started from the initial depth map estimate Fig. 11.3(a). We can observe that the depth maps does not depend much on the degree of image regularization. The depth map regularization constant λ_w, on the other hand, determines smoothness of the depth map. Basically, we can choose between more robust depth map with lower spatial resolution and a depth map with higher spatial resolution and more errors in the areas of weak texture or low

contrast.

As we mentioned in the description of Algorithm I, the algorithm tends to converge faster for higher degree of image regularization (higher λ_u). Therefore, as a rule, we first minimize with some higher degree of image regularization (here $\lambda_u = 10^{-3}$) and finally we use the depth map we got for final restoration with less regularization and higher number of iterations (here we used 5×20 iterations of constrained least squares restoration with TV regularization).

Thus, we have got images Fig. 11.5 and 11.6 using three different depth maps Fig. 11.4(b), Fig. 11.4(d) and Fig. 11.4(f) (results for $\lambda_u = 10^{-3}$ were almost identical so we omit them) and three different values of image regularization constants. Results are divided in two figures, $\lambda_u^f = 10^{-3}$ and $\lambda_u^f = 10^{-4}$ in Fig. 11.5 and $\lambda_u^f = 3 \times 10^{-4}$ in Fig. 11.6. We can see that it is always possible to choose between sharper and noisier (smaller λ_u^f) and softer but less noisy image (higher λ_u^f). Interestingly, the level of depth map regularization has only a minor influence on the restored image.

In the description of Algorithm I we mentioned that the algorithm can be extended to work with color images as well. Here, we show a simplified approach that takes depth maps from Algorithm I and uses them for least squares restoration [33] modified for color regularization using the term (6.11).

Fig. 11.7 shows color version of out-of-focus images from Fig. 11.1. Fig. 11.8 gives result of restoration using depth maps Fig. 11.4(b), Fig. 11.4(d) and Fig. 11.4(f) and two different values of image regularization constant $\lambda_u^f = 10^{-4}$ and $\lambda_u^f = 10^{-5}$. Notice that we can use less regularization and consequently get sharper images since the regularization term (6.11) suppresses noise using information from all three RGB channels.

(a) out-of-focus image, 730 × 650 pixels, $F/5.0$

(b) another out-of-focus image of the same scene, $F/6.3$

(c) ground truth image taken with F/16

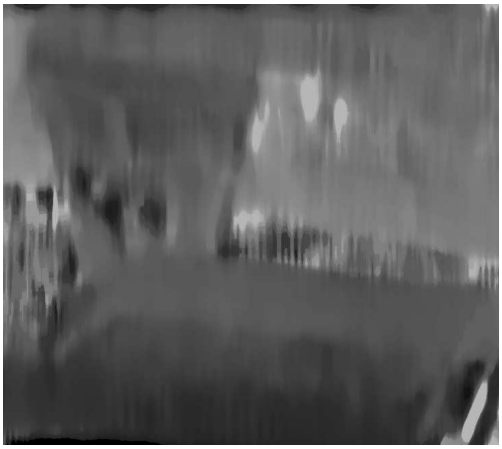
Figure 11.1: Red channel of RGB images in Fig. 11.7. The scene with flowerpot was taken twice from tripod. All the camera settings except of the aperture were kept unchanged. For comparison, the third image was taken with large f-number to achieve large depth of focus. It will serve as a "ground truth".

(a) deconvolution using PSFs valid in the lower part of the image

(b) deconvolution using PSFs valid on the wall in the upper-right corner of the image

(c) deconvolution using PSFs valid at the front of the flowerpot

Figure 11.2: Illustration of the fact that we cannot use space-invariant restoration methods. We used deconvolution with TV regularization and image regularization constant $\lambda_u = 10^{-4}$. In all cases, using only one PSF for the whole image results in clearly visible artifacts.



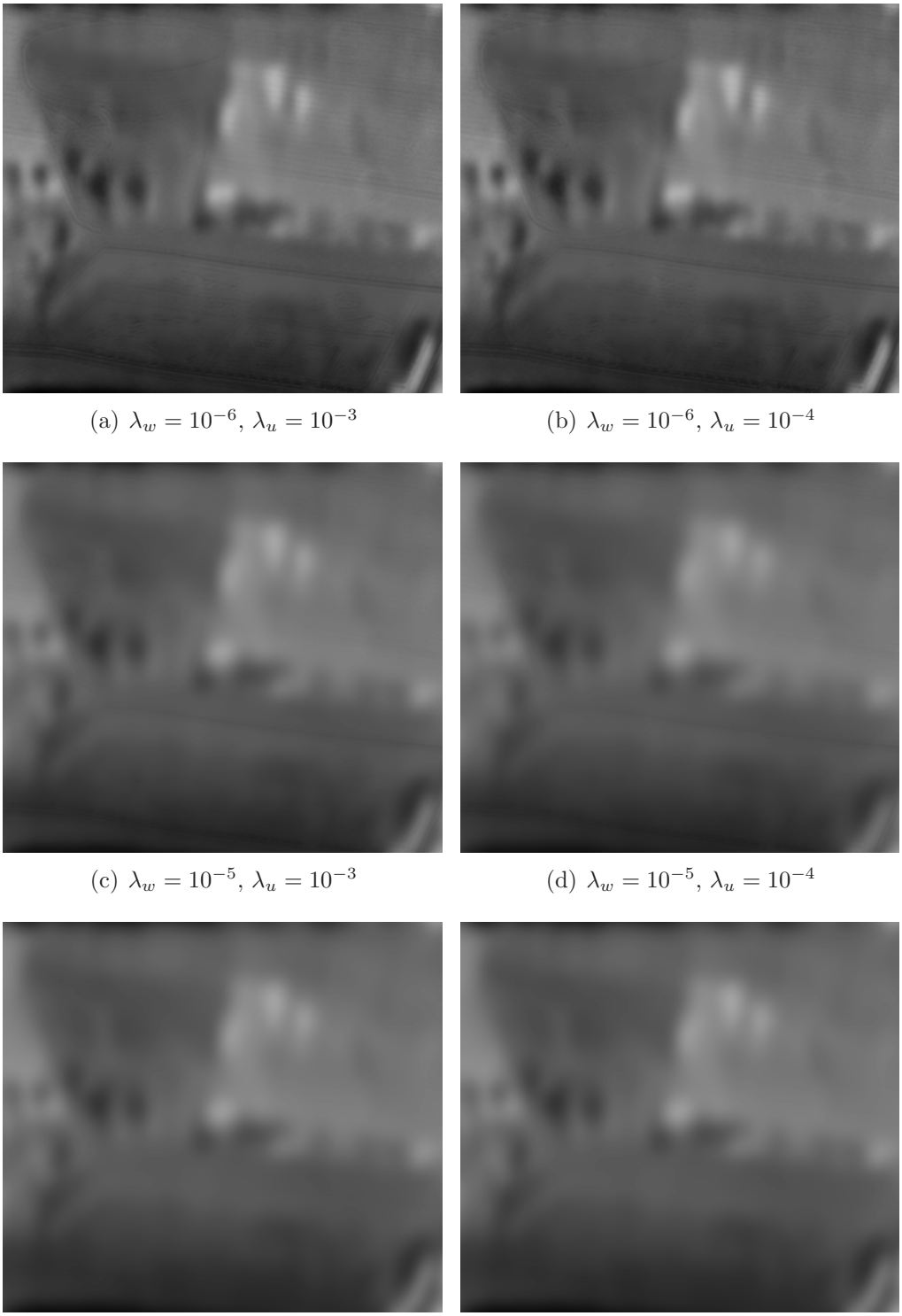
(a) depth map obtained by Algorithm III (7×7 window for error averaging) after smoothing by 23×23 median filter

(b) $\lambda_u = 10^{-3}$

(c) $\lambda_u = 3 \times 10^{-4}$

(d) $\lambda_u = 10^{-4}$

Figure 11.3: Illustration of the fact that we cannot use simple depth recovery methods directly for restoration. Results of TV restoration using depth map (a) for three levels of image regularization. We can see many visible artifacts, especially in the areas of weak texture.



(a) $\lambda_w = 10^{-6}$, $\lambda_u = 10^{-3}$

(b) $\lambda_w = 10^{-6}$, $\lambda_u = 10^{-4}$

(c) $\lambda_w = 10^{-5}$, $\lambda_u = 10^{-3}$

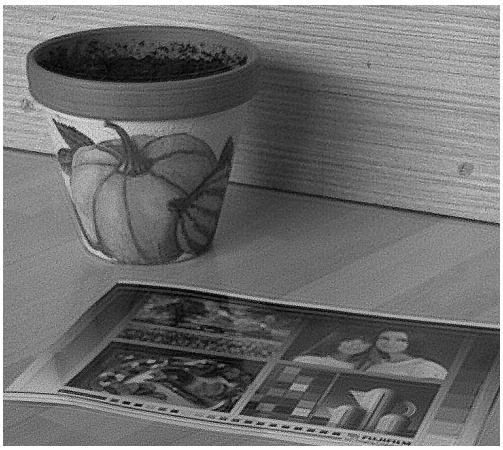
(d) $\lambda_w = 10^{-5}$, $\lambda_u = 10^{-4}$

(e) $\lambda_w = 10^{-4}$, $\lambda_u = 10^{-3}$

(f) $\lambda_w = 10^{-4}$, $\lambda_u = 10^{-4}$

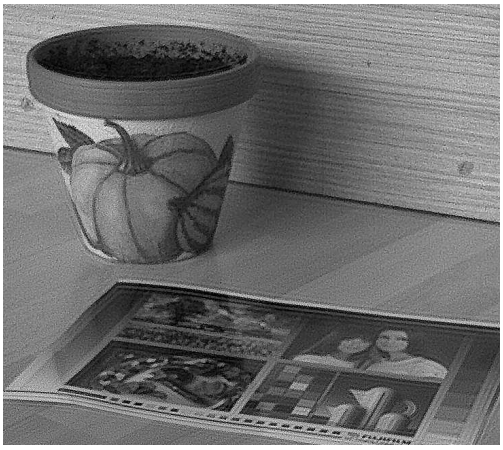
Figure 11.4: Depth maps produced by Algorithm I for three different levels of depth map regularization and two levels of image regularization. In all cases minimization started from depth map Fig. 11.3(a). Iteration scheme $20 \times (8 + 10)$.

(a) restoration using depth map 11.4(b), $\lambda_u^f = 10^{-3}$



(b) restoration using depth map 11.4(b), $\lambda_u^f = 10^{-4}$

(c) restoration using depth map 11.4(d), $\lambda_u^f = 10^{-3}$

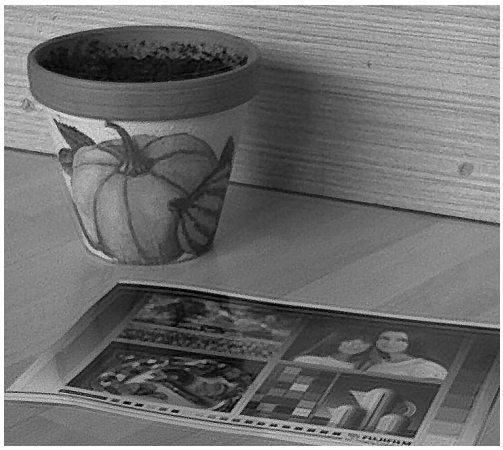


(d) restoration using depth map 11.4(d), $\lambda_u^f = 10^{-4}$

(e) restoration using depth map 11.4(f), $\lambda_u^f = 10^{-3}$

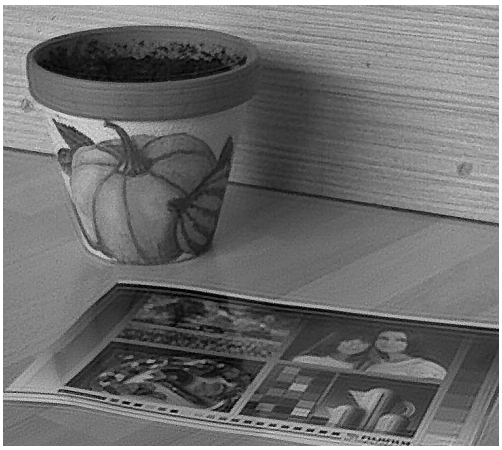
(f) restoration using depth map 11.4(f), $\lambda_u^f = 10^{-4}$

Figure 11.5: Results of restoration using Algorithm I. For final minimization we used depth maps from the right column of Fig. 11.4. For comparison, see ground truth image Fig. 11.1(c). Iteration scheme $20 \times (8 + 10) + 5 \times 20$.



(a) restoration using depth map 11.4(b)

(b) restoration using depth map 11.4(d)



(c) restoration using depth map 11.4(f)

Figure 11.6: Results of restoration using Algorithm I for $\lambda_u^f = 3 \times 10^{-4}$. For comparison, see ground truth image Fig. 11.1(c). Iteration scheme $20 \times (8 + 10) + 5 \times 20$.

(a) out-of-focus image, 730 × 650 pixels, $F/5.0$

(b) another out-of-focus image of the same scene, $F/6.3$

(c) ground truth image taken with $F/16$

Figure 11.7: The flowerpot scene was taken twice from tripod. The only camera setting that changed was aperture. For comparison, the third image was taken with large f-number to achieve large depth of focus. It will serve as a "ground truth" (color version of Fig. 11.1).

(a) restoration using depth map Fig. 11.4(b), $\lambda_u^f = 10^{-5}$

(b) restoration using depth map Fig. 11.4(b), $\lambda_u^f = 10^{-4}$

(c) restoration using depth map Fig. 11.4(d), $\lambda_u^f = 10^{-5}$

(d) restoration using depth map 11.4(d), $\lambda_u^f = 10^{-4}$

(e) restoration using depth map Fig. 11.4(f), $\lambda_u^f = 10^{-5}$

(f) restoration using depth map Fig. 11.4(f), $\lambda_u^f = 10^{-4}$

Figure 11.8: Color restoration using depth maps Fig. 11.4(f), Fig. 11.4(d) and Fig. 11.4(b) computed by Algorithm I.

11.2 Motion blur (I)

Camera motion blur is another frequent type of blur we meet when working with digital cameras. In this thesis, we present two experiments with motion blurred images. Both were taken from the digital camera mounted on a framework that limits motion or vibrations to one vertical plane.

The first experiment documents behavior of our algorithms for images blurred by one-dimensional harmonic motion of the camera. The scene is chosen relatively simple but so as the extent of blur varies significantly throughout the image. The second experiment was set up to show limitations of the proposed algorithms. The scene is much more complex with a lot of small details and there are many places where the depth changes rapidly. Also the camera motion is much more complex, constrained only by the condition that the camera cannot rotate.

Note that the structure of both experiments is similar to the experiment with out-of-focus images.

We took two color images Fig. 11.15(a) and 11.15(b) from a camera mounted on the device vibrating approximately in horizontal (a) and vertical (b) directions, both with shutter speed $T = 5s$. To achieve large depth of focus, we set f-number to $F/16$. The third image Fig. 11.16(b) was taken without vibrations and we use it as ground truth. Algorithm I works basically with intensity (monochromatic) images. For this purpose, we use red channel Fig. 11.9.

We work with model (5.3) that scales PSF according to the distance from camera. Unlike out-of-focus blur, we do not have any prior estimate of prototype PSF $\mathbf{h}_0$. In this case, it is equivalent to the knowledge of the PSF for at least one distance from camera. For this purpose, we took two small sections Fig. 11.10(a) from the right part of the input images and computed PSFs Fig. 11.10(b) using space-invariant blind restoration algorithm [2] (with parameters $\lambda = 1000$, $\varepsilon = 0.1$, $\gamma = 10$, support of both PSFs was set to 11×11 pixels). These PSFs will serve as the prototype PSFs $\mathbf{h}_0$ from relation (5.3).

To show the space-variance of the blur in our images we took another sections Fig. 11.10(c) from the image center (bear in waterfall) and computed PSFs Fig. 11.10(d), again using the method [2]. We can see that the extent of defocus is about half compared to the PSFs Fig. 11.10(b) which is in agreement with our model (5.3).

Similarly to the previous experiment, we will demonstrate that if the image contains areas with as much varying degree of blur as in our experiment, the space-invariant restoration methods (that is methods that use one PSF for the whole image) cannot yield satisfactory results. Let us look at

Fig. 11.11, where we can see deconvolutions using PSFs from Fig. 11.10(b) and Fig. 11.10(d). In addition, Fig. 11.11(c) contains the result of one of the best known blind space-invariant restoration method [2] applied on the whole images. In all the cases the images contain strong artifacts in the areas where the PSFs do not fit. Thus, in Fig. 11.11(a) the bear in the image center is not well restored, in Fig. 11.11(b) the juice box remains somewhat out-of-focus and in Fig. 11.11(c) there are visible artifacts in the whole image.

Now, we will present the application of Algorithms III and I on blurred images Fig. 11.9(a) and (b).

First, we applied Algorithm III to get an initial estimate of depth map Fig. 11.12(b). In the algorithm, we averaged the error by 7×7 window. Afterwards, the result was smoothed by 11×11 median filter. Again, the question arises whether it is possible to use this depth map estimate directly for restoration. The answer is that in most situations it results in significant artifacts in the whole area of the image, as shown in Fig. 11.12(a).

Next, we applied the iteration procedure from p. 41, that is the alternating minimization of functional (6.1). Figures 11.13 and 11.14 show depth maps and restored images for three different levels of depth map regularization. In all 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. We have seen in the previous experiment that the image regularization constant has no much influence on the produced depth map. The influence on the restored image we saw in Fig. 11.5 and 11.6 and is well described in literature [33]. Analogously to previous experiment, we have got visually almost undistinguishable results for different depth maps. In the following experiment we will show that in case of more complex scene we must choose the depth map regularization constant more carefully.

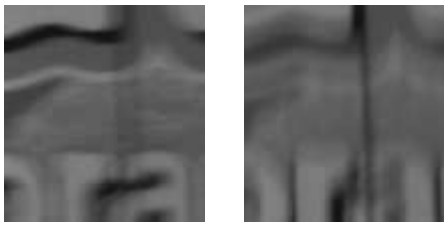
Figure 11.15 shows color originals of motion blurred images from Fig. 11.9. The same way as in the first experiment we employed least squares restoration with color regularization term (6.11). Figure 11.16(a) gives result of restoration for image regularization constant $\lambda_u^f = 10^{-4}$ using depth map Fig. 11.13(a). Results for the other two depth maps Fig. 11.13(b) and 11.13(c) were visually undistinguishable and we withhold them. For final non-blind restoration we used 5×25 iterations.

(a) image blurred by periodic horizontal motion

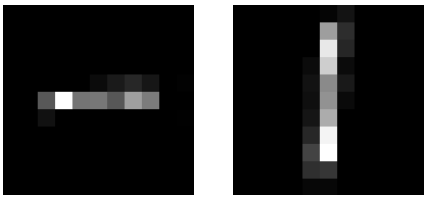
(b) image blurred by periodic vertical motion

(c) ground truth image

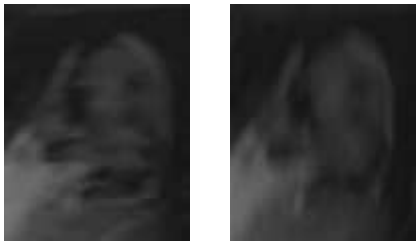
Figure 11.9: Red channel of RGB images (870×580 pixels) from Fig. 11.15. We took two images from the camera mounted on device vibrating in horizontal (a) and vertical (b) directions. For both images, the shutter speed was set to 5s and aperture to $F/16$. For comparison, the third image was taken without vibrations serving as a "ground truth".



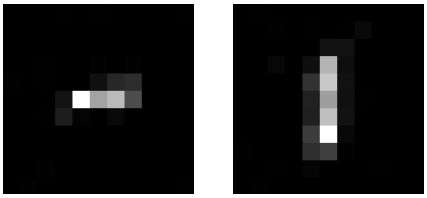
(a) sections of images Fig. 11.9(a) and (b) used for the estimate of PSFs were taken from areas at the juice box on the right (50×54 pixels, $5\times$ enlarged)



(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 the bear images (c)

Figure 11.10: Algorithm I 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. 11.9(a) and (b) where the distance from camera was constant and computed PSFs (b) using blind space-invariant restoration method [2]. For comparison we computed PSFs (d) from sections (c) taken from the image center. We can see that in agreement with our model, the PSFs (d) are a scaled down version of PSFs (b).

(a) deconvolution using PSFs from Fig. 11.10(b), TV regularization, $\lambda_u = 10^{-4}$

(b) deconvolution using PSFs from Fig. 11.10(d), TV regularization, $\lambda_u = 10^{-4}$

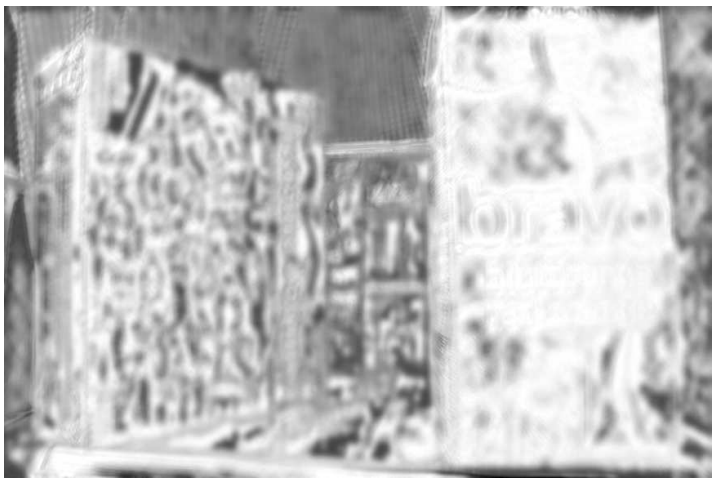
(c) Result of blind space-invariant restoration method [2]. This method belongs to the best known methods for space-invariant restoration.

Figure 11.11: Illustration of the fact that we cannot use space-invariant restoration methods. In all cases, using only one PSF for the whole image results in clearly visible artifacts.

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

(b) depth map got by Algorithm III, error averaging by 7×7 window, result smoothed by 11×11 median filter

Figure 11.12: 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.



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

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

(c) $\lambda_w = 10^{-4}$

Figure 11.13: Depth maps produced by Algorithm I for three different levels of depth map regularization. In all cases minimization started from depth map Fig. 11.12(b) with image regularization constant $\lambda_u = 10^{-4}$.

(a) restoration using depth map 11.13(a)

(b) restoration using depth map 11.13(b)

(c) restoration using depth map 11.13(c)

Figure 11.14: Results of restoration using Algorithm I. We can see that we can get good restoration for different degrees of depth map regularization. For comparison, see ground truth image Fig. 11.9(c). In all cases $\lambda_u^f = 10^{-4}$. Iteration scheme $20 \times (8 + 10)$.

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

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

Figure 11.15: We took two images from the camera mounted on device vibrating in horizontal and vertical directions. For both images, the shutter speed was set to 5s and aperture to $F/16$ (color version of Fig. 11.9).

(a) restoration using depth map Fig. 11.13(a), $\lambda_u^f = 10^{-4}$

(b) ground truth image

Figure 11.16: Result of the color version of Algorithm I. For comparison, the third image was taken by motionless camera serving as a "ground truth". In the case of restored image (a) we used simple white-balance algorithm to make the image more realistic.

11.3 Motion blur (II)

In the third real experiment, we tested the proposed algorithms on images blurred by a complex camera motion blur. As we mention in the description of the previous experiment, it 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 places where the depth changes rapidly. Also the camera motion is more complex. The structure of experiment is again similar to the previous one.

The color images Fig. 11.23(a) and 11.23(b) were taken from the same device limiting motion and vibrations to one vertical plane. We made the framework quiver by a random impulse of hand and took two images in a rapid sequence. This time the shutter speed was set to $T = 1.3s$. To achieve large depth of focus, we used f-number $F/22$. The third image Fig. 11.24(c) was taken without vibrations and we use it as ground truth. In the monochromatic version of the algorithms we work with green channel Fig. 11.17.

The same way as in the previous experiment, we computed PSFs for one distance from camera using algorithm [2] (with parameters $\lambda = 1000$, $\varepsilon = 0.1$ and $\gamma = 10$ for 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. 11.18(a). Resulting masks are in Fig. 11.18(b). For comparison, we cropped sections Fig. 11.18(c) and computed masks Fig. 11.18(d) corresponding to the upper-right corner of the LCD screen in the background part of the image. Again, we can see that our model (5.3) approximately holds.

The use of space-invariant methods Fig. 11.19 is again not acceptable.

Thus, we applied Algorithm III to get an estimate of depth map Fig. 11.20(a). Again, this estimate is not suitable for restoration as illustrated in Fig. 11.20(b).

However, this depth map can be used as the initial estimate for Algorithm I. Figures 11.21 and 11.22 give results for two degrees of depth map regularization. In the previous experiments we saw that the image regularization constant has no much influence on the produced depth map and we indicated sufficiently the influence of this constant on the restored images. Here in both cases we used image regularization constant $\lambda_u = 10^{-3}$ for the alternating minimization and $\lambda_u^f = 10^{-4}$ for final 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 causes visible ringing effects on the places, where distance from camera suddenly changes. Sometimes we must take a compromise according to the situation.

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 estimate. Fortunately, at least in these experiments, it does not affect restoration seriously.

Figure 11.23 shows color originals of motion blurred images from Fig. 11.17. Again, we employ constrained least squares restoration with color regularization term (6.11). Figure 11.24 gives results of restoration using two depth maps obtained using different levels of regularization $\lambda_w = 10^{-6}$ and $\lambda_w = 5 \times 10^{-6}$. Color images pronounce artifacts present in intensity images. Again, we can see wave-like artifacts on the wall in the background if we use smaller value of depth map regularization constant. On the other hand, if we use higher degree of regularization, there are visible ringing effects on the edges, for example at the blossoms near the right edge of the LCD screen. In addition, in either case, we can observe color artifacts present especially on thin objects such as grass-blades. This could be probably removed only by taking into account occlusions present at object edges [65, 66, 67].

11.4 Summary

In this chapter, we have demonstrated behavior of the proposed algorithm on real images. We presented three experiments, one for out-focus blur and two for camera motion blur.

We saw that if the image contains areas with as much varying degree of blur as in our experiments, the space-invariant restoration methods cannot yield satisfactory results, which approved the need for space-variant methods.

Next, we applied Algorithm III to get a rough estimate of depth maps. Experiments showed that it is not possible to use this estimate directly for restoration as it resulted in visible artifacts in the whole area of the image.

We also showed the influence of regularization parameters on the result of minimization. We have seen that the image regularization constant λ_u controls the trade-off between sharpness of the image and noise reduction but has no much influence on the produced depth map. Too much depth map regularization may cause ringing effects on the edges but in turn, if we use too little regularization, the algorithm does not smooth sufficiently areas without texture. For both constants, we must take a compromise according to the character of the scene.

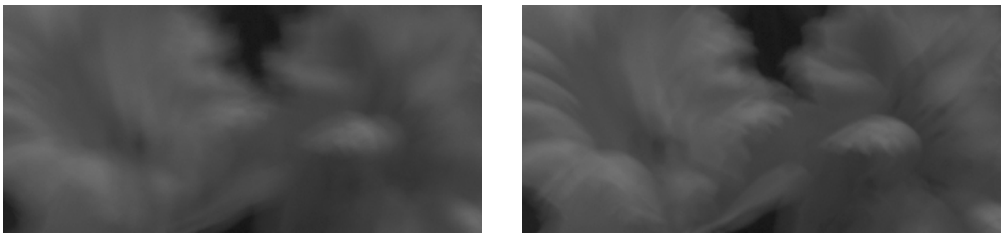
The color experiments confirmed possibility to extend the Algorithm I to color images. In addition, the use of color regularization term (6.11) allowed to use less regularization and consequently to get even sharper images,

because the regularization term suppresses noise using information from all three RGB channels.

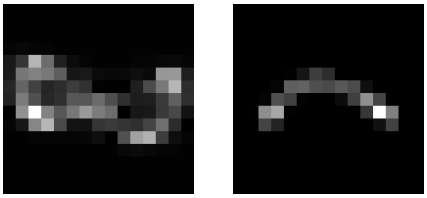
(a) image blurred by space-variant motion blur (first image)

(b) image blurred by space-variant motion blur (second image)

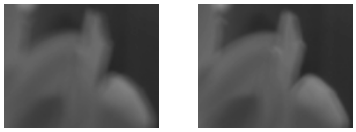
Figure 11.17: Red channel of Fig. 11.23. We took two images from the camera mounted on vibration framework limiting motion to one vertical plane. For both images, the shutter speed was set to 1.3s and aperture to $F/22$. Image size 800×500 pixels.



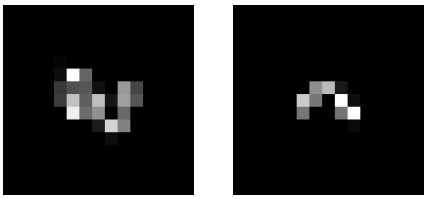
(a) sections of images Fig. 11.17(a) and (b) used for the estimate of PSFs taken from the foreground part of the image (167×353 pixels, $3\times$ enlarged)



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



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



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

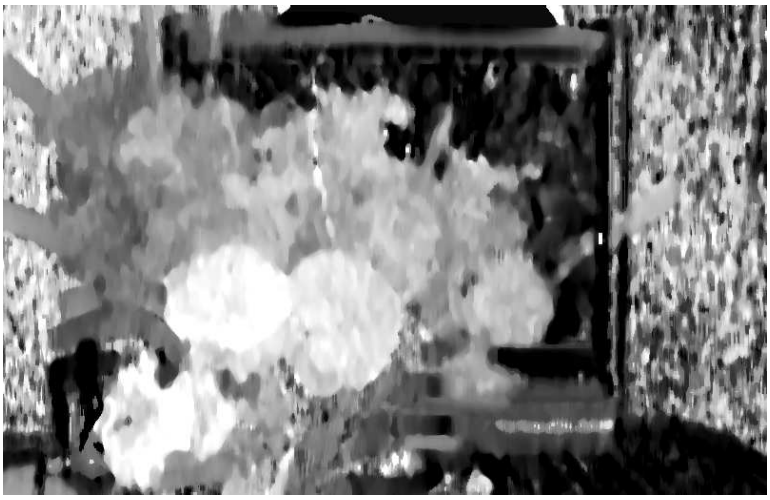
Figure 11.18: Algorithm I needs an estimate of PSF for at least one distance from camera. We took a central part of the images Fig. 11.17(a) and (b) where the degree of blur was approximately constant and computed PSFs (b) using blind space-invariant restoration method [2]. For comparison we computed PSFs (d) from background sections (c). We can see that in agreement with our model, the PSFs (d) are a scaled down version of PSFs (b).

(a) deconvolution using PSFs from Fig. 11.18(b), TV regularization, $\lambda_u = 10^{-4}$

(b) deconvolution using PSFs from Fig. 11.18(d), TV regularization, $\lambda_u = 10^{-4}$

(c) Result of blind space-invariant restoration method [2]. This method belongs to the best known methods for space-invariant restoration.

Figure 11.19: Illustration of the fact that we cannot use space-invariant restoration methods. In all cases, using only one PSF for the whole image results in clearly visible artifacts.



(a) depth map obtained by Algorithm III, error averaging by 11×11 window, result subsequently smoothed by 11×11 median filter

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

Figure 11.20: Illustration of the fact that we cannot use simple depth recovery methods directly for restoration. We can see many artifacts in the whole image.

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

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

Figure 11.21: Depth maps produced by Algorithm I for two different levels of Tikhonov depth map regularization. In both cases, the alternating minimization was initialized with depth map Fig. 11.20(a).

(a) restoration using depth map 11.21(a)

(b) restoration using depth map 11.21(b)

(c) ground truth image

Figure 11.22: Results of restoration using Algorithm I. We can see that lesser depth map regularization (a) may result in artifacts in the areas of weak texture (wall in the background). Higher degree of regularization (b) caused artifacts on the edges (edge between blossoms near the right edge of the LCD screen). For comparison, the third image was taken by motionless camera serving as a "ground truth".

(a) image blurred by camera motion

(b) another image of the same scene blurred by different camera motion

Figure 11.23: We took two images from the camera mounted on the framework limiting motion to one vertical plane. The shutter speed was set to the same value 1.3s and aperture to $F/22$ (color version of Fig. 11.17). Image size 800×500 pixels.

(a) restoration using less regularized depth map 11.21(a), $\lambda_w = 10^{-6}$

(b) restoration using smoother depth map 11.21(b), $\lambda_w = 5 \times 10^{-6}$

(c) ground truth image

Figure 11.24: Result of the color extension of Algorithm I using regularization term (6.11). Notice the color artifacts on grass-blades. For comparison, the third image was taken by motionless camera as a "ground truth".

Chapter 12

Conclusion

In this thesis, we have covered the problems of space-variant image restoration and depth map estimation from two or more blurred images of the same scene in situations, where the degree of blur depends on the distance of the objects from camera. It includes out-of-focus blur, camera motion blur or both simultaneously.

12.1 Evaluation

This section summarizes the results presented in the thesis and progress we achieved with respect to previously published methods.

We developed three algorithms, related to the goals of this thesis, all of them working for both out-of-focus and camera motion blur.

The main result, presented as Algorithm I, is a variational method for image restoration and simultaneous estimation of the depth map. In comparison with other variational methods based on the same idea [7, 8, 9], it can be applied for much broader class of PSFs. For the best of our knowledge, it is the only method working for complex camera motion and optical aberrations. In addition, Algorithm I is robust with respect to noise and solves successfully the problem with non-convexity of the functional.

Algorithm I needs a reasonable initial estimate of depth map. For this purpose, we modified filter based DFD algorithm [1] to work for a broader class of symmetrical PSFs (Algorithm II). The main virtue of this algorithm is speed, its computation consists of only two convolutions. On the other hand, it requires careful calibration and often fails at places of weak texture.

Next, we proposed more general algorithm for estimation of depth maps (Algorithm III), working for arbitrary blurs at the expense of higher time consumption. It also proved to be robust to noise. The principal improve-

ment with respect to existing DFD and depth from motion blur methods is that it places no restrictions on the PSF. Compared to [6] is much simpler to implement.

Besides, we have proposed an extension of Algorithms I and III to color images. As demonstrated in the experimental section, the joint regularization significantly reduces noise contained in individual channels and allows for sharper results.

To clarify expressions derived in Algorithm I, we introduced notation for two linear operators, "space-variant convolution" and "space-variant correlation", which generalize several frequent image processing operations. If implemented in a graphic card or signal processor, they could speed up many video and image processing applications.

The goals of this thesis, specified in Chapter 1.4, have been met. All of the algorithms are able to work with only two input images (goal 1). They place no restrictions on the structure of the scene (goal 2), with the exception of extreme cases, such as very complex objects with holes of dimensions close to width of a pixel. In turn, objects lacking texture often make no harm to restoration since simply there is "nothing to restore". Algorithms I and III put no constraints on the shape of PSF (goal 3) and even Algorithm II works for broader class of PSFs than most older methods. As a consequence, our algorithms can be used for a class of non-trivial motion blurs (goal 4). All the presented methods can be implemented using only a small set of linear operations (goal 5).

12.2 Future work and applications

Probably the most interesting topics for future research stem from a promising application of Algorithm I—reduction of camera shake. In theory, Algorithm I can be extended to arbitrary camera motion. In completely general case, however, an issue arises how to generate and efficiently store all the PSFs, which differ for all combinations of depths, coordinates in the field of view and camera parameters (focal length, plane of focus, aperture). In addition, if we do not know the data from inertial sensors describing camera motion, an interesting and difficult problem arises, how to reconstruct the course of camera motion from the blurred images itself. In all these problems, a thorough analysis of constrains valid for the motion of handheld camera would be very helpful.

Next, the proposed algorithms neglect occlusions at object edges on account of finite lens aperture. The Algorithm I could probably be extended

to the more precise model of blurring described in [65, 66, 67].

As for applications, successful implementation of Algorithm I for general camera motion could result in very powerful anti-shake systems, especially in combination with existing optical stabilizers. Of course, cameras would have to provide information about the true motion they obtain from the inertial sensors of the stabilizer.

Finally, we should mention an interesting application of proposed algorithms in photography, changing of the depth of focus. For this purpose, we can apply Algorithm I directly on two images taken from a tripod.

Appendix A

Proofs related to Algorithm I

For the convenience of the reader we repeat the body of the propositions and corollaries before proving them.

Proposition 1. *Gradients of the error term* Φ *in subspaces corresponding to image* $\mathbf{u}$ *and depth map represented by* $\mathbf{w}$ *can be expressed as*

$$\frac{\partial \Phi}{\partial \mathbf{u}} = \sum_{p=1}^{P} \mathbf{e}_p \circledast_v h_p(\mathbf{w}) = \sum_{p=1}^{P} \mathbf{u} *_v h_p(\mathbf{w}) \circledast_v h_p(\mathbf{w}) - \mathbf{z}_p \circledast_v h_p(\mathbf{w}), \quad \text{(A.1)}$$

$$\frac{\partial \Phi}{\partial \mathbf{w}} = \mathbf{u} \sum_{p=1}^{P} \mathbf{e}_p \circledast_v \frac{\partial h_p(\mathbf{w})}{\partial \mathbf{w}}, \quad \text{(A.2)}$$

where $\frac{\partial h_p(\mathbf{w})}{\partial \mathbf{w}}[x, y; s, t]$ *is the derivative of the mask related to image point* (x, y) *with respect to the value of* $\mathbf{w}(x, y)$.

Proof. Proofs are straightforward. The basic idea is to find the Riesz representation for directional derivatives. It exists as the derivatives are bounded linear operators. The representing function we have found is nothing else than the wanted gradient (Fréchet derivative).

To show the first equality, recall that $\Phi = \sum_1^P \Phi_p$. Let us denote $h_p(\mathbf{w})$ as $\mathbf{h}$. We can treat it as a constant for the moment. The directional derivative of Φ_p in a direction $\mathbf{g}$ is

$$\frac{\partial}{\partial \lambda} \Phi_p(\mathbf{u} + \lambda \mathbf{g}, \mathbf{h})_{\,|\lambda=0} = \int_{\mathcal{D}} \mathbf{e}_p(x, y)(\mathbf{g} *_v \mathbf{h})[x, y] \, dxdy \quad \text{(A.3)}$$

$$= \int_{\mathcal{D}} \mathbf{e}_p(x, y) \left[\int_{\Omega} \mathbf{g}(x - s, y - t)\mathbf{h}(x - s, y - t; s, t) ds dt \right] dxdy \quad \text{(A.4)}$$

$$= \int_{\mathcal{D}} \mathbf{g}(x', y') \left[\int_{\mathcal{D}} \mathbf{e}_p(x, y)\mathbf{h}(x', y'; x - x', y - y') dxdy \right] dx'dy', \quad \text{(A.5)}$$

which after back substitution $s' = x - x'$ and $t' = y - y'$ yields

$$\int_{\mathcal{D}} \mathbf{g}(x', y') \left[\int_{\Omega} \mathbf{e}_p(x' - s', y' - t') \mathbf{h}(x', y'; -s', -t') ds' dt' \right] dx' dy'. \tag{A.6}$$

It is exactly the Riesz representation of directional derivative (A.3). Since $\Phi = \sum_1^P \Phi_p$, the inner integral gives the right side of (A.1).

To prove (A.2), in parallel to the proof of (A.1), we express directional derivative

$$\frac{\partial}{\partial \lambda} \Phi_p(\mathbf{u}, \mathbf{w} + \lambda \mathbf{g}) \mid_{\lambda=0} = \tag{A.7}$$

$$= \int_{\mathcal{D}} \mathbf{e}_p(x, y) \tag{A.8}$$

$$\cdot \int_{\Omega} \mathbf{u}(x - s, y - t) \frac{\partial h_p(\mathbf{w}(x - s, y - t) + \lambda \mathbf{g}(x - s, y - t))}{\partial \lambda} [s, t]_{\lambda=0} \, ds dt dx dy. \tag{A.9}$$

Using the chain rule for each particular $[s, t]$ it equals

$$\int_{\mathcal{D}} \mathbf{e}_p(x, y) \left[\int_{\Omega} \mathbf{u}(x - s, y - t) \mathbf{g}(x - s, y - t) \frac{\partial h_p(\mathbf{w}(x - s, y - t))}{\partial w} [s, t] \, ds dt \right] dx dy \tag{A.10}$$

and after substitution $x' = x - s$, $y' = y - t$, $s' = -s$ and $t' = -t$

$$\iint_{\mathcal{D} \times \Omega} \mathbf{e}_p(x' - s', y' - t') \left[\mathbf{u}(x', y') \mathbf{g}(x', y') \frac{\partial h_p(\mathbf{w}(x', y'))}{\partial w} [-s', -t'] \, ds' dt' \right] dx' dy'. \tag{A.11}$$

Now, by getting $\mathbf{g}$ and $\mathbf{u}$ out of the inner integral we get the Riesz representation of (A.7)

$$\int_{\mathcal{D}} \mathbf{g}(x', y') \left[\mathbf{u}(x', y') \int_{\Omega} \mathbf{e}_p(x' - s', y' - t') \frac{\partial h_p(\mathbf{w}(x', y'))}{\partial w} [-s', -t'] \, ds' dt' \right] dx' dy'. \tag{A.12}$$

The inner bracket of (A.12) is exactly the right side of (A.2) for image p. □

Appendix B

Proofs related to Algorithm II

Proposition 2. *Let* $\mathbf{u}(x, y)$ *be a third-order polynomial*[1] *of two variables and* $\mathbf{z}_i = \mathbf{u} * \mathbf{h}_i$, $i = 1, 2$, *where* $\mathbf{h}_i$ *are energy preserving (*$\int \mathbf{h} = 1$*) PSFs symmetric about axes* x, y *and both axes of quadrants. Then*

$$\sigma_2^2 - \sigma_1^2 = 2\frac{\mathbf{z}_2 - \mathbf{z}_1}{\nabla^2\left(\frac{\mathbf{z}_1+\mathbf{z}_2}{2}\right)}, \tag{B.1}$$

where σ_1^2, σ_2^2 *are the second moments of* $\mathbf{h}_1$ *and* $\mathbf{h}_2$ *and* ∇^2 *is the symbol for Laplacian.*

Proof. We follow the technique used in [1] for Gaussian masks. It is based on the idea that convolution of finite-order polynomials with a mask $\mathbf{h}$ can be expressed using derivatives of the polynomial and moment of the mask. Let the third-order polynomial $\mathbf{u}$ is given by

$$\mathbf{u}(x, y) = \sum_{m=0}^{3}\sum_{n=0}^{3-m} a_{m,n}x^m y^n \tag{B.2}$$

and $\mathbf{z} = \mathbf{u} * \mathbf{h}$. Then the derivatives of $\mathbf{u}$ of order higher than three vanish and we can write

$$\mathbf{z}(x, y) = \sum_{0<=m+n<=3} \frac{(-1)^{m+n}}{m!n!}\frac{\partial^m\partial^n}{\partial x^m\partial y^n}\mathbf{u}(x, y)\ \mathbf{h}_{m,n}, \tag{B.3}$$

where the moments $\mathbf{h}_{m,n}$ of the point spread function $\mathbf{h}$ are defined by

$$\mathbf{h}_{m,n} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^m y^n \mathbf{h}(x, y)dxdy. \tag{B.4}$$

[1] Two-dimensional third-order polynomial is a polynomial $P(x, y) = \sum_{m=0}^{3}\sum_{n=0}^{3-m} a_{m,n}x^m y^n$.

Now we make use of the fact that almost all the moments of $\mathbf{h}$ up to third order are zero, which eliminates almost all terms of (B.3).

Almost all the moments of $\mathbf{h}$ up to the third order are zero, which eliminates almost all terms of (B.3). Since $\mathbf{h}$ is symmetrical about the x and y axes, $\mathbf{h}_{1,0} = \mathbf{h}_{0,1} = \mathbf{h}_{1,2} = \mathbf{h}_{2,1} = \mathbf{h}_{3,0} = \mathbf{h}_{0,3} = \mathbf{h}_{1,1} = 0$.

There are only three nonzero moments left, namely the zeroth moment $\mathbf{h}_{0,0}$ and the second moments. We know that $\mathbf{h}_{0,0} = 1$ as $\mathbf{h}$ preserves energy. Thus,

$$\mathbf{z}(x,y) = \mathbf{u}(x,y) + \frac{1}{2!}\frac{\partial^2 \mathbf{u}}{\partial x^2}\,\mathbf{h}_{2,0} + \frac{1}{2!}\frac{\partial^2 \mathbf{u}}{\partial y^2}\,\mathbf{h}_{0,2}. \tag{B.5}$$

In addition, because of symmetry about the axes of quadrants, we get $\mathbf{h}_{2,0} = \mathbf{h}_{0,2} = \sigma^2$ and consequently

$$\mathbf{z}(x,y) = \mathbf{u}(x,y) + \frac{\sigma^2}{2}\left(\frac{\partial^2 \mathbf{u}}{\partial x^2} + \frac{\partial^2 \mathbf{u}}{\partial y^2}\right) = \mathbf{u}(x,y) + \frac{\sigma^2}{2}\nabla^2 \mathbf{u}. \tag{B.6}$$

Since $\mathbf{u}$ is a third-order polynomial, applying $\frac{\partial^2}{\partial x^2}$ and $\frac{\partial^2}{\partial y^2}$ on (B.6) gives

$$\frac{\partial^2 \mathbf{z}}{\partial x^2} = \frac{\partial^2 \mathbf{u}}{\partial x^2} \text{ and } \frac{\partial^2 \mathbf{z}}{\partial y^2} = \frac{\partial^2 \mathbf{u}}{\partial y^2} \tag{B.7}$$

which after substitution to (B.6) gives sort of a deconvolution formula

$$\mathbf{u}(x,y) = \mathbf{z}(x,y) - \frac{\sigma^2}{2}\nabla^2 \mathbf{z}. \tag{B.8}$$

As $\mathbf{z}_1$ and $\mathbf{z}_2$ originated in the same image $\mathbf{u}$, according to (B.7) $\nabla^2 \mathbf{z}_1 = \nabla^2 \mathbf{z}_2$. In practice it does not hold exactly (because of noise for example) and it is better to take the average value

$$\frac{\nabla^2 \mathbf{z}_1 + \nabla^2 \mathbf{z}_2}{2} = \nabla^2\left(\frac{\mathbf{z}_1 + \mathbf{z}_2}{2}\right), \tag{B.9}$$

which, using (B.8) for $\mathbf{z}_1$ and $\mathbf{z}_2$, gives (B.1).

□

Proposition 3. *Let $\mathbf{u}(x,y)$ be a third-order polynomial[1] of two variables and $\mathbf{z}_i = \mathbf{u} * \mathbf{h}_i$, $i = 1, 2$, where $\mathbf{h}_i$ are energy preserving ($\int \mathbf{h} = 1$) one-dimensional even PSFs oriented in the direction of the x-axis. Then*

$$\sigma_2^2 - \sigma_1^2 = 2\frac{\mathbf{z}_2 - \mathbf{z}_1}{\frac{\partial^2}{\partial x^2}\left(\frac{\mathbf{z}_1 + \mathbf{z}_2}{2}\right)}, \tag{B.10}$$

where σ_1^2, σ_2^2 are the second moments of $\mathbf{h}_1$ and $\mathbf{h}_2$.

Proof. The proof of Proposition 3 is similar to the proof of Proposition 2. Again, we assume third-order polynomials $\mathbf{u}$ and $\mathbf{z}_i = \mathbf{u} * h_i$ satisfying (B.3).

Again, almost all the moments of $\mathbf{h}$ up to the third order are zero, which eliminates almost all terms of (B.3). Indeed, $\mathbf{h}_{m,n} = 0$ for $n \neq 0$, because $\mathbf{h}(x, y) = 0$ for $y \neq 0$. Similarly, since $\mathbf{h}$ is symmetrical about the y-axis, $\mathbf{h}_{1,0} = \mathbf{h}_{3,0} = 0$. There are only two nonzero moments left, namely $\mathbf{h}_{0,0} = 1$ because $\mathbf{h}$ preserves energy and the second moment $\mathbf{h}_{2,0} = \sigma^2$.

Thus,

$$\mathbf{z}(x, y) = \mathbf{u}(x, y) + \frac{1}{2!}\frac{\partial^2 \mathbf{u}}{\partial x^2}\,\mathbf{h}_{2,0} = \mathbf{u}(x, y) + \frac{\sigma^2}{2}\frac{\partial^2 \mathbf{u}}{\partial x^2}. \tag{B.11}$$

Since $\mathbf{u}$ is a third-order polynomial, applying $\frac{\partial^2}{\partial x^2}$ on (B.11) gives

$$\frac{\partial^2 \mathbf{z}}{\partial x^2} = \frac{\partial^2 \mathbf{u}}{\partial x^2}, \tag{B.12}$$

which after substitution to (B.11) gives sort of a deconvolution formula

$$\mathbf{u}(x, y) = \mathbf{z}(x, y) - \frac{\sigma^2}{2}\frac{\partial^2 \mathbf{z}}{\partial x^2}. \tag{B.13}$$

As $\mathbf{z}_1$ and $\mathbf{z}_2$ originated in the same image $\mathbf{u}$, according to (B.12)

$$\frac{\partial^2 \mathbf{z}_1}{\partial x^2} = \frac{\partial^2 \mathbf{z}_2}{\partial x^2}. \tag{B.14}$$

In practice it does not hold exactly (because of noise for example) and it is better to take the average value

$$\left(\frac{\partial^2 \mathbf{z}_1}{\partial x^2} + \frac{\partial^2 \mathbf{z}_2}{\partial x^2}\right)/2 = \frac{\partial^2}{\partial x^2}\left(\frac{\mathbf{z}_1 + \mathbf{z}_2}{2}\right), \tag{B.15}$$

which, using (B.13) for $\mathbf{z}_1$ and $\mathbf{z}_2$, gives (B.10).

□

Corollary 2. *Let $\mathbf{u}(x, y)$ be a third-order polynomial of two variables and $\mathbf{z}_i = \mathbf{u} * \mathbf{h}_i$, $i = 1, 2$, where $\mathbf{h}_i$ are energy-preserving rectangular impulses of length d_i oriented in the direction of the x-axis. Then*

$$d_2^2 - d_1^2 = 24\frac{\mathbf{z}_2 - \mathbf{z}_1}{\frac{\partial^2}{\partial x^2}\left(\frac{\mathbf{z}_1 + \mathbf{z}_2}{2}\right)}. \tag{B.16}$$

Proof.

$$\sigma^2 = \mathbf{h}_{2,0} = \int_{-\infty}^{\infty}\int_{-d/2}^{d/2} x^2 \frac{\delta}{d} dxdy = \int_{-d/2}^{d/2} \frac{x^2}{d} dx \int_{-\infty}^{\infty} \delta dy \quad \text{(B.17)}$$

$$= \int_{-d/2}^{d/2} \frac{x^2}{d} = \frac{2}{d}\int_0^{d/2} x^2 = \frac{d^2}{12}. \quad \text{(B.18)}$$

□

Bibliography

[1] M. Subbarao and G. Surya, "Depth from defocus: a spatial domain approach," *International Journal of Computer Vision*, vol. 3, no. 13, pp. 271–294, 1994.

[2] F. Šroubek and J. Flusser, "Multichannel blind iterative image restoration," *IEEE Trans. Image Processing*, vol. 12, no. 9, pp. 1094–1106, Sept. 2003.

[3] R. Redondo, F. Šroubek, S. Fischer, and G. Cristobal, "Multifocus fusion with multisize windows," in *Proceedings of SPIE. Applications of Digital Image Processing XXVIII*, A. G. Tescher, Ed. SPIE, Bellingham, 2005, pp. 410–418.

[4] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision*, vol. 47, pp. 7–42, 2002.

[5] Y. Y. Schechner and N. Kiryati, "Depth from defocus vs. stereo: How different really are they?" in *Proc. Int. Conf. on Pattern Recognition*, 1998, pp. 1874–1786.

[6] P. Favaro and S. Soatto, "A geometric approach to shape from defocus," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 27, no. 3, Mar. 2005.

[7] 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. Machine Intell.*, vol. 21, no. 7, July 1999.

[8] P. Favaro, M. Burger, and S. Soatto, "Scene and motion reconstruction from defocus and motion-blurred images via anisothropic diffusion," in *ECCV 2004, LNCS 3021, Springer Verlag, Berlin Heidelberg*, T. Pajdla and J. Matas, Eds., 2004, pp. 257–269.

[9] 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*, vol. 1, no. 1, 2004, pp. 631–637.

[10] A. Kubota, K. Kodama, and K. Aizawa, "Registration and blur estimation methods for multiple differently focused images," in *Proc. IEEE Int. Conf. Image Processing*, 1999, pp. 447–451.

[11] A. Kubota and K. Aizawa, "A new approach to depth range detection by producing depth-dependent blurring effect," in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, 2001, pp. 740–743.

[12] H. Li, S. Manjunath, and S. Mitra, "Multisensor image fusion using the wavelet transform," *Graphical Models and Image Processing*, vol. 3, no. 57, pp. 235–245, 1995.

[13] M. Ben-Ezra and S. K. Nayar, "Motion-based motion deblurring," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 26, no. 6, pp. 689–698, June 2004.

[14] M. Šorel and J. Flusser, "Blind restoration of images blurred by complex camera motion and simultaneous recovery of 3D scene structure," in *Proceedings of the Fifth IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), Athens*, Dec. 2005, pp. 737–742.

[15] ——, "Simultaneous recovery of scene structure and blind restoration of defocused images," in *Proceedings of the Computer Vision Winter Workshop 2006. CVWW'06.*, O. Chum and V. Franc, Eds. Czech Society for Cybernetics and Informatics, Prague, 2006, pp. 40–45.

[16] M. Šorel, "Multichannel blind restoration of images with space-variant degradations," Research Center DAR, Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic, Prague, Tech. Rep. 2006/28, 2006.

[17] M. Šorel and J. Flusser, "Space-variant restoration of images degraded by camera motion blur," *IEEE Trans. Image Processing*, 2007, submitted.

[18] ——, "Restoration of color images degraded by space-variant motion blur," in *Proc. Int. Conf. on Computer Analysis of Images and Patterns*, 2007, submitted.

[19] ——, "Restoration of out-of-focus images with applications in microscopy," *J. Opt. Soc. Am. A*, work-in-progress.

[20] M. Šorel and J. Šíma, "Robust implementation of finite automata by recurrent RBF networks," in *Proceedings of the SOFSEM, Seminar on Current Trends in Theory and Practice of Informatics, Milovy, Czech Republic*. Berlin: Springer-Verlag, LNCS 1963, 2000, pp. 431–439.

[21] ——, "Robust RBF finite automata," *Neurocomputing*, vol. 62, pp. 93–110, 2004.

[22] A. P. Pentland, "Depth of scene from depth of field," in *Proc. Image Understanding Workshop*, Apr. 1982, pp. 253–259.

[23] ——, "A new sense for depth of field," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 9, no. 4, pp. 523–531, July 1987.

[24] J. Ens and P. Lawrence, "An investigation of methods for determining depth from focus," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, no. 2, pp. 97–108, Feb. 1993.

[25] Y. Xiong and S. A. Schafer, "Moment and hypergeometric filters for high precision computation of focus, stereo and optical flow," Carnegie Mellon University, Tech. Rep. CMU-RI-TR-94-28, Sept. 1994.

[26] M. Watanabe and S. K. Nayar, "Rational filters for passive depth from defocus," *International Journal of Computer Vision*, vol. 3, no. 27, pp. 203–225, 1998.

[27] F. Deschênes, D. Ziou, and P. Fuchs, "Simultaneous computation of defocus blur and apparent shifts in spatial domain," in *Proc. Int. Conf. on Vision Interface*, 2002, p. 236.

[28] J. S. Fox, "Range from translational motion blurring," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, June 1988, pp. 360–365.

[29] I. M. Rekleitis, "Optical flow recognition from the power spectrum of a single blurred image," in *Proc. Int. Conf. Image Processing*, vol. 3, Sept. 1996, pp. 791–794.

[30] D. L. Tull and A. K. Katsaggelos, "Regularized blur-assisted displacement field estimation," in *Proc. Int. Conf. Image Processing*, vol. 3, Sept. 1996, pp. 85–88.

[31] W. Chen, N. Nandhakumar, and W. N. Martin, "Estimating image motion from smear: A sensor system and extensions," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, p. 412, 1996.

[32] 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.

[33] M. R. Banham and A. K. Katsaggelos, "Digital image restoration," *IEEE Signal Processing Mag.*, vol. 14, no. 2, pp. 24–41, Mar. 1997.

[34] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, pp. 259–268, 1992.

[35] D. Mumford and J. Shah, "Optimal approximation by piecewise smooth functions and associated variational problems," *Comm. Pure Appl. Math.*, vol. 42, pp. 577–685, 1989.

[36] 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.

[37] J. G. Nagy and D. P. O'Leary, "Restoring images degraded by spatially variant blur," *SIAM J. Sci. Comput.*, vol. 19, no. 4, pp. 1063–1082, 1998.

[38] D. Kundur and D. Hatzinakos, "Blind image deconvolution," *IEEE Signal Processing Mag.*, vol. 13, no. 3, pp. 43–64, May 1996.

[39] R. G. Lane and R. H. T. Bates, "Automatic multichannel deconvolution," *J. Opt. Soc. Am. A*, vol. 4, no. 1, pp. 180–188, Jan. 1987.

[40] G. R. Ayers and J. C. Dainty, "Iterative blind deconvolution method and its application," *Optical Letters*, vol. 13, no. 7, pp. 547–549, July 1988.

[41] C. A. Ong and J. A. Chambers, "An enhanced NAS-RIF algorithm for blind image deconvolution," *IEEE Trans. Image Processing*, vol. 8, no. 7, pp. 988–992, July 1999.

[42] Y.-L. You and M. Kaveh, "Blind image restoration by anisothropic regularization," *IEEE Trans. Image Processing*, vol. 8, no. 3, Mar. 1999.

[43] M. Ng, R. Plemmons, and S. Qiao, "Regularization of RIF blind image deconvolution," *IEEE Trans. Image Processing*, vol. 9, no. 6, pp. 1130–1138, June 2000.

[44] 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.

[45] G. Harikumar and Y. Bresler, "Efficient algorithms for the blind recovery of images blurred by multiple filters," in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, Lausanne, Switzerland, 1996, pp. 97–100.

[46] ——, "Perfect blind restoration of images blurred by multiple filters: Theory and efficient algorithms," *IEEE Trans. Image Processing*, vol. 8, no. 2, pp. 202–219, Feb. 1999.

[47] G. B. Giannakis and R. W. Heath, "Blind identification of multichannel FIR blurs and perfect image restoration," *IEEE Trans. Image Processing*, vol. 9, no. 11, pp. 1877–1896, Nov. 2000.

[48] D. L. Angwin, "Adaptive image restoration using reduced order model based Kalman filters," Ph.D. dissertation, Rensselaer Polytech. Inst., Troy, NY, Dept. Elect. Eng. Comput. Sci., 1989.

[49] R. L. Lagendijk and J. Biemond, "Block-adaptive image identification and restoration," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, May 1991, pp. 2497–2500.

[50] 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, June 1991, pp. 146–156.

[51] A. Rav-Acha and S. Peleg, "Restoration of multiple images with motion blur in different directions," in *Fifth IEEE Workshop on Applications of Computer Vision*, Dec. 2000, pp. 22–28.

[52] A. N. Rajagopalan, S. Chaudhuri, and U. Mudenagudi, "Depth estimation and image restoration using defocused stereo pairs," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 26, no. 11, pp. 1521–1525, Nov. 2004.

[53] W. T. Welford, *Aberrations of Optical Systems*. Adam Hilger, Bristol, Philadelphia and New York, 1986, reprint 1991.

[54] D. J. Heeger and A. D. Jepson, "Subspace methods for recovering rigid motion," *International Journal of Computer Vision*, vol. 7, no. 2, pp. 95–117, 1992.

[55] I. Klapp and Y. Yitzhaky, "Angular motion point spread function model considering aberrations and defocus effects," *J. Opt. Soc. Am. A*, vol. 23, no. 8, pp. 1856–1864, Aug. 2006.

[56] A. Tikhonov and V. Arsenin, *Solution of ill-posed problems*. New York: Wiley, 1977.

[57] 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 Processing*, vol. 1, no. 3, pp. 322–336, July 1992.

[58] D. Tschumperlé and R. Deriche, "Vector-valued image regularization with pdes: A common framework for different applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 4, pp. 506–517, 2005.

[59] ——, "Diffusion PDE's on vector-valued images," *IEEE Signal Processing Mag.*, vol. 19, no. 5, pp. 16–25, Sept. 2002.

[60] P. Meer and I. Weiss, "Smoothed differentiation filters for images," *Journal of Visual Communication and Image Representation*, vol. 3, pp. 58–72, 1992.

[61] 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.

[62] B. R. Hunt, "The application of constrained least-squares estimation to image restoration by digital computer," *IEEE Transactions on Computers*, vol. C-22, pp. 805–812, Sept. 1973.

[63] N. Nguyen, P. Milanfar, and G. Golub, "Efficient generalized cross-validation with applications to parametric image restoration and resolution enhancement," *IEEE Trans. Image Processing*, vol. 10, no. 9, pp. 1299–1308, 2001.

[64] B. Zitová and J. Flusser, "Image registration methods: a survey," *Image and Vision Computing*, vol. 11, no. 21, pp. 977–1000, 2003.

[65] N. Asada, H. Fujiwara, and T. Matsuyama, "Seeing behind the scene: Analysis of photometric properties of occluding edges by the reversed projection blurring model," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, no. 2, pp. 155–167, Feb. 1998.

[66] S. Bhasin and S. Chaudhuri, "Depth from defocus in presence of partial self occlusion," in *Proc. IEEE Int. Conf. Computer Vision*, vol. 1, July 2001, pp. 488–493.

[67] 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*, vol. 2, 2003, pp. 579–586.