Chapter 1

Restoration in the presence of unknown spatially varying blur

MICHAL ŠOREL

Department of Image Processing
Institute of Information Theory and Automation
Academy of Sciences of the Czech Republic
Prague, Czech republic
sorel@utia.cas.cz

FILIP ŠROUBEK

Department of Image Processing
Institute of Information Theory and Automation
Academy of Sciences of the Czech Republic
Prague, Czech republic
sroubekf@utia.cas.cz

1.1 Introduction

The last two decades brought significant progress in the development of efficient methods for classical deconvolution and super-resolution problems in both the single and multi-image scenarios. Most of these methods work with blurs modelled by convolution, which assumes that the properties of blur are the same

in the whole image (space-invariant blur). Unfortunately, in practice, the blur is typically spatially variant. The most common types of space-variant blur are defocus, optical aberrations and motion blur caused by either camera motion or motion of objects.

Extension of deconvolution methods to spatially varying blur is not straightforward. What makes such restoration problems more challenging than in the case of the space-invariant blur, is a much higher number of unknowns that must be estimated. Consequently, the solution is ill-posed and requires additional constrains that must be chosen depending on the type of blur we wish to suppress. The requirement to remove only certain types of blur while keeping others is surprisingly common. A typical example is the removal of motion blur from portrait pictures taken in low-light conditions while keeping a small depth of focus. Similarly, it is usually desirable to remove optical aberrations but we may wish to preserve motion blur conveying the sensation of speed.

In the last few years, despite the complexity of space-variant deblurring, we can observe an increasing effort in this direction of research, including difficult problems such as the blur dependent on the depth of scene or several independently moving objects. In this chapter, we give an overview of the current state of the art in the space-variant restoration, addressing the latest results in both deblurring and super-resolution.

The chapter is divided into two main parts. In Sec. 1.2, we describe mathematical models used for spatially varying blur and basic restoration approaches connected with these models. Our purpose is not to give a complete survey of all known algorithms, instead we just briefly outline the models and point out interesting papers that appeared in the last several years to indicate the current trends in the research of space-variant restoration. Among others, we introduce models describing the blur caused by camera motion by three-dimensional kernels analogous to those used in standard deconvolution algorithms and models used for complex scenes consisting of several independently moving objects.

The second part of this chapter (Sec. 1.3) details a new approach to space-variant super-resolution for images blurred by camera motion or any other blur changing slowly enough so that it can be locally modelled by convolution. It is one of the first attempts to apply true super-resolution to data with space-variant blur.

1.2 Blur models

The previous chapter addressed general problems of image restoration, including deconvolution that assumes space-invariant blurring, working with model

$$g = u * h + n \tag{1.1}$$

where g and u are the blurred and sharp (original) images, respectively, h is a convolution kernel and n a white Gaussian noise $N(0, \sigma^2)$.

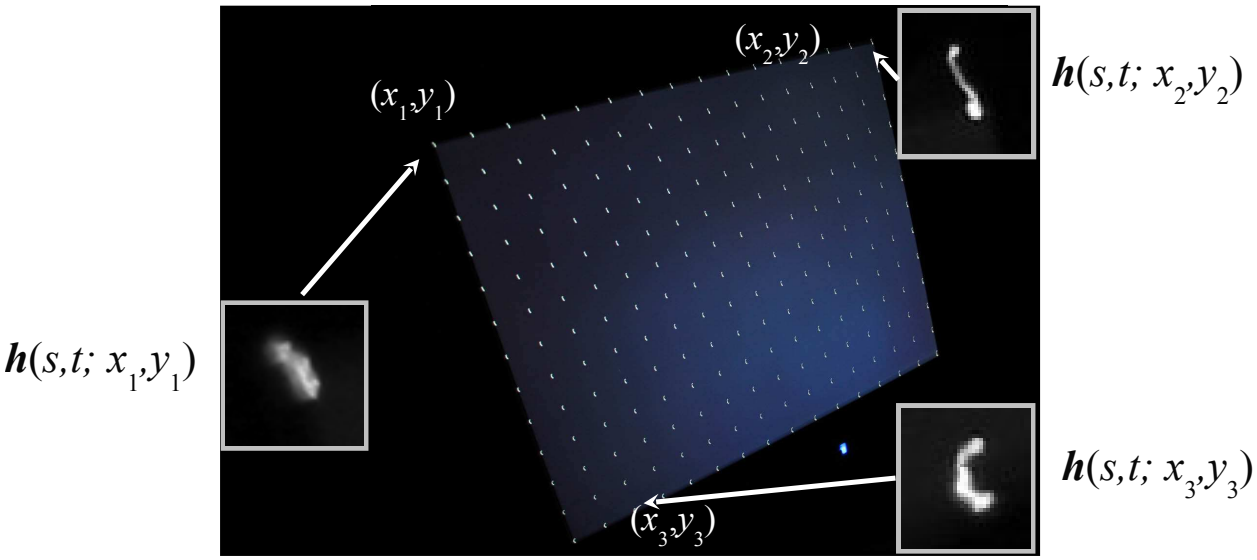


Figure 1.1: Spatially varying PSF for motion blur caused by camera shake. The image was acquired in a dark room by taking a picture of an LCD displaying regularly spaced white dots.

Spatially varying blur can be described by a general linear operator

$$g = Hu + n \tag{1.2}$$

The operator H can be written in a form naturally generalizing standard convolution as

$$[Hu](x,y) = \int u(x-s, y-t)h(s,t,x-s,y-t)dsdt \tag{1.3}$$

with the point-spread function (PSF) h now dependent on the position (third and fourth variable) in the image. Vice-versa, convolution with a kernel $\tilde{h}(s,t)$ is a special case of (1.3) with $h(s,t,x,y) = \tilde{h}(s,t)$ for an arbitrary position (x,y). Note that the convolution kernel h in (1.1) is sometimes also called a PSF. In this chapter, we reserve the expression PSF for h in (1.3), which is a function of four variables. It is consistent, however, to use the term kernel for the function $h(s,t,x_0,y_0)$ taken as a function of two variables with fixed x_0 and y_0, when the operator H can be locally approximated by convolution with $h(s,t,x_0,y_0)$ in a neighborhood of the point (x_0,y_0).

In practice, we work with a discrete representation, where the same notation can be used with the following differences: PSF h is defined on a discrete set of coordinates, the integral sign in (1.3) becomes a sum, operator H corresponds to a sparse matrix and u to a vector obtained by stacking columns of the image into one long vector. For convolution, H is a block-Toeplitz matrix with Toeplitz blocks and each column of H corresponds to the same kernel. In the space-variant case, as each column corresponds to a different position (x,y), it may contain a different kernel $h(s,t,x,y)$.

Figures 1.1 and 1.2 show two examples of space-variant PSFs. Both images were acquired by taking

Figure 1.2: Defocus PSF acquired by deliberately defocusing an LCD covered by regularly spaced white dots. Vignetting clips the polygonal shape of the PSF, especially on the periphery of the field of view.

a picture of an LCD displaying regularly spaced white dots. Fig. 1.1 was taken from hand with a long shutter time. The smears we can see correspond directly to locally valid convolution kernels, i.e. the PSF h on fixed positions. Three close-ups show the PSF in the top left, top right and bottom left corners of the LCD for fixed coordinates (x_1, y_1), (x_2, y_2) and (x_3, y_3), respectively. Fig. 1.2 is an out-of-focus image of the same LCD, which gives again the corresponding PSF. Notice the irregular shape of the PSF, caused by vignetting. While the darkening of corners due to vignetting is a well known problem of wide-angle lenses at wide apertures, it is less known that vignetting affects also the PSF, cutting off part of its circular (for full aperture) or polygonal shape giving significantly asymmetrical PSFs.

If we know the PSF, we are able to restore the image. As described in the previous chapter, the most common Bayesian approach achieves this by picking the most probable image u, which is equivalent to minimization

$$\arg\min_u \frac{1}{2\sigma^2} \|g - Hu\|^2 - \log p(u) \tag{1.4}$$

where $p(u)$ is an estimate of the prior probability of u. For super-resolution, which requires multiple input images g_i, the solution is of the form

$$\arg\min_x \frac{1}{2\sigma^2} \sum_i \|g_i - DH_i u\|^2 - \log p(x) \tag{1.5}$$

where D is an operator modeling resolution loss and H_i the operator of blurring corresponding to input g_i. Note that all efficient numerical algorithms minimizing (1.4) and (1.5) require the knowledge of the operator

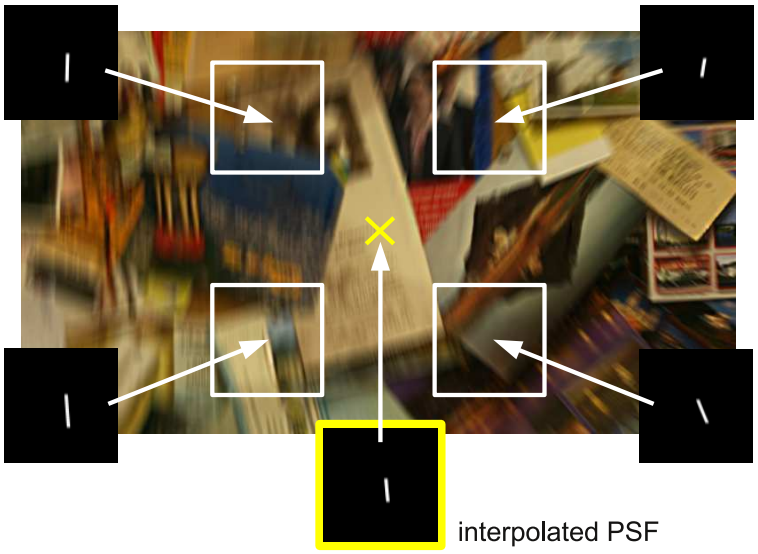


Figure 1.3: If the PSF varies slowly, we can estimate convolution kernels on a grid of positions and approximate the PSF in the rest of the image by interpolation of four adjacent kernels.

Figure 1.4: Original sharp image (left) and the same image blurred by simulation of camera rotation (right).

adjoint to H

$$[H^* u](x,y) = \int u(x-s, y-t) h(-s,-t,x,y) ds dt \tag{1.6}$$

For details, see [1] describing a more general form of (1.4) and (1.5) that includes the process of image registration, image distortion, etc.

In practice, we usually do not know the PSF and it must be estimated. For the space-invariant blur, we can use one of many elaborated blind deconvolution methods [2, 3, 4]. Estimation of the space-variant PSF in its general form (1.3) is too complex and ambiguous. As a rule, it cannot be expressed by an explicit formula but in many cases it has a special structure that can be exploited. For example, the blur caused by camera rotation is limited by three degrees of freedom of rigid body rotation. If we have an estimate of the camera rotation from inertial sensors, we are able to reconstruct the PSF and deblur the image using equation (1.4) as described by Joshi et al. in [5]. Nevertheless, the authors also report that

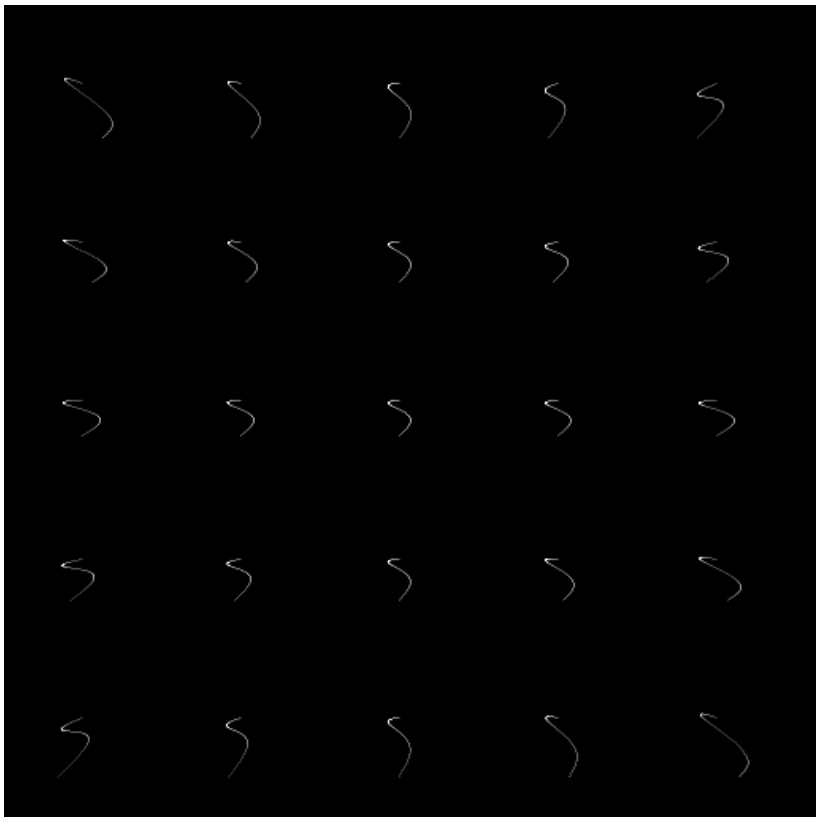


Figure 1.5: PSF $h(s, t, x_i, y_i)$ rendered at 25 positions (x_i, y_i).

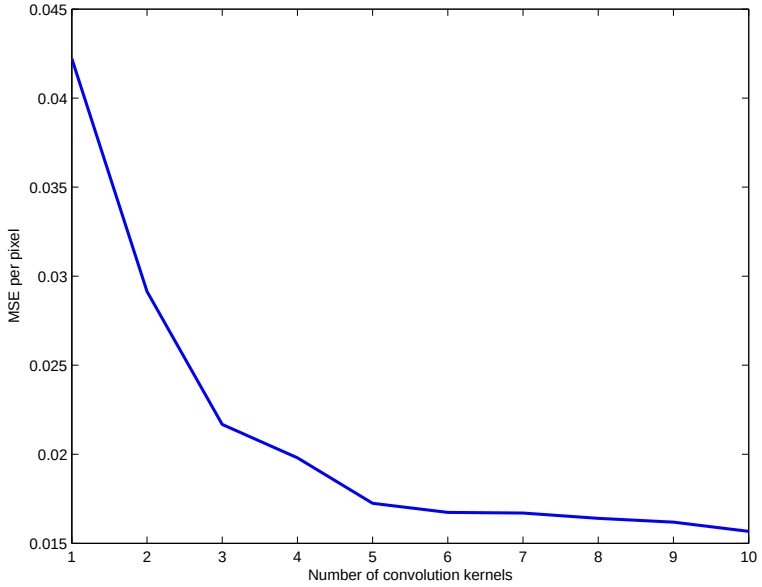


Figure 1.6: Graph of the MSE as a function of the number of kernels used for PSF approximation.

Figure 1.7: Deblurring of the right image from Fig. 1.4 using (from left to right) 1, 2 and 3 kernels per image width. Obviously the occurrence of artifacts decreases with the number of kernels.

motion tracking using inertial sensors is prone to significant errors accumulating over the time of capture and resulting PSF is not precise enough to get artifact-free images. It is not clear if these problems are due to principal limitations, camera hardware or the algorithm used for integration, though. Another possibility is to attach an auxiliary high-speed camera of lower resolution to estimate the PSF using for example optical flow techniques [6, 7, 8].

Unfortunately, in practice, we mostly do not have external sensors or devices providing information about the PSF and the PSF must be estimated directly from the input images. In the rest of this section, we describe the properties of the PSF for the most frequent types of blur and corresponding strategies to estimate the PSF.

There are several types of blur that can be assumed to change slowly with position, which allows to approximate the blurring locally by convolution. Under this assumption we can estimate locally valid convolution kernels that give us a local estimate of the PSF. This usually holds for motion blurs caused by camera shake and optical aberrations. For defocus it holds for approximately planar scenes.

The kernels are usually estimated at a set of regularly spaced positions (see Fig. 1.12), using one of already mentioned blind deconvolution methods, working either with a single image [2, 3], or more precise methods working with multiple images [4]. If it is possible to change camera settings, we can also fuse information from one blurred and one noisy/underexposed image [9, 10, 11].

Once the local convolution kernels are computed, they can be used to estimate the PSF for an arbitrary position (x, y). The simplest possibility is to divide the image to a set of regularly spaced patches, each with an assigned convolution kernel. The main problem of this approach are blocking artifacts appearing at patch boundaries.

One way to deal with these artifacts is to assign the estimated kernels just to patch centers (instead of the whole patch) and approximate the PSF h in intermediate positions by bilinear interpolation as indicated

in Fig. 1.3. Accuracy can be improved by first bringing the estimated kernels into normalized positions (which can be advantageously accomplished by means of regular or complex moments of the PSF's, see [12]), performing bilinear interpolation, and then returning back the kernels to their original position.

An advantage of this solution is that the PSF changes smoothly, thus avoiding the blocking artifacts. Moreover, the corresponding operator H can be computed in time comparable with the time of standard convolution using the fast Fourier transform (FFT) [13, 14]. The main reason are simple formulas for blur operators created as a linear combination of a finite set of convolution kernels. Indeed, if the PSF h is defined as a combination of kernels $\sum_i w_i h_i$, then

$$Hu = \sum_i (w_i u) * h_i \tag{1.7}$$

$$H^* u = \sum_i w_i (u * h_i^c) \tag{1.8}$$

where the symbol h_i^c denotes the convolution kernel h_i rotated by 180 degrees. For linear interpolation, the weight functions w_i satisfy the constraint $\sum_i w_i(x, y) = 1$ for an arbirtrary position (x, y), and $w_i(x, y) = 1$ in the center (x, y) of the window where the kernel h_i was estimated. In our paper [11], we show how to use this model to deblur an image, if another underexposed image taken with sufficiently short shutter time is available. The same model in a more versatile setup working with only one input image and using a recent blind deconvolution method [15] is shown in [16].

Hirsch et al. in [17] show that this model also allows to compute the operator adjoint to an operator U, acting on a vector h consisting of all vectorized kernels h_i and satisfying $Uh = Hu$. In theory, this could be useful in blind algorithms estimating simultaneously all kernels h_i. The practical value of this relation was not shown, though.

Here, we demonstrate a simple deblurring experiment that indicates the feasibility of the interpolation of kernels. The left image in Fig. 1.4 was blurred by simulating an irregular camera rotation about x and y axes, giving the right blurred image. Fig. 1.5 depicts the PSF $h(s, t, x, y)$ by rendering the function $h(s, t, x_i, y_i)$ as a function of the first two parameters at 25 positions (x_i, y_i).

Next, we deblurred the image in the standard Bayesian manner (1.4) with total variation regularization [18]. The PSF was approximated by linear interpolation of k convolution kernels, where the number k was varied from 1 to 10, using relations (1.7) and (1.8). For $k = 1$, it corresponds to standard deconvolution using the kernel valid in the image center. For $k = 5$, the algorithm works with the kernels shown in Fig. 1.5. Fig. 1.6 shows the mean square error of the result decreasing as k increases. Fig. 1.7 shows the resulting images for $k = 1 \ldots 3$. In Sec. 1.3 we will demonstrate that the same approach can be used for super-resolution as well.

This approach is relatively fast and can be used for most types of blur. Certain limitation is that the local estimation of the convolution kernels can fail because of weak texture, sensor saturation or for example misregistration. In the following section, we show the special properties of the blur caused by camera motion that can be used to estimate the PSF in a more robust way. However, this is achieved at the expense of higher time complexity.

1.2.1 Camera motion blur

The blur caused by camera motion is limited by six degrees of freedom of a rigid body motion, most commonly decomposed to three rotations and three translations. The main obstacle when dealing with this type of blur is that for translations, the blur depends on scene distance (depth). As a consequence, under general camera motion, we need to estimated the depth map, which makes the algorithm complicated and very time consuming. Nevertheless, there are algorithms that work satisfactorily, assuming certain additional constraints on the camera motion. For example [1] shows results of deblurring for the blur caused by camera translatation along an arbirtrary curve in one plane and for the out-of-focus blur. We refer the interested readers to [19, 1, 20] and references therein.

In the following paragraphs, we describe more practical approaches that use the fact that certain types of camera motion can be neglected in practice. Most common assumption is that all camera translations can be neglected, making the blur independent of scene depth. For example, for pictures taken from hand, Joshi et al. [5] tested exposures up to half a second and found that the typical translation in the direction of view (z-axis) was just a few millimeters in depth and had only very small effect for lenses of common focal length.

The assumption of negligible translations was used to compute the PSF for example in the above mentioned paper [6], getting the information about camera motion from an auxiliary high-speed camera and in [5] using inertial sensors for the same purpose. The same holds for papers [11] and [16] mentioned in the previous section.

Assuming only rotations, the simplest way is to transform the image so that the blur becomes a convolution and apply common deconvolution techniques. An immediate example is rotation about the optical axis (z-axis) that, expressed in polar coordinates, corresponds to one-dimensional translation. Therefore any blur caused by such a rotation can be described by one-dimensional convolution. Similar transforms can be written for rotations about an arbitrary fixed axis. In practice, however, camera motion is rarely limited to one axis. Moreover, the interpolation necessary to transform the image is an additional source of error introduced into the process.

A more versatile approach applied only recently is to express the operator of blurring in a specially

chosen set of basis images u_i

$$Hu = \sum_i u_i k_i \tag{1.9}$$

which allows to work with spatially varying blur in a manner similar to common convolution. The images u_i correspond to all possible transforms (rotations in our case) within a specified range of motions. Note however that unlike common convolution such operators do not commute. The functions k_i are referred to as kernels or motion density functions.

Whyte et al. [21] consider rotations about three axes up to several degrees and describe blurring by the corresponding three-dimensional kernel. For blind deconvolution, the algorithm uses a straightforward analogy of the well know blind deconvolution algorithm [2] based on marginalization over the latent sharp image. The only difference is the use of (1.9) instead of convolution. For deblurring, following the kernel estimation phase, it uses the corresponding modification of the Richardson-Lucy algorithm.

Gupta et al. [22] adopted a similar approach but instead of rotations about x and y axes consider translations in these directions. Because of the dependence of translation on depth, they require that the scene is approximately planar and perpendicular to the optical axis. Interestingly, in this case it is not necessary to know the real distance because the corresponding kernel works in pixel units. They first estimate locally valid convolution kernels by the original blind deconvolution algorithm [2] and compute the corresponding sharp image patches. In the second step, they estimate the kernels k_i from (1.9) using the knowledge of both the observed image and an estimate of the sharp image made up of the uniformly distributed patches from the previous step. They do not use all patches but choose iteratively a subset of patches and check consistence with the rest by a RANSAC-like algorithm. The image is regularized by standard smoothness priors applied separately on derivatives in x and y directions. The kernel is regularized to be sparse by a $\|.\|_p$ norm applied on kernel values and to be continuous using a quadratic penalty on kernel gradient.

An obvious advantage of the kernel model (1.9) is that it is very robust with respect to local non-existence of texture as well as local inaccuracies of the used model – sensor saturation for example. On the other hand, it may be considerably more time consuming than algorithms based on local approximations by convolutions described in the previous section and used also in the super-resolution algorithm we propose in Sec. 1.3. The bottleneck is the computation of all possible image transforms that must be repeated many times – basically in each step of the algorithm. Another disadvantage is that the actual motion may be more complicated and it is difficult to combine (1.9) with other models.

1.2.2 Scene motion blur

An especially difficult situation is that of the blur caused by object motion, as objects usually move independently of each other, often in different directions. In order to achieve good quality of deblurring, the object

must be precisely segmented, taking into account partial occlusion close to object outline. Moreover, object blur may appear simultaneously with the blur due to camera motion causing another, possibly spatially varying, blurring in the background.

Similarly to other types of blur, the algorithms differ according to the available input. Single-image approaches are attractive because of its ability to work for example with pictures already taken. The quality of restoration is limited, however, because of insufficient information and principle ambiguity. True super-resolution is impossible. Multi-image approaches, working usually with a video stream, can achieve a higher quality of restoration. On the other hand, the involved registration is an additional source of possible errors.

A frequent simplifying assumption is that the background is sharp, which holds when the camera is fixed (surveillance cameras) or mounted on a tripod stand. In addition, the blur of each object is often modeled by a standard convolution. An attempt in this direction was [23], using level-sets to segment the objects and ignoring the occlusion. Super-resolution was considered in [24], limited to known and negligible PSF.

The blur caused by one moving object on a sharp background, includding occlusion, can be described by a relatively simple formula [25, 26]

$$g = f * h + (1 - w * h)b \tag{1.10}$$

where f and b are the foreground and background images, h the convolution kernel of the foreground and w is the support of f corresponding to the area of the foreground object. The values of w are 0 for the background and 1 for the foreground.

Deblurring using even such a simple model is not easy. Raskar et al. [25] give a detailed analysis of possible cases, including the most ambiguous when the extent of blur is larger than the size of the moving object. Of course, for several moving objects or a blurred background, the situation becomes even more complicated.

In recent years, several papers appeared describing deblurring methods that work with only one image. Of course, the model is even more simplified by ignoring the oclussion effect. The objects are segmented based on various intensity, color or blur cues.

Most of the methods follow the pioneering paper of Levin [27]. She assumed that objects move with a constant velocity and are therefore blurred by a one-dimensional rectangular kernel. The distribution of derivatives then changes as a function of the width of the blur kernel. The algorithm first estimates the direction of motion as the direction with minimal variation of derivatives [28]. The image is then segmented based on the statistics of derivatives expected in an average natural image.

Chakrabarti et al. [29] extend the ideas from [27]. Again, all objects are assumed to be blurred by the same motion kernel being chosen from a discrete set of possible candidates corresponding to horizontal

or vertical box filters of certain length. To describe the likelihood of a given window to be blurred by a candidate kernel, they propose a more elaborate likelihood measure based on Gaussian scale mixtures. This likelihood is combined with statistical distributions for object and background colors described by a mixture of Gaussians into a MRF model with smoothness constraints.

Dai and Wu [30] came with an interesting constraint, analogous to optical flow constraint equation, that links the gradient of the blurred image with a difference of the sharp image in the direction of blur. The formula again holds for motion blurs that can be locally described as a convolution with a rectangular impulse. Derivation is straightforward, based on the fact that the derivative of a rectangular impulse is zero everywhere except the beginning and end of the impulse, where it is equal to plus and minus Dirac delta functions. The relation can be written as

$$(\nabla g) \cdot b = u(x + b/2) - u(x - b/2) \tag{1.11}$$

where b is the locally valid vector of motion blur – it has the same direction as the motion kernel and its size is equal to the length of the kernel. The dot in (1.11) denotes the dot product of vectors. The authors show that almost any optical flow algorithm that uses the optical flow constraint can be modified to work with (1.11) as well. They detail variants considering global motion models (affine and rotational) as well as a nonparametric model.

Liu et al. [31] locally detect and classify motion blur using a set of cues, including the local slope of power spectrum, gradient histogram, maximum saturation and variance of autocorrelation function in different directions. The image is classified to blurred and non-blurred regions based on the decision of a Bayes classifier, previously trained using a set of training images. Authors combine the results with a segmentation method based on graph cuts [32] and show examples of results superior to [27].

1.2.3 Defocus and aberrations

The last important type of blur is defocus and related optical aberrations. Readers are probably familiar with the basic properties of out-of-focus blur, in particular that the ideal PSF has a circular shape (often called pillbox in literature) and the inverse of its radius grows linearly with the distance from the plane of focus. In practice, the PSF is given by convolution of diaphragm shape with a diffraction pattern. Basic equations related to these PSFs are summarized in [1]. Note that here we do not consider more complicated optical systems such as confocal microscopes, where the PSF is three-dimensional [33].

Compared to the motion blur, there are much fewer papers addressing the restoration tasks for out-of-focus blur. There are several probable reasons. The first one is that the PSF depends on object distance. Consequently, to deblur the image, we need to estimate also the depth map, which makes the problem complex and time-consuming. The respective algorithms are described in [19, 1, 20] and references therein.

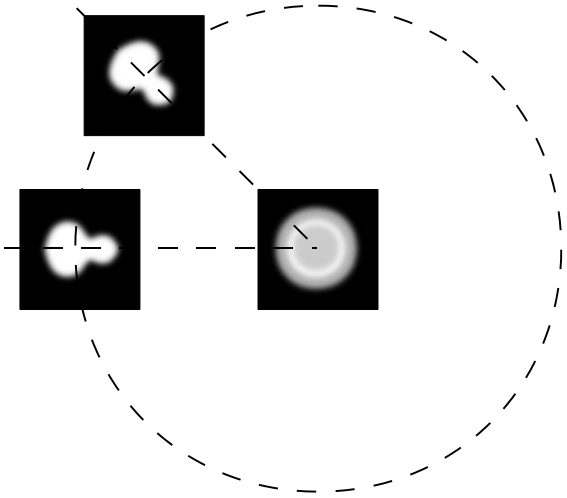


Figure 1.8: Three types of PSF symmetry for defocus and optical aberrations for a symmetrical optical system.

Second, the single-image blind deconvolution algorithms needed to estimate the PSF are less reliable than for the motion blur and using more than one image is less practical. Finally, the results of deblurring are not so impressive, because defocus destroys most information contained in high frequences.

Interestingly, there are special cases, where the problem of defocus is formally the same as the problem with moving objects mentioned in the previous section. Gu et al. [34] remove the blur caused by dirty lenses or thin occluders such as window shutters or grilles. Both cases can be modeled by the image formation model (1.10), originally formulated for an object moving on a sharp background. In this case, the dirt or an occluder corresponds to the foreground object f, this time blurred by a defocus PSF. For thin occluders in a known distance, they need two images with different apertures. For the dirty lenses, they estimate the occluder f by calibration using projected patters or by autocalibration from a long sequence of video images.

The PSF of optical aberrations is much more complicated than that of pure defocus and it is mostly impossible to describe it by an explicit formula. Quite often, all we can say about the PSF are three symmetries caused by the rotational symmetry of common optical systems – see Fig. 1.8. First, the PSF in the image center is radially symmetric. Second, the PSF is axially symmetrical about the axis connecting the image center and the position, where the PSF is measured. Finally, the PSF in the same distance from the image center differ only by orientation. In practice, the PSF is usually measured experimentally.

In spite of the complexity of optical aberrations, paradoxically, removing the optical aberrations is an easier problem than pure defocus, because their PSF can be measured for a particular lens in advance and does not depend so much on depth. Moreover, we usually need to remove aberrations only where the image is in focus and this distance is provided by many modern camera in the EXIF data. Indeed, there are several

commercial applications for this task.

In theory, the PSF can be measured by observing very small light sources placed regularly across the field of view. Oddly enough, our experience is that such measurement is relatively difficult to do in practice and it is simpler and more precise to compute the PSF from a picture of a known calibration pattern. The same observation is mentioned in [35]. There are several papers using a set of slanted edges to get a line-spread function that corresponds to a 1D projection of the 2D PSF. However, for the purpose of deblurring or super-resolution, it is usually preferable to directly compute the two-dimensional PSF [36, 37, 35], which is possible even with sub-pixel resolution. The basic principle is to look locally for a convolution kernel h minimizing

$$\|g - D(u * h)\|^2 \tag{1.12}$$

where g is a patch of the observed image, u the known sharp calibration pattern and D a down-sampling operator. Minimization of (1.12) is a solution of a system of linear equations in the least square sense. Further references can be found in [35].

From the recent literature, we would like to point out the paper of Kee et al. [38], showing that aberrations of high quality lenses (Canon $24 - 105mm\ f/4$, Canon $17 - 40mm\ f/4$) can be locally described as a convolution with an oriented Gaussian specified by a covariance matrix. An unpleasant property of aberrations is that they change with spatial location and both the focal length and aperture. Fortunately, [38] also shows that the covariance matrix changes smoothly and can be described by a low-order polynomial of four variables (x, y, focal length and aperture).

1.3 Space-variant super-resolution

In Sec. 1.2, we went through basic types of space-variant blur and models used to describe it, and indicated approaches how these models could be used for deblurring. To the best of our knowledge, there are basically no true super-resolution algorithms working with spatially-varying blur.

In the rest of this chapter, we propose such a space-variant super-resolution algorithm, based on the local detection of convolution kernels and approximating the PSF by linear interpolation (1.7) between the positions where the kernels were detected. The use of this approach for deblurring was demonstrated in Figs. 1.4-1.7.

The proposed algorithm works for an arbitrary blur that changes slowly enough that it can be locally approximated by convolution. It requires the input images (it needs at least five input images for the SR factor of 2) to be at least approximately registered.

1.3.1 Algorithm

We have K input frames g_k ($k = 1, \ldots, K$), which are noisy, blurred, and downsampled representations of some "ideal" image u. Let us divide each image g_k into overlapping patches denoted as g_k^p, where p is the patch index, $p = 1, \ldots, P$. Blurring H_k is space-variant, but we assume that the convolution (space-invariant) model holds in each patch. The formation model thus writes locally as

$$g_k^p = DH_k^p u^p + n_k \,. \tag{1.13}$$

Both the degradation D and noise n_k are space-invariant and therefore the index p is omitted. The blurring operator acting on the original patch u^p is space-invariant and thus H_k^p denotes convolution with some PSF h_k^p. Smaller patches achieve better approximation of the true space-variant case. However the patch size is limited from below by the PSF size.

Solving the above equation for each patch becomes a multichannel blind deconvolution and super-resolution problem. A flowchart of the proposed algorithm is summarized in Fig. 1.9. The algorithm consists of four steps:

1. **Splitting**. We split the input frames into patches. Restrictions on the patch size implied by the PSF size are discussed in Sec. 1.3.2.

2. **PSF estimation**. We estimate the convolution kernels (h_k^p) in each patch p and frame k. For this purpose we apply patch-wise the blind super-resolution algorithm [39] and use the estimated kernels. The reconstructed patches are ignored. This is described in Sec. 1.3.3.

3. **PSF refinement**. We refine the estimated kernels by replacing those with erroneous masks with interpolated ones. Estimating PSFs in each patch, especially if the overlapping scheme is considered, would be very time consuming. Therefore, we divide the frames in a non-overlapping fashion and use the interpolation method to generate PSFs in every location; see Fig. 1.3. The interpolation scheme is covered in Sec. 1.3.4.

4. **Space-variant deblurring and super-resolution**. The final step takes the estimated PSFs and computes the high-resolution sharp image using the Bayesian approach (1.5) as described in Sec. 1.3.5.

Estimating PSFs locally has several advantages. We can apply robust PSF estimation algorithm, such as [39], which works also in the presence of the decimation operator and which compensates for misregistration in the form of translation by shifting PSFs. If we assume that locally any misregistration manifests itself as translation, then by estimating shifted PSFs locally we can compensate for a more complex geometrical transformation than just translation.

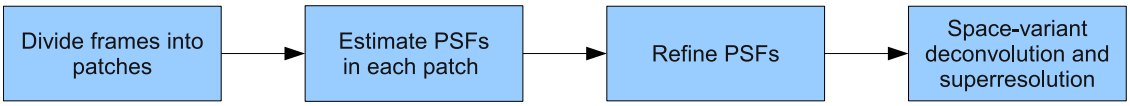


Figure 1.9: Flowchart of the algorithm.

The decimation operator simulates the behavior of digital sensors by performing convolution with a sensor PSF followed by downsampling with some step ε. We will also refer to the sampling step ε as a *SR factor*. It is important to underline that ε is a user-defined parameter. The sensor PSF is modeled as a Gaussian function, which is experimentally justified in [40]. A physical interpretation of the sensor blur is that the sensor is of finite size and it integrates impinging light over its surface. The sensitivity of the sensor is highest in the middle and decreases towards its borders with a Gaussian-like decay. We can insert the sensor PSF inside the blurring H_k and regard D solely as a downsampling operator.

1.3.2 Splitting

From the theoretical point of view, small patches better approximate local space-invariant nature of degradation H_k. However, small patches may lack information necessary to estimate PSFs (we have more unknowns than equations). It is therefore important to derive limitations on the patch size. Intuitively we see that the minimum patch size must depend on the PSF size and the number of input frames K. However, to derive exact constraints we need to analyze the subspace method proposed in blind super-resolution [39] for PSF estimation in the presence of decimation D.

First we rewrite the local observation model (1.13) in terms of convolution operators and use a vector-matrix notation. To simplify the notation, we will assume all image and PSF supports square, and the downsampling factor ε integer ($\varepsilon = 2, 3, \ldots$) and same in both directions. An extension to rectangular supports is straightforward. Rational SR factors ε can be considered as well, if polyphase decomposition is used; see [41]. We will omit the patch index p and the reader should keep in mind that in the following discussion we refer to patches and not to whole images. The size of all patches in all frames g_k are assumed to be equal and denoted as G. The size of the corresponding original patch u is U. The maximum PSF size of h_k is denoted as H. Next, we define a convolution matrix. Let u be an arbitrary discrete image of size U, then $\mathbf{u}$ denotes an image column vector of size U^2 and $\mathbf{C}_A\{u\}$ denotes a matrix that performs convolution of u with an image of size A. The convolution matrix can have a different output size. Adopting the Matlab naming convention, we distinguish two cases: "full" convolution $\mathbf{C}_A\{u\}$ of size $(U + A - 1)^2 \times A^2$ and "valid" convolution $\mathbf{C}^v_A\{u\}$ of size $(U - A + 1)^2 \times A^2$. In both cases the convolution matrix is a Toeplitz-block-Toeplitz matrix. Let $\mathcal{G} := [\mathbf{G}_1, \ldots, \mathbf{G}_K]$, where $\mathbf{G}_k = \mathbf{C}^v_A\{g_k\}$ is the "valid" convolution matrix of

g_k acting on an $A \times A$ support. Assuming no noise, we can express $\mathcal{G}$ using the observation model (1.13) as

$$\mathcal{G} = \mathbf{DU}\mathcal{H}\,, \tag{1.14}$$

where $\mathbf{D}$ is the downsampling matrix with the sampling factor ε and the appropriate size to match the term on the right, $\mathbf{U} := \mathbf{C}^{v}_{\varepsilon A+H-1}\{u\}$ and

$$\mathcal{H} := [\mathbf{C}_{\varepsilon A}\{h_1\}\mathbf{D}^T, \ldots, \mathbf{C}_{\varepsilon A}\{h_K\}\mathbf{D}^T]\,, \tag{1.15}$$

Note that the transposed matrix $\mathbf{D}^T$ behaves as an upsampling operator that interlaces the original samples with $(\varepsilon - 1)$ zeros. Estimating the PSFs h_k from the observed patches g_k as proposed in blind super-resolution relies on the following observation. If $\mathbf{DU}$ is of full column rank then the null space of $\mathcal{G}$ is equal to the null space of $\mathcal{H}$ and since we can calculate the null space of $\mathcal{G}$ we get information about PSFs h_k. If the original patch is not degenerated, e.g. constant image, and contains some details then $\mathbf{DU}$ is in general of full column rank if it has at least as many rows as columns. The difference between the number of columns and rows of $\mathcal{H}$, which is $KA^2 - (\varepsilon A + H - 1)^2$, bounds from below the null-space dimension of $\mathcal{H}$ and thus the null-space size of $\mathcal{G}$. From the bound follows that $A \geq \frac{H-1}{\sqrt{K}-\varepsilon}$, where A is the windows size, for which we construct the convolution matrices $\mathbf{G}_k$. The necessary condition for keeping the null space of $\mathcal{G}$ equal to the null space of $\mathcal{H}$ is to have at least as many rows as columns in $\mathcal{G}$. Thus from the size of $\mathcal{G}$ follows that $(G - A + 1)^2 \geq A^2 K$ and after substituting for A we obtain

$$G \geq \left\lceil \frac{\sqrt{K}(H-2) + H + \varepsilon - 1}{\sqrt{K} - \varepsilon} \right\rceil \tag{1.16}$$

As one would expect, the necessary minimum patch size G decreases with the decreasing PSF size H and/or increasing number of input frames K. For a better interpretation of the size constraint, Fig. 1.10 shows G as a function of H and K. For example in the case of $\varepsilon = 2$ and blurs of maximum size 10×10, the minimum patch size is over 100×100 if only 5 input frames are used but much smaller patch sizes roughly 30×30 are sufficient if 10 input frames are used.

1.3.3 PSF estimation

As in the previous section, the following discussion applies to individual patches and the index p is omitted. Since we can estimated the null space of $\mathcal{H}$ from the null space of $\mathcal{G}$ in (1.14), we have means to determine the original PSFs directly from the observed input patches. In the blind super-resolution algorithm [39] it was shown that the PSFs must satisfy

$$\mathcal{N}\mathbf{h} = \mathbf{0}\,, \tag{1.17}$$

where the matrix $\mathcal{N}$ contains convolution matrices with images lying in the null space of $\mathcal{G}$ and $\mathbf{h}$ are K PSFs stacked in one vector. PSFs are not uniquely determined by the above equation, since $\mathcal{N}$ is rank deficient

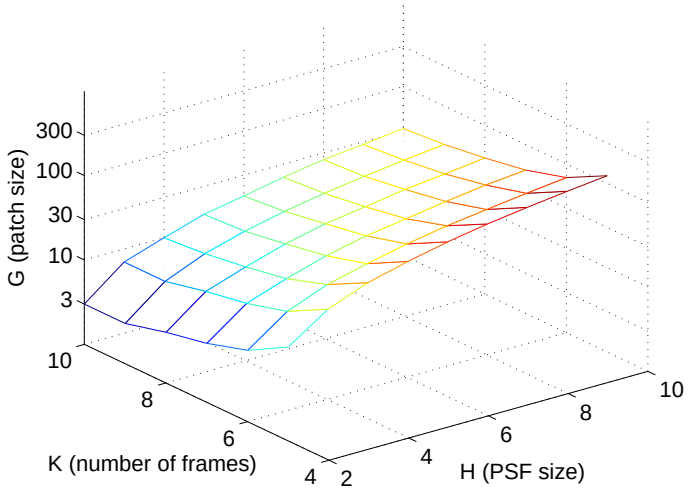


Figure 1.10: The minimum patch size as a function of number of input frames and PSF size. The SR factor is 2. Note that the patch size axis is in the logarithmic scale.

by at least ε^2 due to the presence of the downsampling matrix $\mathbf{D}$ in (1.14). In addition, the matrix rank further decreases, if the PSF support is larger than the maximum support of the true PSFs. In practice we rarely know the correct support size exactly and therefore we work with overestimated sizes. Last but not least, the null space approach neglects noise in its derivation. These facts prevent to use (1.17) for PSF estimation directly. Instead we use the MAP approach to estimate both the latent patch u and the PSFs h_k, and combine two observation models. First observation model is in a standard form $M_1(u, \{h_k\}) \equiv \frac{\mu_1}{2} \sum_{k=1}^{K} \|D(u * h_k) - g_k\|^2$, where μ_1 is inversely proportional to the variance of noise n_k and $\|\cdot\|$ denotes the ℓ_2 norm. For simplicity, we assume the same noise variance in all frames and patches and therefore single parameter μ_1 suffices. The second observation model is defined by (1.17) as $M_2(h_1, \ldots, h_K) \equiv \frac{\mu_2}{2} \|\mathcal{N}\mathbf{h}\|^2$, where μ_2 is inversely proportional to the variance of noise. Then the MAP approach is equivalent to solving the optimization problem:

$$\min_{u,\{h_k\}} M_1(u, \{h_k\}) + M_2(\{h_k\}) + R_u(u) + R_h(\{h_k\}) \,, \tag{1.18}$$

where R_u, R_h are image and PSF regularizers.

A popular recent approach to image regularization is to assume that the unknown image u is represented as a linear combination of few elements of some frame (usually an overcomplete dictionary) and force this sparse representation by using the ℓ_1 norm (or ℓ_0). Arguably, the best known and most commonly used image regularizer, which belongs to the category of sparse priors, is the total variation (TV) norm [18]. The

isotropic TV model is the ℓ_1 norm of image gradient magnitudes and takes the form

$$R_u(u) = \phi(\nabla u) = \sum_i \sqrt{(\nabla_x u(i))^2 + (\nabla_y u(i))^2}\,, \tag{1.19}$$

where $\phi(x) = \|x\|$. The TV regularizer thus forces the solution to have sparse image gradient. Depending on the type of data, one can have sparsity in different domains. This modification is however easy to achieve. All we have to do is to replace derivatives with a transformation (e.g. wavelet-like multi-scale transform), which gives sparse representation of our data.

For the PSF regularization R_h, we found sufficient to enforce in this term only positivity. One can add TV regularization as in the case of images (or sparsity of intensity values) if the expected blurs are for example motion blurs.

The standard approach to solve the optimization problem (1.18) is called *alternating minimization* (AM) and will be adopted here as well. We split the problem into two subproblems:

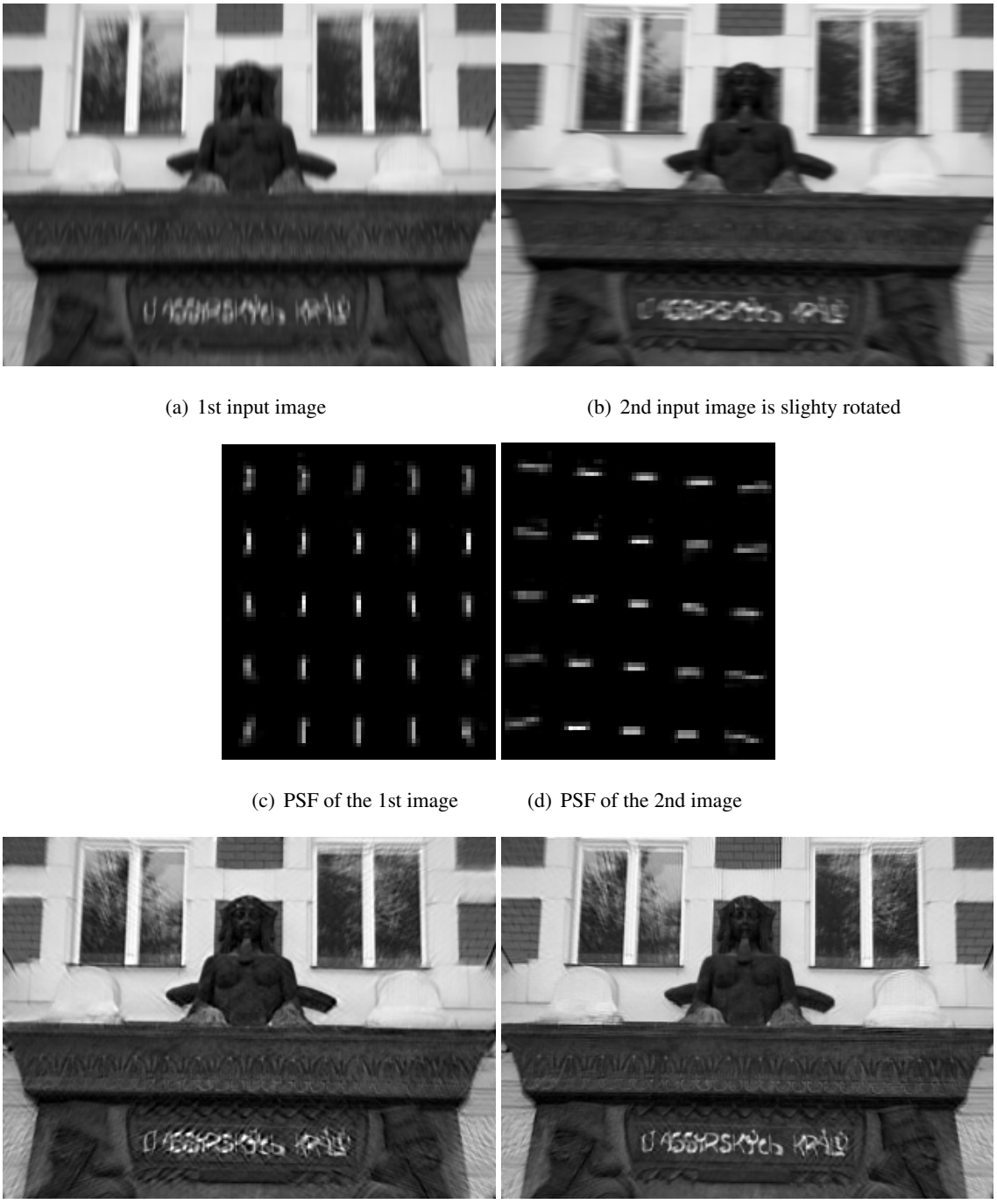
$$\text{“}u\text{-step”:} \quad \min_u M_1(u, \{h_k\}) + R_u(u) \tag{1.20}$$

and

$$\text{“}h\text{-step”:} \quad \min_{\{h_k\}} M_1(u, \{h_k\}) + M_2(\{h_k\}) + R_h(\{h_k\})\,, \tag{1.21}$$

and alternate between them. Convergence to the global minimum is theoretically not guaranteed since the unknown variables are coupled in the data term M_1. However, in our formulation all the terms are convex and thus each subproblem separately converges to its global minimum and it can be solved efficiently, e.g., by the augmented Lagrangian method [42].

We have to underline that the primary output of this step are estimated PSFs in each patch and frame. We are not interested in the reconstructed patches u, which are in this stage merely byproducts of AM. We observe that AM can be greatly accelerated (reducing the number of alternations necessary for convergence) if the weight μ_1 is set smaller than appropriate for the current noise level. This means that we give more weight to the TV regularization term and small details in the estimated patch u are wiped out. Only main features (strong edges) inside the estimated patch u are reconstructed. The advantage is that the h-step becomes more stable with such u and PSFs can be estimated with the same accuracy. This idea is similar to recent improvements proposed in the single-channel blind deconvolution [3]. In order for the single-channel case to work, we must perform few tricks. One trick is to slightly enhance edges of the estimated image after each iteration, which avoids the trivial solution, i.e., the estimated image being equal to the input blurry image and PSF being a delta function. Another trick is to remove edges that correspond to small objects, which would otherwise favor smaller PSFs than the true ones. Using TV regularization with larger weight than necessary simulates in a way the above tricks and boosts the PSF estimation step.



(a) 1st input image (b) 2nd input image is slighty rotated

(c) PSF of the 1st image (d) PSF of the 2nd image

(e) result of deconvolution (f) result of space-variant deblurring

Figure 1.11: Experiment illustrating an important property of the proposed approach – ability to compensate for registration inaccuracies. Notice how the estimated PSF of the 2nd image compensates for rotation.

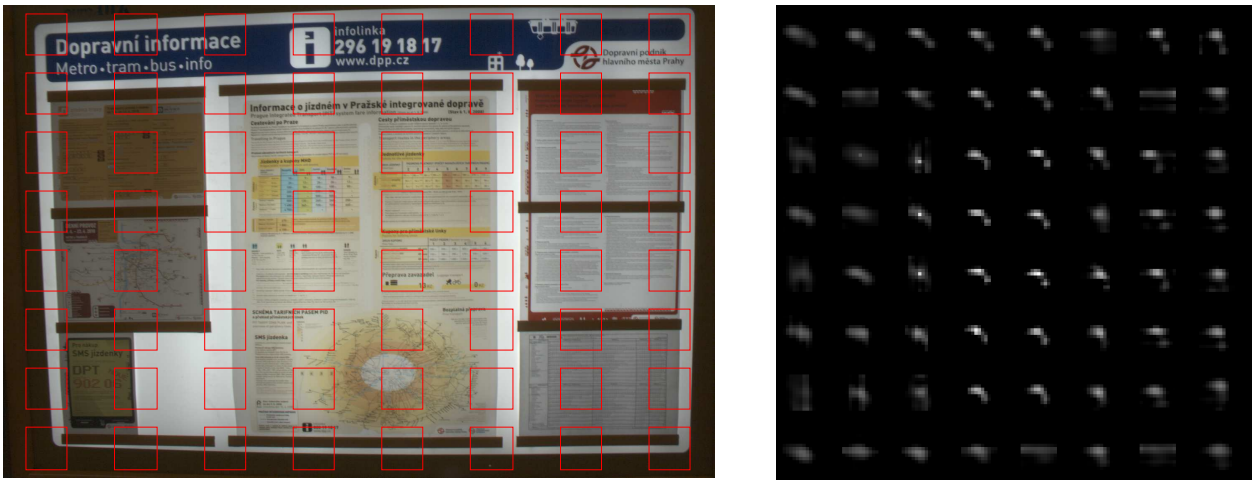


Figure 1.12: First of six input images (1700×1130 pixels) used for super-resolution (left) and 8×8 local convolution kernels computed by the SR algorithm [39] (right). Spatial variance of the PSF is obvious. Squares in the image depict patches, in which the kernels were estimated.

1.3.4 PSF refinement

Ideally, we want to have estimated PSFs in every position (pixel) of input frames. This is computationally very demanding as the number of patches is equal to the number of pixels and patches heavily overlap. However, we assume that PSFs vary slowly so that locally (inside each patch) they can be considered as constant. Calculating thus PSFs in every position is an unnecessary labor. We estimate PSFs on a coarser mesh and find the remaining ones by the interpolation method described in 1.2. Another reason for PSF interpolation is if the PSF estimation fails in some patches. Then we must use adjacent PSFs to interpolate and replace the erroneous one. There are basically two reasons why kernel estimation fails [11]. The first reason are patches with no details, which corresponds to the situation that **DU** in (1.14) is column rank deficient. To identify such cases, we compute the entropy of the estimated kernels and take those with the entropy above some threshold. The other case of failure is pixel saturation caused by light levels above the sensor range. This situation can be identified by computing the kernel energy, i.e. the sum of kernel values. For valid kernels the energy should be one. Therefore, we simply remove kernels whose sum is too different from unity, again above some threshold. These two thresholds are user parameters.

1.3.5 Deconvolution and super-resolution

Once we have the convolution kernel estimated (or interpolated from neighboring patches) in each patch of every input frame, we can use the kernels to approximate the PSF in an arbitrary position using linear interpolation. The respective blurring operators can be easily implemented using relations (1.7) and (1.8). As mentioned in Sec. 1.2, this approach was used for deconvolution in papers [11] and [16].

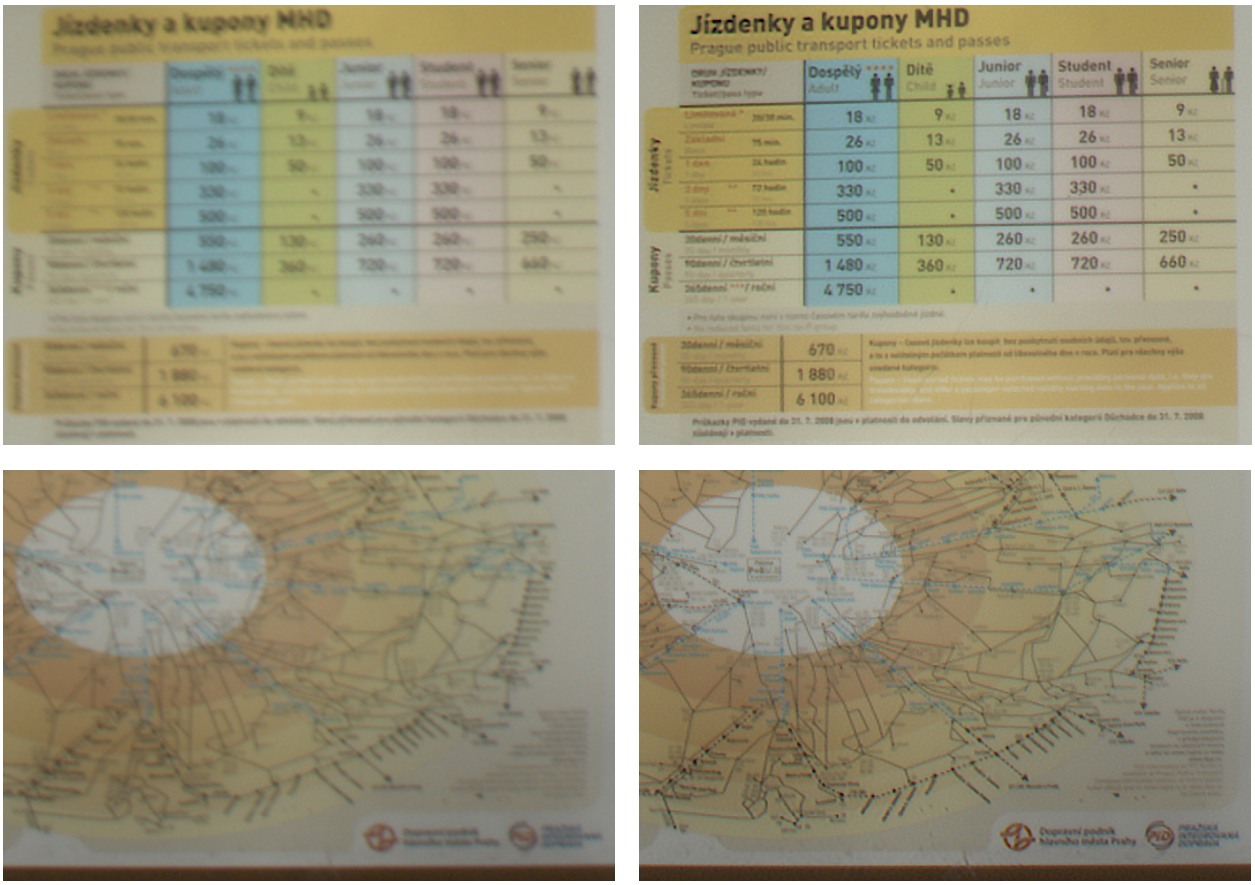


Figure 1.13: Two details of the blurred low-resolution image from Fig. 1.12 (left column) and the resulting high-resolution image of 3400×2260 pixels (right column).

We propose to extend this approach to super-resolution. The idea is that we can perform space-variant deblurring and super-resolution in one step and directly obtain the high-resolution image using (1.5). Indeed, the experiments show that the PSF approximated in this manner give satisfactory results, locally comparable with the result of the space-invariant super-resolution methods. Moreover, since the PSF changes smoothly between the patches, there are almost no visible artifacts.

This final step is analogous to the u-step in (1.20) of the kernel estimation stage except that this time we work with the whole frames and not just patches. We solve

$$\min_u \frac{\mu_1}{2} \sum_{k=1}^{K} \|DH_k u - g_k\|^2 + \phi(\nabla u)\,, \tag{1.22}$$

where H_k is the space-variant convolution operator, which is created from estimated PSFs h_k^p, $p = 1, \dots P$. In theory, each column of H_k contains a PSF corresponding to that position and estimated in the second step or interpolated in the third step. In our implementation, we used regularization by total variation (1.19) as in the PSF estimation step.

1.3.6 Experiments

This section describes two experiments. The first one in Fig. 1.11 illustrates an important ability of the proposed method to work with input images that are slightly misregistered. The second experiment (Figs. 1.12-1.13) is an example of real results.

Our super-resolution method requires multiple input images, which are properly registered. Super-resolution by itself is very sensitive to registration and sub-pixel accuracy is generally needed. However, registration methods rarely achieve such accuracy. It was shown in [39] that blind deconvolution and super-resolution can automatically register shifted images (misregistration in the form of translation) by shifting centers of estimated convolution kernels.

In practice, we often have more serious misregistration, such as rotation or projective distortion. It turns out that our local approach can handle such more complex geometric transforms. The reason is that with rotations up to several degrees, any similarity transform can be locally approximated by translation. The same is true even with more complicated transforms we can meet in practice.

This important feature can be demonstrated by the following experiment. An image was degraded by two different space-variant blurs simulating camera rotation around the x and y axes, respectively. The second image was first rotated by 1 degree. The resulting degraded and misregistered images are in Fig. 1.11(a) and (b). We divided the images into 5×5 patches and applied the blind super-resolution algorithm [39]. The estimated convolution kernels in the first and second images are in Fig. 1.11(c) and Fig. 1.11(d), respectively. Notice that the estimated blurs in Fig. 1.11(d) are shifted to compensate for rotation of the second image. Fig. 1.11(f) shows the estimated sharp image. For comparison, Fig. 1.11(e) shows also the result of the space-invariant method applied to the whole image, which is equivalent to the case of one large patch. Artifacts are noticeable compared to the space-variant case. However, even for the space-variant deblurring, since the rotation violates the convolution model, some artifacts appear in areas farther away from the center of rotation.

In the second experiment, we took a sequence of six images. One of them is shown in Fig. 1.12, together with $8 \times 8 = 64$ kernels estimated in the first phase of the algorithm. We can see that the PSF is significantly space-variant and the image becomes aberrated along boundaries. The result of the space-variant super-resolution is shown in two close-ups in Fig. 1.13. We do not show the whole image, because its size (3400×2260) would be to large to tell the difference compared to the original low-resolution image in Fig. 1.12. The improvement of resolution is clearly visible and the image contains almost no artifacts.

1.4 Summary

In this chapter, we outlined current approaches to deblurring in the presence of spatially varying blur. Untill now, these algorithm have not been extended to super-resolution.

About a half of this chapter is devoted to a new super-resolution algorithm working for slowly varying blurs. Its main advantage is the possibility to combine several types of blur, such as the camera motion blur, defocus and aberrations and even compensate for small registration errors. In addition, it is much faster that approaches working with more restrictive models and it is not much slower than standard deconvolution methods. On the other hand, it does not cope well with the blur caused by object motion.

The most interesting direction of future research is probably the extension of deblurring methods designed to work with moving objects to super-resolution, including the rigorous treatment of occlusion effects along object boundaries.

Acknowledgement

This work was supported in part by the Grant Agency of the Czech Republic project P103/11/1552.

Bibliography

[1] M. Sorel, F. Sroubek, and J. Flusser, "Towards super-resolution in the presence of spatially varying blur," in *Super-resolution imaging* (P. Milanfar, ed.), pp. 187–218, CRC Press, 2010.

[2] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. Freeman, "Removing camera shake from a single photograph," *ACM Transactions on Graphics, SIGGRAPH 2006 Conference Proceedings, Boston, MA*, vol. 25, pp. 787–794, 2006.

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

[4] F. Sroubek and J. Flusser, "Multichannel blind deconvolution of spatially misaligned images," *IEEE Trans. Image Process.*, vol. 14, pp. 874–883, July 2005.

[5] N. Joshi, S. B. Kang, C. L. Zitnick, and R. Szeliski, "Image deblurring using inertial measurement sensors," *ACM Trans. Graph.*, vol. 29, pp. 30:1–30:9, July 2010.

[6] M. Ben-Ezra and S. Nayar, "Motion deblurring using hybrid imaging," in *Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on*, vol. 1, pp. I–657 – I–664 vol.1, June 2003.

[7] M. Ben-Ezra and S. K. Nayar, "Motion-based motion deblurring," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, pp. 689–698, June 2004.

[8] Y.-W. Tai, H. Du, M. S. Brown, and S. Lin, "Image/video deblurring using a hybrid camera," in *CVPR*, 2008.

[9] M. Tico, M. Trimeche, and M. Vehvilainen, "Motion blur identification based on differently exposed images," in *Proc. IEEE Int. Conf. Image Processing*, pp. 2021–2024, 2006.

[10] L. Yuan, J. Sun, L. Quan, and H.-Y. Shum, "Image deblurring with blurred/noisy image pairs," in *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, (New York, NY, USA), p. 1, ACM, 2007.

[11] M. Sorel and F. Sroubek, "Space-variant deblurring using one blurred and one underexposed image," in *Proceedings of the 16th IEEE international conference on Image processing*, ICIP'09, (Piscataway, NJ, USA), pp. 157–160, IEEE Press, 2009.

[12] J. Flusser, T. Suk, and B. Zitová, *Moments and Moment Invariants in Pattern Recognition*. J. Wiley, 2009.

[13] T. G. Stockham, Jr., "High-speed convolution and correlation," in *Proceedings of the April 26-28, 1966, Spring joint computer conference*, AFIPS '66 (Spring), (New York, NY, USA), pp. 229–233, ACM, 1966.

[14] 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.

[15] S. Cho and S. Lee, "Fast motion deblurring," *ACM Transactions on Graphics (SIGGRAPH ASIA 2009)*, vol. 28, no. 5, p. article no. 145, 2009.

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

[17] M. Hirsch, S. Sra, B. Scholkopf, and S. Harmeling, "Efficient filter flow for space-variant multiframe blind deconvolution," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 607 –614, June 2010.

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

[19] M. Sorel and J. Flusser, "Space-variant restoration of images degraded by camera motion blur," *IEEE Trans. Image Process.*, vol. 17, pp. 105–116, Feb. 2008.

[20] 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.), pp. 257–269, 2004.

[21] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, "Non-uniform deblurring for shaken images," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 491 –498, June 2010.

[22] A. Gupta, N. Joshi, C. L. Zitnick, M. Cohen, and B. Curless, "Single image deblurring using motion density functions," in *Proceedings of the 11th European conference on Computer vision: Part I*, ECCV'10, (Berlin, Heidelberg), pp. 171–184, Springer-Verlag, 2010.

[23] L. Bar, N. A. Sochen, and N. Kiryati, "Restoration of images with piecewise space-variant blur," in *SSVM*, pp. 533–544, 2007.

[24] H. Shen, L. Zhang, B. Huang, and P. Li, "A MAP approach for joint motion estimation, segmentation, and super resolution," *IEEE Trans. Image Process.*, vol. 16, pp. 479–490, Feb. 2007.

[25] R. Raskar, A. Agrawal, and J. Tumblin, "Coded exposure photography: motion deblurring using fluttered shutter," in *ACM SIGGRAPH 2006 Papers*, SIGGRAPH '06, (New York, NY, USA), pp. 795–804, ACM, 2006.

[26] A. Agrawal, Y. Xu, and R. Raskar, "Invertible motion blur in video," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 1–8, 2009.

[27] A. Levin, "Blind motion deblurring using image statistics," in *Advances in Neural Information Processing Systems (NIPS)*, pp. 841–848, 2006.

[28] Y. Yitzhaky, I. Mor, A. Lantzman, and N. S. Kopeika, "Direct method for restoration of motion-blurred images," *Journal of The Optical Society of America A-optics Image Science and Vision*, vol. 15, 1998.

[29] A. Chakrabarti, T. Zickler, and W. T. Freeman, "Analyzing spatially-varying blur," in *CVPR'10*, (San Francisco, CA, USA), pp. 2512–2519, June 2010.

[30] S. Dai and Y. Wu, "Motion from blur," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1 –8, June 2008.

[31] R. Liu, Z. Li, and J. Jia, "Image partial blur detection and classification," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pp. 1 –8, June 2008.

[32] V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, pp. 147 –159, Feb. 2004.

[33] M. J. Nasse and J. C. Woehl, "Realistic modeling of the illumination point spread function in confocal scanning optical microscopy," *J. Opt. Soc. Am. A*, vol. 27, pp. 295–302, Feb. 2010.

[34] J. Gu, R. Ramamoorthi, P. Belhumeur, and S. Nayar, "Removing image artifacts due to dirty camera lenses and thin occluders," *ACM Trans. Graph.*, vol. 28, pp. 144:1–144:10, Dec. 2009.

[35] R. Szeliski, *Computer Vision: Algorithms and Applications (Texts in Computer Science)*. Springer, 2010.

[36] T. L. Williams, *The optical transfer function of imaging systems*. Institute of Physics Publishing, London, 1999.

[37] N. Joshi, R. Szeliski, and D. J. Kriegman, "PSF estimation using sharp edge prediction," *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, pp. 1–8, 2008.

[38] E. Kee, S. Paris, S. Chen, and J. Wang, "Modeling and removing spatially-varying optical blur," in *Proceedings of IEEE International Conference on Computational Photography (ICCP)*, 2011.

[39] F. Sroubek, G. Cristobal, and J. Flusser, "A Unified Approach to Superresolution and Multichannel Blind Deconvolution," *IEEE Transactions on Image Processing*, vol. 16, pp. 2322–2332, Sept. 2007.

[40] D. Capel, *Image Mosaicing and Super-Resolution (Cphc/Bcs Distinguished Dissertations.)*. SpringerVerlag, 2004.

[41] F. Sroubek, J. Flusser, and G. Cristobal, "Super-resolution and blind deconvolution for rational factors with an application to color images," *The Computer Journal*, vol. 52, pp. 142–152, 2009.

[42] M. Afonso, J. Bioucas-Dias, and M. Figueiredo, "Fast image recovery using variable splitting and constrained optimization," *Image Processing, IEEE Transactions on*, vol. 19, pp. 2345–2356, Sept. 2010.