

*Michal Šorel, Filip Šroubek and Jan Flusser*

---

***Book title goes here***



# 1

---

## *Towards super-resolution in the presence of spatially varying blur*

### CONTENTS

1.1	Introduction .....	4
1.1.1	Representation of spatially varying PSF .....	5
1.1.2	General model of resolution loss .....	5
1.1.3	Bayesian view of solution .....	7
1.2	Defocus and optical aberrations .....	10
1.2.1	Geometrical optics .....	11
1.2.2	Approximation of PSF by 2D Gaussian function .....	13
1.2.3	General form of PSF for axially-symmetric optical systems .....	14
1.2.4	Diffraction .....	15
1.2.5	Summary .....	18
1.3	Camera Motion Blur .....	18
1.3.1	Rotation .....	19
1.3.2	No rotation .....	20
1.4	Scene motion .....	21
1.5	Algorithms .....	22
1.5.1	Super-resolution of a scene with local motion .....	22
1.5.2	Smoothly changing blur .....	25
1.5.3	Depth-dependent blur .....	27
1.6	Conclusion .....	31

The effective resolution of an imaging system is limited not only by the physical resolution of an image sensor but also by blur. If the blur is present, super-resolution makes little sense without removing the blur. Some super-resolution methods considering space-invariant blur are described in other chapters of this book. The presence of a *spatially varying blur* makes the problem much more challenging and for the present, there are almost no algorithms designed specifically for this case. We argue that the critical part of such algorithms is precise estimation of the varying blur, which depends to large extent on a specific application and type of blur.

In this chapter, we discuss possible sources of spatially varying blur, such as defocus, camera motion or object motion. In each case we review known approaches to blur estimation, illustrate their performance on

experiments with real data and indicate problems that must be solved to be applicable in super-resolution algorithms.

---

## 1.1 Introduction

At the very beginning, we should remark that in this chapter we consider only algorithms working with multiple acquisitions – situations where we fuse information from several images to get an image of better resolution. To our best knowledge, there are no true super-resolution algorithms working with unknown space-variant blur. A first step in this direction is the algorithm [34], detailed in Sec. 1.5.1. On the other hand, considerable amount of literature exists on deblurring of images degraded by space-variant blur. Our results [33, 32, 31] are described in Sec. 1.5, other relevant references [4, 22, 14, 8, 20] are commented in more detail at the beginning of Secs. 1.4 and 1.5.3.

We do not treat super-resolution methods working with one image that need a very strong prior knowledge – either in the form of shape priors describing whole objects or sets of possible local patches in the case of example based methods [11, 7, 13]. Nor we consider approaches requiring hardware adjustments such as special shutters (coded-aperture camera [15]), camera actuators (motion-invariant photography [16]) or sensors (Penrose pixels [5]). However, these approaches can be considered in the same framework presented in this chapter.

We first introduce a general model of image acquisition that includes sampling, which we need for modeling resolution loss. This model is used for deriving a Bayesian solution to the problem of super-resolution. Next, a substantial part of the chapter discusses possible sources of spatially varying blur, such as defocus, camera motion or object motion. Where possible, we included analytical expressions for the corresponding point-spread function (PSF). In each case we discuss possible approaches for blur estimation and illustrate their use in algorithms described in the second part of the chapter. Where the existing algorithms work only with deblurring, we indicate problems that must be solved to be applicable in true super-resolution.

All the above mentioned types of spatially varying blur can be described by a linear operator  $H$  acting on an image  $u$  in the form

$$[Hu](x, y) = \int u(x - s, y - t)h(s, t, x - s, y - t) dsdt, \quad (1.1)$$

where  $h$  is a PSF. We can look at this formula as a convolution with a

PSF that changes with its position in the image. The convolution is a special case thereof with the PSF independent of coordinates  $x$  and  $y$ , i. e.  $h(s, t, x, y) = h(s, t)$  for an arbitrary  $x$  and  $y$ .

In practice, we work with a discrete representation of images and the same notation can be used with the following differences. Operator  $H$  in (1.1) corresponds to a matrix and  $u$  to a vector obtained by stacking columns of the image into one long vector. In the case of convolution,  $H$  is a block-Toeplitz matrix with Toeplitz blocks and each column of  $H$  contains the same PSF. In the space-variant case, each column may contain a different PSF that corresponds to the given position.

### 1.1.1 Representation of spatially varying PSF

An obvious problem of spatially varying blur is that the PSF is now a function of four variables. Except trivial cases, it is hard to express it by an explicit formula. Even if the PSF is known, we must solve the problem of efficient representation.

If the PSF changes smoothly without discontinuities, we can store the PSF on a discrete set of positions and use interpolation to approximate the whole function  $h$  (see Fig. 1.7). If the PSF is not known, as is usually the case, the local PSF's must be estimated as in the method described in Sec. 1.5.

Another type of representation is necessary if we consider for example moving objects, where the blur changes sharply at object boundaries. Then we usually assume that the blur is approximately space-invariant inside objects, and the PSF can be represented by a set of convolution kernels for each object and a corresponding set of object contours.

Final case occurs when the PSF depends on the depth. If the relation cannot be expressed by an explicit formula, as in the case of ideal pillbox function for defocus, we must store a table of PSF's for every possible depth.

### 1.1.2 General model of resolution loss

Let us represent the scene by two functions: intensity values of an ideal image  $u(x, y)$  and a depth map  $d(x, y)$ . A full 3D representation is necessary only if occlusion is considered, which will not be our case.

Digital imaging devices have limited achievable resolution due to many theoretical and practical restrictions. In this section, we show a general model of image acquisition, which comprises commonly encountered degradations. Depending on the application, some of these degradations are known and some can be neglected.

First, light rays emanating from the scene come from different direc-

tions before they enter the lens as the camera orientation and position change, which can be modeled by a geometric transformation of the scene. Second, several external and internal phenomena degrade the perceived image. The external effects are, e.g., atmospheric turbulence and relative camera-scene motion. The internal effects include out-of-focus blur and all kinds of aberrations. As the light passes through the camera lens, warping due to lens distortions occurs. Finally, a camera digital sensor discretizes the image and produces a digitized noisy image  $g(x, y)$ . An acquisition model, which embraces all the above radiometric and geometric deformations, can be written as a composition of operators

$$g = DLHWu + n. \quad (1.2)$$

Operators  $W$  and  $L$  denote geometric deformation of the original scene and lens distortions, respectively. Blurring operator  $H$  describes the external and internal radiometric degradations.  $D$  is a decimation operator modeling the camera sensor and  $n$  stands for additive noise. Our goal is to solve an inverse problem, i.e., to estimate  $u$  from the observation  $g$ .

The decimation operator  $D$  consists of filtering followed by sampling. Filtering is a result of diffraction, shape of light sensitive elements and void spaces between them (fill factor), which cause the recorded signal to be band-limited. Sampling can be modeled by multiplication by a sum of delta functions placed on an evenly spaced grid. For principle reasons,  $D$  is not invertible but we will assume that its form is known.

Many restoration methods assume that the blurring operator  $H$  is known, which is only seldom true in practice. The first step towards more general cases is to assume that  $H$  is a traditional convolution with some unknown PSF. This model is true for some types of blurs (see e.g. [23]) and narrow-angle lenses. In this chapter, we go one step further and assume spatially varying blur, which is the most general case that encompasses all the radiometric degradations if occlusion is not considered. Without additional constraints, the space-variant model is too complex. Various scenarios that are space-variant and allow solution are discussed in Sec. 1.5.

If lens parameters are known, one can remove lens distortions  $L$  from the observed image  $g$  without affecting blurring  $H$ , since  $H$  precedes  $L$  in (1.2). There is a considerable amount of literature on estimation of distortion [36, 2]. In certain cases the distortion can be considered as a part of the estimated blurring operator as in the algorithm 1.5.2.

A more complicated situation materializes in the case of geometric deformation  $W$ . If a single acquisition is assumed, calculation of  $W$  is obsolete since we can only estimate  $Wu$  as a whole. In the case

of multiple acquisitions in (1.3), the image  $u$  is generally deformed by different geometric transforms  $W_k$ 's and one has to estimate each  $W_k$  by a proper image registration method [38]. By registering the images  $g_k$ 's, we assume that the order of operators  $H_k$  and  $W_k$  is interchanged. In this case the blurring operator is  $\tilde{H}_k = W_k^{-1}H_kW_k$  ( $H_kW_k = W_kW_k^{-1}H_kW_k = W_k\tilde{H}_k$ ). If  $H_k$  is a standard convolution with some PSF  $h_k$  and  $W_k$  denotes a linear geometric transform, then by placing  $W_k$  in front of  $H_k$ , the new blurring operator  $\tilde{H}_k$  remains a standard convolution but with  $h_k$  warped according to  $W_k$ . If  $W_k$  denotes a nonlinear geometric transform, then after interchanging the order,  $\tilde{H}_k$  becomes a space-variant convolution operator in general. It is important to note that the blurring operator is unknown and instead of  $H_k$  we are estimating  $\tilde{H}_k$ , which is an equivalent problem as long as the nature of both blurring operators remains the same. Thus to avoid extra symbols, we keep the symbol  $H_k$  for the blurring operator even if it would be more appropriate to write  $\tilde{H}_k$  from now on.

As mentioned in the introduction, we need multiple acquisitions to have enough information to improve resolution. Hence we write

$$g_k = DW_kH_ku + n_k = D_kH_ku + n_k, \quad (1.3)$$

where  $k = 1, \dots, K$ ,  $K$  is the number of input images, lens distortions  $L$  are not considered,  $D$  remains the same in all the acquisitions, and the order of operators  $H_k$  and  $W_k$  has been interchanged. We denote the combined operator of  $W_k$  and  $D$  as  $D_k = DW_k$  and assume it is known.

In practice, there may be local degradations that are still not included in the model. A good example is a local motion that violates an assumption of global image degradation. If this is the case, restoration methods often fail. In order to increase flexibility of the above model, we introduce a masking operator  $M$ , which allows us to select regions that are in accordance with the model. The operator  $M$  multiplies the image with an indicator function (mask), which has ones in the valid regions and zeros elsewhere. The final acquisition model is then

$$g_k^v = M_kD_kH_ku + n_k = G_ku + n_k, \quad (1.4)$$

where  $g_k^v$  denotes the  $k$ -th acquired image with invalid regions masked out. The whole chain of degradations will be denoted as  $G_k$ . More about masking is in Sec. 1.5.1.

### 1.1.3 Bayesian view of solution

There are a number of possible directions, from which we can approach the problem of super-resolution. One of the most frequent is the Bayesian

approach, which we adopt here as well. Other approaches can be considered as approximations to the Bayesian solution.

An important fact is that if we know degradation operators  $G_k$ , the MAP (maximum a posteriori) solution under the assumption of Gaussian noise<sup>1</sup> corresponds to the minimum of a functional

$$E(u) = \sum_k \frac{1}{2\sigma_k^2} \|G_k u - g_k^v\|^2 + Q(u), \quad (1.5)$$

where the first term describes an error of our model and the second term  $Q(u)$  is a so called regularization term that corresponds to the negative logarithm of the prior probability of the image  $u$ . Noise variance in the  $k$ -th image is denoted as  $\sigma_k$ .

The prior probability is difficult to obtain and it is often approximated by statistics of the image gradient distribution. A good approximation for common images is for example total variation regularization [21]

$$Q(u) = \lambda \int_{\Omega} |\nabla u|, \quad (1.6)$$

which corresponds to an exponential decay of gradient magnitude. The total variation term can be replaced by an arbitrary suitable regularizer (Tikhonov, Mumford-Shah, etc.) [3, 29, 25]. The functional (1.5) can be extended to color images in quite a straightforward manner. The error term of the functional is summed over all three color channels ( $u_r$ ,  $u_g$ ,  $u_b$ ) as in [28]:

$$Q(u) = \lambda \int \sqrt{|\nabla u_r|^2 + |\nabla u_g|^2 + |\nabla u_b|^2}. \quad (1.7)$$

This approach has significant advantages as it suppresses noise effectively and prevents color artifacts at edges.

To minimize functional (1.5) we can use many existing algorithms, depending on a particular form of the regularization term. If it is quadratic (such as the classical Tikhonov regularization), we can use an arbitrary numerical method for solution of systems of linear equations. In the case of total variation, the problem is usually solved by transforming the problem to a sequence of linear subproblems. In our implementations, we use the half-quadratic iterative approach as described for example in [32].

The derivative of functional (1.5) with the total variation regularizer

---

<sup>1</sup>Poisson noise can be considered by prescaling the operators  $G_k$  in equation (1.5) according to values of corresponding pixels in  $g_k$ .

(1.7) can be written as

$$\frac{\partial E(u)}{\partial u} = \sum_k \frac{G_k^*(G_k u - g_k^v)}{\sigma_k^2} - \lambda \operatorname{div} \left( \frac{\nabla u}{|\nabla u|} \right). \quad (1.8)$$

$G_k^* = H_k^* D_k^* M_k^*$  is an operator adjoint to  $G_k$  and it is usually easy to construct. Adjoint masking  $M_k^*$  is equal to the original masking  $M_k$ . If  $D_k$  is downsampling, then  $D_k^*$  is upsampling. The operator adjoint to  $H_k$  defined in (1.1) can be written as

$$[H^* u](x, y) = \int u(x - s, y - t) h(-s, -t, x, y) ds dt. \quad (1.9)$$

We can imagine this correlation-like operator as putting the PSF to all image positions and computing dot product. The gradient of any regularization functional of form  $\int \kappa(|\nabla u|)$ , where  $\kappa$  is an increasing smooth function, can be found in [28].

If we know the operators  $G_k$ , the solutions are in principle known, though the implementation of the above formulas can be quite complicated. In practice however, the operators  $G_k$  are not known and must be estimated.

Especially in the case of spatially varying blur, it turns out to be indispensable to have at least two observations of the same scene, which gives us additional information that makes the problem more tractable. Moreover, to solve such a complicated ill-posed problem, we must exploit the internal structure of the operator, according to the particular problem we solve. Some parts of the composition of sub-operators in (1.2) are known, some can be neglected or removed separately – for example geometrical distortion. In certain cases we can remove the downsampling operator and solve only a deblurring problem, if we find out that we work at diffraction limit (read more about diffraction in 1.2.4). All the above cases are elaborated in the section on algorithms 1.5.

Without known PSF's it is in principle impossible to register precisely images blurred by motion. Consequently, it is important that image restoration does not necessarily require sub-pixel and even pixel precision of the registration. The registration error can be compensated in the algorithm by shifting the corresponding part of the space-variant PSF. Thus the PSF estimation provides robustness to misalignment. As a side effect, misalignment due to lens distortion does not harm the algorithm as well.

In general, if each operator  $G_k = G(\boldsymbol{\theta}_k)$  depends on a set of parameters  $\boldsymbol{\theta}_k = \{\theta_k^1, \dots, \theta_k^P\}$ , we can again solve the problem in the MAP framework and maximize the joint probability over  $u$  and  $\{\boldsymbol{\theta}_k\} = \{\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_K\}$ . As the image and degradation parameters can be usually

considered independent, the negative logarithm of probability gives a similar functional

$$E(u, \{\boldsymbol{\theta}_k\}) = \sum_{k=1}^K \frac{1}{2\sigma_k^2} \|G(\boldsymbol{\theta}_k)u - g_k^v\|^2 + Q(u) + R(\{\boldsymbol{\theta}_k\}), \quad (1.10)$$

where the additional term  $R(\{\boldsymbol{\theta}_k\})$  corresponds to a (negative logarithm of) prior probability of degradation parameters. The derivative of the error term in (1.10) with respect to the  $i$ -th parameter  $\theta_k^i$  of  $\boldsymbol{\theta}_k$ , equals

$$\frac{\partial E(u, \{\boldsymbol{\theta}_k\})}{\partial \theta_k^i} = \frac{1}{\sigma_k^2} \left\langle \frac{\partial G(\boldsymbol{\theta}_k)}{\partial \theta_k^i} u, G(\boldsymbol{\theta}_k)u - g_k^v \right\rangle + \frac{\partial R(\{\boldsymbol{\theta}_k\})}{\partial \theta_k^i}, \quad (1.11)$$

where  $\langle \cdot \rangle$  is the standard inner product in  $L_2$ . In discrete implementation,  $\frac{\partial G(\boldsymbol{\theta}_k)}{\partial \theta_k^i}$  is a matrix that is multiplied by the vector  $u$  before computing the dot product.

Each parameter vector  $\boldsymbol{\theta}_k$  can contain registration parameters for images, PSF's, depth maps, masks for masking operators, etc. according to the type of degradation we consider.

Unfortunately in practice, it is by no means easy to minimize the functional (1.10). We must solve the following issues:

1. How to express the  $G_k$  as a function of parameters  $\boldsymbol{\theta}_k$ , which may be sometimes complex – for example dependence of PSF on the depth of scene. We also need to be able to compute the corresponding derivatives.
2. Design an efficient algorithm to minimize non-convex functional we derive. In particular, the algorithm should not get trapped in a local minimum.

All this turns out especially difficult in the case of spatially varying blur, which is also the reason why there are so few papers considering super-resolution or just deblurring in this framework.

An alternative to MAP approach is to estimate the PSF in advance and then proceed with (non-blind) restoration by minimization over the possible images  $u$ . This can be regarded as an approximation to MAP. One such approach is demonstrated in Section 1.5.2.

To finalize this section, note that MAP approach may not give optimal results, especially if we do not have enough information and the prior probability becomes more important. This is a typical situation for blind deconvolution of one image. It was documented (blind deconvolution method [10] and analysis [15]) that in these cases marginalization approaches can give better results. On the other hand, we are interested in the cases of multiple available images, where the MAP approach seems to be appropriate.

---

## 1.2 Defocus and optical aberrations

This chapter describes degradations produced by optical lens systems and relation of the involved PSF to camera parameters and three-dimensional structure of an observed scene (depth).

We describe mainly the geometrical model of optical systems and corresponding PSF's, including the approximation by a Gaussian PSF. We mention also the case of general axially-symmetric optical system. Finally, we describe diffraction effects even though these can be considered space-invariant. The classical theory of Seidel aberrations [6] is not treated here as in practice the PSF is measured by an experiment and there is no need to express it in the form of the related decomposition. Also the geometrical distortion is omitted as it actually introduces no PSF and can be compensated by a geometrical transformation of images.

### 1.2.1 Geometrical optics

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

In real optical systems, there is also a roughly circular *aperture*, a hole formed by the blades that limit the pencils of rays propagating through the lens (rays emanate within solid angle subtended by the aperture). The aperture size is usually specified by *f-number*  $F = f/2\rho$ , where  $\rho$  is the radius of the aperture hole and  $f$  is a focal length. 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 the position of focal plane. 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<sup>2</sup>.

If the aperture is assumed to be circular, the graph of the PSF has a cylindrical shape usually called a *pillbox* in literature. When we describe

---

<sup>2</sup>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 [35]. Unfortunately most conventional lenses are not telecentric.

the appearance of the PSF in the image (or photograph), we speak about a *blur circle* or a *circle of confusion*.

It can be easily seen from the similarity of triangles that the blur circle radius for an arbitrary point at distance  $l$  is

$$r = \rho \zeta \left( \frac{1}{\zeta} + \frac{1}{l} - \frac{1}{f} \right) = \rho \zeta \left( \frac{1}{l} - \frac{1}{l_s} \right), \quad (1.12)$$

where  $\rho$  is the aperture radius,  $\zeta$  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  $\zeta$  using the relation  $1/f = 1/l_s + 1/l$ .

Notice the importance of inverse distances in these expressions. The expression (1.12) tells us that the radius  $r$  of the blur circle grows proportionally to the difference between inverse distances of the object and of the plane of focus. Other quantities,  $\rho$ ,  $\zeta$  and  $f$ , depend only on the camera settings and are constant for one image.

Thus, PSF can be written as

$$h(s, t, x, y) = \begin{cases} \frac{1}{\pi r^2(x, y)}, & \text{for } s^2 + t^2 \leq r^2(x, y), \\ 0, & \text{otherwise,} \end{cases} \quad (1.13)$$

where  $r(x, y)$  denotes the radius  $r$  of the blur circle corresponding to the distance of point  $(x, y)$  according to (1.12). Given camera parameters  $f$ ,  $\zeta$  and  $\rho$ , matrix  $r$  is only an 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 (1.12) and we get linear relation between the radii of blur circles in the first and the second image

$$r_2(x, y) = \frac{\rho_2 \zeta_2}{\rho_1 \zeta_1} r_1(x, y) + \rho_2 \zeta_2 \left( \frac{1}{\zeta_2} - \frac{1}{\zeta_1} + \frac{1}{f_1} - \frac{1}{f_2} \right) \quad (1.14)$$

Obviously, if we take both images with the same camera settings except for the aperture, i. e.  $f_1 = f_2$  and  $\zeta_1 = \zeta_2$ , we get the right term zero and the left equal to the ratio of f-numbers.

In reality the aperture is not a circle but a polygonal shape 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 PSF support is really circular. Still assuming geometrical optics, the aperture blur projects on the image plane with a scale changing the same way as for circular aperture, i. e. with a ratio

$$w = \frac{l' - \zeta}{l'} = \zeta \left( \frac{1}{l} - \frac{1}{l_s} \right) = \frac{1}{l} \zeta + \zeta \left( \frac{1}{\zeta} - \frac{1}{f} \right) \quad (1.15)$$

and consequently

$$h(s, t, x, y) = \frac{1}{w^2(x, y)} \hat{h}\left(\frac{s}{w(x, y)}, \frac{t}{w(x, y)}\right), \quad (1.16)$$

where  $\hat{h}(s, t)$  is the shape of the aperture. The PSF keeps the unit integral thanks to the normalization factor  $1/w^2$ . Comparing (1.15) with (1.12), one can readily see that the blur circle (1.13) is a special case of (1.16) for  $w(x, y) = r(x, y)/\rho$  and

$$\hat{h}(s, t) = \begin{cases} \frac{1}{\pi\rho^2}, & \text{for } s^2 + t^2 \leq \rho^2, \\ 0, & \text{otherwise.} \end{cases} \quad (1.17)$$

Combining (1.15) for two images yields, analogously to (1.14),

$$w_2(x, y) = \frac{\zeta_2}{\zeta_1} w_1(x, y) + \zeta_2 \left( \frac{1}{\zeta_2} - \frac{1}{\zeta_1} + \frac{1}{f_1} - \frac{1}{f_2} \right). \quad (1.18)$$

Notice that if the two images differ only in the aperture, then the scale factors are the same, i. e.  $w_2 = w_1$ . The ratio  $\rho_2/\rho_1$  from (1.14) is hidden in the different scale of the aperture hole.

### 1.2.2 Approximation of PSF by 2D Gaussian function

In practice, due to lens aberrations and diffraction effects, PSF will be a circular blob, with brightness falling off gradually rather than sharply. Therefore, most algorithms use two-dimensional Gaussian function instead of pure pillbox shape. To map the variance  $\sigma$  to real depth, [26] proposes to use relation  $\sigma = r/\sqrt{2}$  together with (1.12) with the exception of very small radii. Our experiments showed that it is often more precise to state the relation between  $\sigma$  and  $r$  more generally as  $\sigma = \kappa r$ , where  $\kappa$  is a constant found by camera calibration (for the lenses and settings we tested  $\kappa$  varied around 1.2). Then analogously to (1.14) and (1.18)

$$\sigma_2 = \alpha\sigma_1 + \kappa\beta, \quad \alpha, \beta \in \mathbf{R}. \quad (1.19)$$

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

Corresponding PSF can be written as

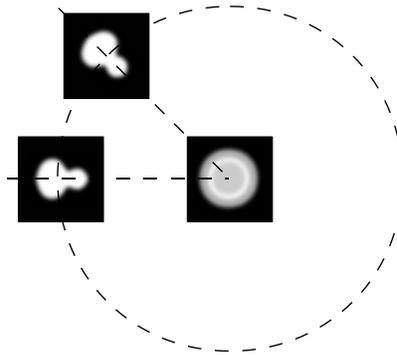
$$h(s, t, x, y) = \frac{1}{2\pi\kappa^2 r^2(x, y)} e^{-\frac{s^2+t^2}{2\kappa^2 r^2(x, y)}}. \quad (1.20)$$

If possible we can calibrate the whole (as a rule monotonous) relation between  $\sigma$  and distance (or its representation) and consequently between  $\sigma_1$  and  $\sigma_2$ .

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  $3\sigma$  or  $4\sigma$ . Moreover, for common optical systems, an arbitrary real out-of-focus PSF has a finite support anyway.

### 1.2.3 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 the reality well. For poorly corrected optical systems, rays can be aberrated from their ideal paths to such an extent that it results in very irregular PSF's. In general, aberrations depend on the distance of the scene from the camera, position in the image and on the camera settings  $f$ ,  $\zeta$  and  $\rho$ . As a rule, the lenses are well corrected in the image center, but towards the edges of the image PSF may become completely asymmetrical.



**FIGURE 1.1**

Three types of PSF symmetry in an optical system symmetrical about the optical axis.

Common lenses are usually axially-symmetric, i.e. they behave independently of its rotation about the optical axis. For such systems, it is easily seen (see Fig. 1.1) 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 (two left PSF's in Fig. 1.1),

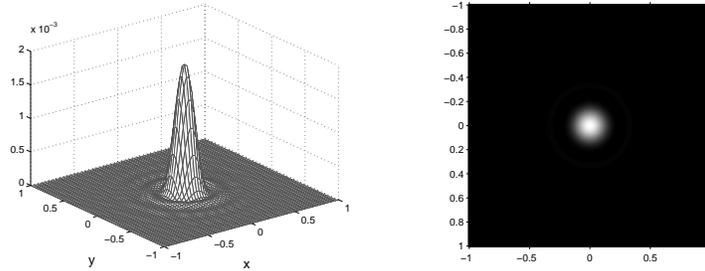
3. for points of the same distance from the image center and corresponding to objects of the same depth, PSF's have the same shape, but they are rotated about the angle given by angular difference of their position with respect to the image center (again can be seen at two left PSF's in Fig. 1.1).

The second and third property can be written as

$$h(s, t, x, y) = h\left(\frac{|(-t, s)(x, y)^T|}{|(x, y)|}, \frac{(s, t)(x, y)^T}{|(x, y)|}, 0, |(x, y)|\right). \quad (1.21)$$

In most cases, it is impossible to derive an explicit expression for the PSF. On the other hand, it is relatively easy to get it by a raytracing algorithm. The above mentioned properties of the axially-symmetric optical system can be used to save memory as we need not to store PSF's 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 PSF's every time they are used.

#### 1.2.4 Diffraction



**FIGURE 1.2**

Airy function: surface plot (left) and the corresponding grayscale image (right). The side lobes are very small and do not appear in the image plot. For this reason we often talk about Airy disk as only the central lobe is clearly visible.

Diffraction is a wave phenomenon which makes a beam of parallel light passing through an aperture to spread out instead of converging to one point. For a circular aperture it shapes the well known Airy disk (see Fig. 1.2). The smaller the aperture, the larger the size of the disk and the signal is more blurry. Due to the diffraction the signal becomes

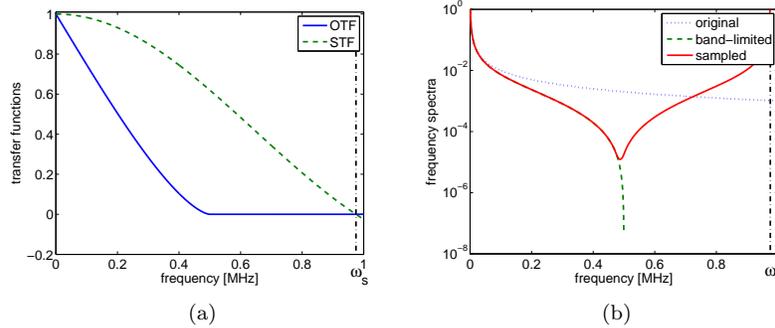
band-limited, which defines a theoretical maximum spatial resolution and hence implies limits on super-resolution as will be shown later.

On a sensor array the signal is sampled by photosensitive devices (CCD/CMOS). Driven by marketing requirements of more and more megapixels, present day cameras were brought very close to this diffraction limit. Especially it is true for compacts with their small sensors. It means that we cannot neglect this phenomenon and should incorporate the corresponding PSF to deblurring algorithms.

To study the frequency response of a diffraction-limited optical system, we use transfer functions, i. e. the Fourier transform of PSF's. If we assume an ideal circular aperture, neglect the defocus phenomena and other aberrations, the Optical Transfer Function (OTF) of the system due to diffraction is given [19] as

$$\text{OTF}(\omega) = \begin{cases} \frac{2}{\pi} \left( \cos^{-1} \left( \frac{\omega}{\omega_c} \right) - \frac{\omega}{\omega_c} \sqrt{1 - \left( \frac{\omega}{\omega_c} \right)^2} \right) & \text{for } \omega < \omega_c \\ 0 & \text{otherwise,} \end{cases} \quad (1.22)$$

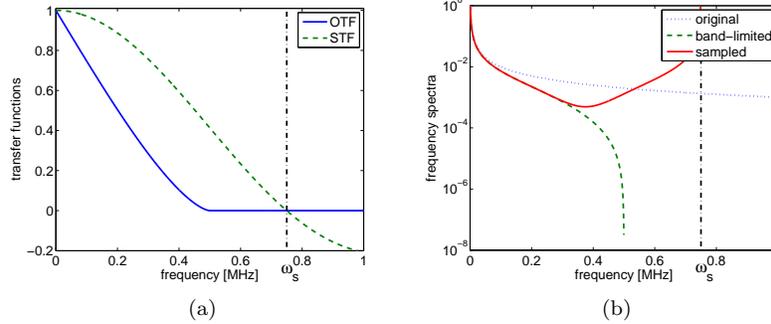
where  $\omega = \sqrt{\omega_x^2 + \omega_y^2}$  is the radial frequency in a 2D frequency space  $[\omega_x, \omega_y]$ , and  $\omega_c = 1/(F\lambda)$  is the cutoff frequency of the lens ( $\lambda$  is the wavelength of incoming light). For example for aperture  $F = 4$  and  $\lambda = 500\text{nm}$  (in the middle of visible light), the cutoff frequency is  $\omega_c = 0.5\text{MHz}$  and the corresponding OTF is plotted in Fig. 1.3(a) as a solid line.



**FIGURE 1.3**

Correctly sampled signal: (a) Optical transfer function and sensor transfer function; (b) Signal spectrum modified by diffraction and sensor sampling.

Assuming a square sensor without cross-talk, the Sensor Transfer Function (STF) is given by:

**FIGURE 1.4**

Under-sampled signal: (a) Optical transfer function and sensor transfer function; (b) Signal spectrum modified by diffraction and sensor sampling.

$$\text{STF}(\omega_x, \omega_y) = \text{sinc}\left(\frac{\pi w \omega_x}{\omega_s}\right) \text{sinc}\left(\frac{\pi w \omega_y}{\omega_s}\right), \quad (1.23)$$

where  $\text{sinc}(x) = \sin(x)/x$  for  $x \neq 0$  and  $\text{sinc}(0) = 1$ ,  $\omega_s$  is the sampling frequency, and  $w$  is the relative width of the square pixel ( $w \leq 1$ ). For the fill-factor of 100% ( $w = 1$ ) and if the signal is properly sampled ( $\omega_s = 2\omega_c$ ), the corresponding STF is plotted in Fig. 1.3(a) as a dashed line. As can be seen, the OTF is the main reason for a band-limited signal, since no information above its cutoff frequency passes through the optical system.

Fig. 1.3(b) summarizes the effects of diffraction and sensor sampling on signal spectra. If the frequency spectrum of an original signal is modeled as a decaying dotted line, the spectrum of the band-limited signal is the attenuated dashed line, and the spectrum of the sampled signal is the solid line. The maximum frequency representable by the sampled signal is  $\frac{1}{2}\omega_s$ , which in this case is close to the cutoff frequency  $\omega_c$  (proper sampling), and no aliasing is available, i.e. the solid line matches the dashed line. It is clear that if super-resolution is applied to such data, no high-frequency information can be extracted and super-resolution merely interpolates.

On the other hand, if the optical system is undersampling the signal, the corresponding OTF and STF looks as in Fig. 1.4(a). For the given aperture, wavelength and fill-factor, OTF is the same but STF shrinks. The sampled signal (solid line) has its high frequencies (around  $\frac{1}{2}\omega_s$ ) disrupted due to aliasing as Fig. 1.4(b) illustrates. In this case, super-

resolution can in principle unfold the signal spectra and recover the high-frequency information.

As mentioned above, the sampling of current consumer cameras approaches the diffraction limit which limits performance of any super-resolution algorithm. For example, a typical present day 10MP compact camera Canon PowerShot SX120 IS has its cut-off frequency about 2500 to 4000 per sensor width<sup>3</sup>, depending on the aperture, with maximum x-resolution 3600 pixels. Especially with higher f-numbers it is very close to the theoretical limit. On the other hand, highly sensitive cameras (often near and mid-infrared) still undersample the images which leaves enough room for substantial resolution improvements.

If the decimation operator  $D$  is not considered in the acquisition model (1.2), the diffraction effect can be neglected as the degradation by  $H$  is far more important. Since the deconvolution algorithm estimates  $H$ , OTF and STF can be considered as part of  $H$  and thus estimated automatically as well. In the case of super-resolution, inclusion of  $D$  is essential as the goal is to increase sampling frequency. The diffraction phenomenon is irreversible for frequencies above the cutoff frequency  $\omega_c$  and it is thus superfluous to try to increase image resolution beyond  $2\omega_c$ . (1.2). The diffraction phenomenon is irreversible and thus we will assume that the original image  $u$  is already bandlimited. The decimation operator  $D$  will model only STF and sampling.

### 1.2.5 Summary

In this section, 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. In practice, it is indispensable to take diffraction effects into account as many cameras are close to their diffraction limits.

---

<sup>3</sup>Aperture  $f/2.8 - 4.3$ , sensor size  $1/2.5''$  (5.5mm width),  $3600 \times 2700$  maximum resolution, the diffraction limit (cut-off frequency), given by  $\omega_c = 1/(F\lambda)$ , is about 2500/sensor width (for  $F = 4.3$ ) up to 4000/sensor width ( $F = 2.8$ ). Light wavelength  $\lambda$  is taken as  $500nm$ .

### 1.3 Camera Motion Blur

In this section we analyze various types of camera motion for the classical pinhole camera model. We treat the case of a general motion in all six degrees of freedom and detail the special cases of camera rotation and translation in a plane.

To model camera motion blur by a PSF  $h$  from (1.1), we need to express the PSF as a function of the camera motion and a depth of the scene. In the case of a general camera motion, it can be computed from the formula for *velocity field* [12, 8] that gives apparent velocity of the scene for the point  $(x, y)$  of the image at time instant  $\tau$  as

$$v(x, y, \tau) = \frac{1}{d(x, y, \tau)} \begin{bmatrix} -1 & 0 & x \\ 0 & -1 & y \end{bmatrix} T(\tau) + \begin{bmatrix} xy & -1 - x^2 & y \\ 1 + y^2 & -xy & -x \end{bmatrix} \Omega(\tau), \quad (1.24)$$

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

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

$$h(s, t, x, y) = \int \delta(s - \bar{x}(x, y, \tau), t - \bar{y}(x, y, \tau)) d\tau, \quad (1.25)$$

where  $\delta$  is the two-dimensional Dirac delta function.

Complexity of derivation of an analytical form of (1.25) depends on the form of velocity vectors  $\Omega(\tau)$  and  $T(\tau)$ . Though most algorithms do not work directly with analytical forms and use a discrete representation extending standard convolution masks.

#### 1.3.1 Rotation

Excessive complexity of a general camera movement can be overcome by imposing certain constraints. A good example is an approximation used

in almost all<sup>4</sup> optical image stabilizers that they consider only rotational motion in two axes. What concerns ordinary photographs, it turns out that in most situations (landscapes and cityscapes without close objects, some portraits), translation can be neglected.

If we look at formula (1.24) with no translation, i. e.  $T(\tau) = 0$ , we can see that the velocity field is independent of depth and changes slowly – realize that  $x$  and  $y$  are in focal length units which means the values are usually less than one (equals one for the border of an image taken with 35mm equivalent lens). As a consequence, also the PSF has no discontinuities, the blur can be considered locally constant and can be locally approximated by convolution. This property can be used to efficiently estimate the space-variant PSF, as described in Sec. 1.5.2.

### 1.3.2 No rotation

A more complicated special case it to disallow rotation and assume that the change of depth is negligible with an implication that also the velocity in the direction of view can be considered zero ( $T(3) = 0$ ). It can be easily seen [32] that in this special case, the PSF can be expressed explicitly using the knowledge of the PSF for one fixed depth of scene.

If the camera does not rotate, that is  $\Omega = [0, 0, 0]^T$ , and moves in only one plane perpendicular to the optical axis ( $T_z(\tau) = 0$ ), equation (1.24) becomes

$$v(x, y, \tau) = \frac{1}{d(x, y, \tau)} \begin{bmatrix} -T_x(\tau) \\ -T_y(\tau) \end{bmatrix}. \quad (1.26)$$

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

The same is true for the corresponding PSF's we get according to relation (1.25). Let us denote the PSF corresponding to an object of the depth equal to the focal length as  $h_0$ . Note that this “prototype” PSF also corresponds to the path covered by the camera. Recall that the depth is given in focal length units. After linear substitution in the

---

<sup>4</sup>Recently Canon announced Hybrid IS that works with translational movements as well.

integral (1.25) we get

$$h(s, t, x, y) = d^2(x, y)h_0(sd(x, y), td(x, y)). \quad (1.27)$$

Equation (1.27) implies that if we recover the PSF for an arbitrary fixed depth, we can compute it for any other depth by simple stretching proportionally to the ratio of the depths.

---

## 1.4 Scene motion

The degradation models we have discussed so far resulted either in the camera motion or in the global scene motion. In many real scenarios, the observed scene is not static but contains moving objects. Local changes inflicted by moving objects are twofold. First, local motion creates additional varying blurring, and second, occlusion of the background may occur. To include these two phenomena in the acquisition model is complicated as it requires segmentation based on motion detection. Most restoration methods assume a rigid transform (e.g. homography) as the warping operator  $W$  in (1.3). If the registration parameters can be calculated, we can spatially align input images. If local motion occurs, the warping operator must implement a non-global transform, which is difficult to estimate. In addition, warping by itself cannot cope with occlusion. A reasonable approach is to segment the scene according to results obtained by local-motion estimation and deal with individual segments separately. Several attempts in this direction were explored in literature recently. Since PSF's may change abruptly, it is essential to precisely detect boundaries, where the PSF's change, and consider boundary effects. An attempt in this direction was for example proposed in [4], where level-sets were utilized. Another interesting approach is to identify blurs and segment the image accordingly by using local image statistics as proposed, e.g., in [14]. All these attempts consider only convolution degradation. If decimation is involved, then space-variant super-resolution was considered, e.g., in [22]. However, this technique assumes that PSF's are known or negligible. A method restoring scenes with local motion, which would perform blind deconvolution and super-resolution simultaneously, has not been proposed yet.

A natural way to avoid the extra burden implied by local motion is to introduce masking as in (1.4). Masking eliminates occluded, missing or corrupted pixels. In the case of local motion, one can proceed in the following way. A rigid transform is first estimated between the input images and inserted in the warping operator. Then discrepancies in the

registered images can be used for constructing masks. More details are provided in the next section on algorithms, 1.5.1.

---

## 1.5 Algorithms

This section outlines the deblurring and super-resolution algorithms that in a way consider spatially varying blur.

As we already mentioned, for the present, there are no super-resolution methods working with unknown spatially varying blur. Deblurring and super-resolution share the same problem of blur estimation and, as we saw in the introduction, it is useful to consider both in the same framework. This section describes deblurring algorithms based on the MAP framework explained in the introduction, where a similar approach could be used for true super-resolution as well.

As the number of blur parameters increases, so does the complexity of estimation algorithms. We will progress our review from simple to more complex scenarios. If the blur is space-invariant except relatively small areas, we can use a space-invariant method supplemented with masking described in the introduction. An algorithm of this type is described in Sec. 1.5.1. If the blur is caused by a more complex camera movement, it generally varies across the image but not randomly. The PSF is constrained by six degrees of freedom of a rigid body motion. Moreover, if we limit ourselves to only rotation, we not only get along with three degrees of freedom, but we also avoid the dependence on a depth map. This case is described in Section 1.5.2. If the PSF depends on the depth map, the problem becomes more complicated. Section 1.5.3 provides possible solutions for two such cases: defocus with a known optical system and blur caused by camera motion. In the latter case, the camera motion must be known or we must be able to estimate from the input images.

### 1.5.1 Super-resolution of a scene with local motion

We start with a super-resolution method [34] that works with space-invariant PSF's and treats possible discrepancies as an error of the convolutional model. This model can be used for super-resolution of a moving object on a stationary background. A similar approach with more elaborated treatment of object boundaries was applied for deblurring in a simplified case of unidirectional steady motion in [1].

We assume the  $K$ -channel acquisition model in (1.4) with  $H_k$  being

convolution with an unknown PSF  $h_k$  of a small support. The corresponding functional to minimize is (1.10) where  $\{\boldsymbol{\theta}_k\} = \{\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_K\}$  consists of registration parameters for images  $g_k$ 's, PSF's  $h_k$ 's, and masks for masking operators  $M_k$ 's. Due to the decimation operators  $D_k$ 's, the acquired images  $g_k$ 's are of lower resolution than the sought-after image  $u$ . Minimization of the functional provides estimates of the PSF's and original image. As the PSF's are estimated in the scale of the original image, positions of PSF's centroids correspond to sub-pixel shifts in the scale of the acquired images. Therefore by estimating PSF's, we automatically estimate shifts with sub-pixel accuracy, which is essential for a good performance of super-resolution. One image from the input sequence is selected as a reference image  $g_r$  ( $r \in 1, \dots, K$ ) and registration is performed with respect to this image. If the camera position changes slightly between acquisitions, which is typically a case of video sequences, we can assume homography model. However, homography cannot compensate for local motion, whereas masking can to some extent. Discrepancies in preregistered (with homography) images give us regions where local motion is highly probable. Masking out such regions and performing simultaneously blind deconvolution and super-resolution, produces naturally looking high-resolution images.

The algorithm runs in two steps:

1. Initialize parameters  $\{\boldsymbol{\theta}_k\}$ : Estimate homography between the reference frame  $g_r$  and each  $g_k$  for  $k \in 1, \dots, K$ . Calculate masks  $M_k$ 's and construct decimation operators  $D_k$ 's. Initialize  $\{h_k\}$  with delta functions.
2. Minimization of  $E(u, \{\boldsymbol{\theta}_k\})$  in (1.10): alternate between minimization with respect to  $u$  and with respect to  $\{\boldsymbol{\theta}_k\}$ . Run this step for a predefined number of iterations or until a convergence criterion is met.

To determine  $M_k$ , we take the difference between the registered image  $g_k$  and the reference image  $g_r$  and threshold its magnitude. Values below 10% of the intensity range of input images are considered as correctly registered and the mask is set to one in these regions; remaining areas are zeroed. In order to attenuate the effect of misregistration errors, the morphological operator “closing” is then applied to the mask. Note that  $M_r$  will be always identity and therefore high-resolution pixels of  $u$  in regions of local motion will be at least mapped to low-resolution pixels of  $g_r$ . Depending on how many input images map to the original image, the restoration algorithm performs any task from simple interpolation to well-posed super-resolution.

The regularization term  $R(\{\boldsymbol{\theta}_k\})$  is a function of  $h_k$ 's and utilizes

relations between all the input images  $g_k$ 's. An exact derivation is given in [23]. Here, we leave the discussion by stating that the regularization term is of the form

$$R(\{h_k\}) \propto \sum_{\substack{1 \leq i, j \leq K \\ i \neq j}} \|h_i * g_j - h_j * g_i\|^2, \quad (1.28)$$

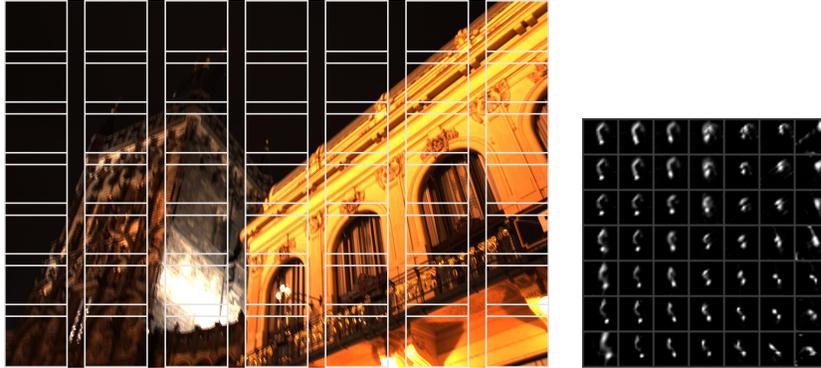
which is convex.



**FIGURE 1.5**

Super-resolution of a scene with local motion. The first row shows five consecutive input frames acquired by a web camera. The second row shows masks (white areas), which indicate regions with possible local motion. The third row shows the estimated original image using simple interpolation (left), super-resolution without masking (central), and proposed super-resolution with masking (right).

We use a standard web camera to capture a short video sequence of a child waving a hand with following setting: 30 FPS, shutter speed 1/30s, and resolution  $320 \times 200$ . An example of 5 low-resolution frames is in the top row in Fig. 1.5. The position of the waving hand slightly differs from frame to frame. Registering the frames in the first step of the algorithm removes homography. Estimated masks in the middle row in Fig. 1.5 show that most of the erroneous pixels are around the waving hand. Note that only the middle frame, which is the reference one and

**FIGURE 1.6**

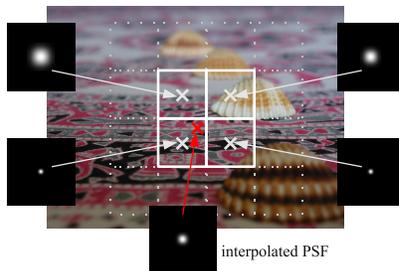
A night photo taken from hand with shutter speed 1.3s. The right image shows PSF's computed within white squares on the left using the algorithm described in Section 1.5.2. Short focal length (36mm equivalent) accents spatial variance of the PSF.

does not have any mask, provides information about the pixels in the region of the waving hand. Comparison of estimating the high-resolution frame with and without masking together with simple interpolation is in the bottom row. Ignoring masks results in heavy artifacts in the region of local motion. On the contrary, masking produces smooth results with the masked-out regions properly interpolated. Remaining artifacts are the result of imprecise masking. Small intensity differences between the images, which set the mask to one, do not always imply that the corresponding areas in the image are properly registered. Such situation may occur for example in regions with a small variance or periodic texture.

### 1.5.2 Smoothly changing blur

This section demonstrates space-variant restoration in situations where the PSF changes gradually without sharp discontinuities, which means that the blur can be locally approximated by convolution. A typical case is the blur caused by camera shake, when taking photos of a static scene without too close objects from hand. Under these conditions, the rotational component of camera motion is dominant and, as was shown in Sec. 1.3.1, the blur caused by camera rotation does not depend on the depth map.

In principle, in this case, the super-resolution methods that use convolution could be applied locally and the results of deconvolution/super-resolution could be fused together. Unfortunately, it is not easy to sew

**FIGURE 1.7**

If the blur changes gradually, we can estimate convolution kernels on a grid of positions and approximate the PSF in the rest of the image (bottom kernel) by interpolation from four adjacent kernels.

the patches together without artifacts on the seams. An alternative way is first to use the estimated PSF's to approximate the spatially varying PSF by interpolation of adjacent kernels (see Fig. 1.7) and then compute the image of improved resolution by minimization of the functional (1.5). The main problem of these naive procedures is that they are relatively slow, especially if applied on too many positions. A partial speed up of the latter can be achieved at the expense of precision by estimating the PSF based solely on blind deconvolution and then upscaling to the desired resolution. This algorithm has not been tested yet.

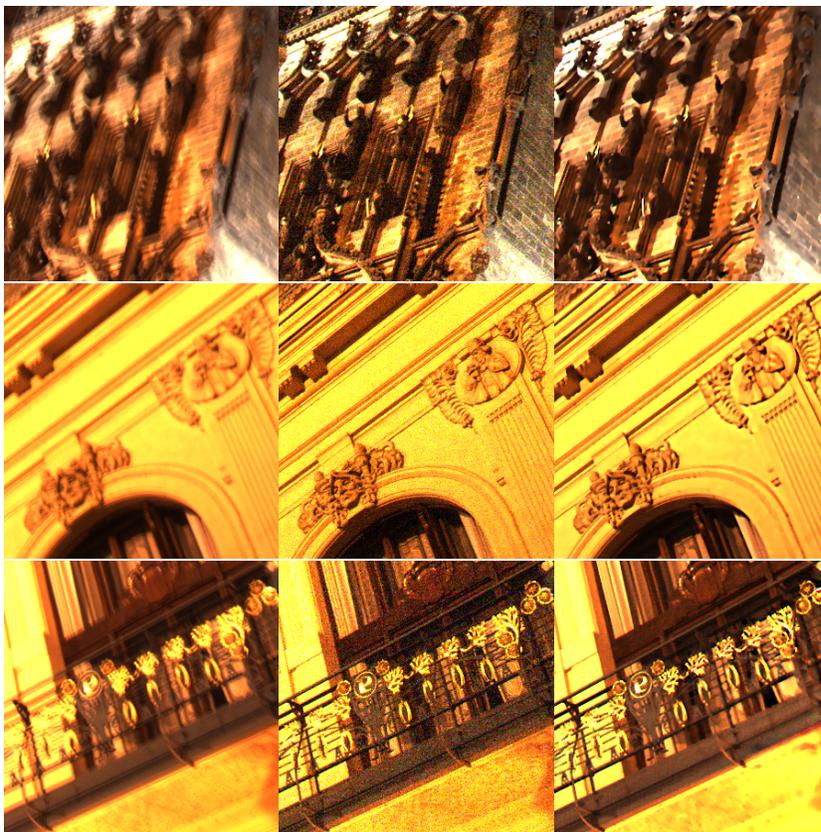
To see, whether the interpolation of the PSF can work in practice and what is the necessary density of the PSF's, we applied this approach for the purpose of image stabilization in [33].

We worked with a special setup that simplifies the involved computations and makes them more stable. It considers the possibility to set the exposure time of the involved camera, which is an acceptable assumption as we can always balance noise with motion blur by setting a suitable shutter speed. In particular, we set the exposure time of one of the images to be so short, that the image is sharp, of course at the expense of noise amplification. The whole idea was explored relatively recently [27, 17, 37].

In Fig. 1.6, we can see a night photo of a historical building taken at ISO 100 with shutter speed 1.3s. The same photo was taken once more at ISO 1600 with 2 stops under-exposure to achieve a hand-holdable shutter time 1/50s. The following algorithm fuses them to get one sharp photo.

The algorithm works in three phases:

1. Robust image registration



**FIGURE 1.8**

Details of restoration. From left to right – the blurred image, noisy image and the result of the algorithm combining them to get a low-noise sharp photo.

2. Estimation of convolution kernels (Fig. 1.6 right) on a grid of windows (white squares in Fig. 1.6 left) followed by an adjustment at places where the estimation failed
3. Restoration of the sharp image by minimizing the functional (1.5). The PSF described by the operator  $H$  for the blurred image is approximated by interpolation from the kernels estimated in the previous step.

We do not describe in detail the image registration here. Just note that the ambiguous registration discussed in Section 1.1.3 does not harm

the procedure because the registration error is compensated by the shift of the corresponding part of the PSF.

The second step is a critical part of the algorithm and we describe it here in more detail. In the example in Fig. 1.6, we took 49 square sub-windows (white squares), in which we estimated kernels  $h_{i,j}$  ( $i, j = 1..7$ ). The estimated kernels are assigned to centers of the windows where they were computed. In the rest of the image, the PSF  $h$  is approximated by bilinear interpolation from blur kernels in the four adjacent sub-windows.

The blur kernel corresponding to each white square is calculated as

$$h_{i,j} = \arg \min_c \|d_{i,j} * c - z_{i,j}\|^2 + \alpha \|\nabla c\|^2, \quad c(\mathbf{x}) \geq 0, \quad (1.29)$$

where  $h_{i,j}(s, t)$  is an estimate of  $h(x_0, y_0; s, t)$ ,  $x_0, y_0$  being the center of the current window  $z_{i,j}$ ,  $d_{i,j}$  the corresponding part of the noisy image, and  $c$  the locally valid convolution kernel.

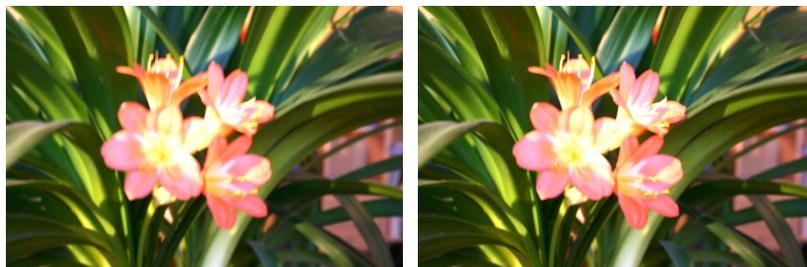
The kernel estimation procedure (1.29) can naturally fail. In a robust system, such kernels must be identified, removed and replaced by for example an average of adjacent (valid) kernels. There are basically two reasons why kernel estimation fails – a lack of texture and pixel saturation. Two simple measures, sum of the kernel values and its entropy turned out to be sufficient to identify such failures.

For minimization of the functional (1.5), we used a variant of the half-quadratic iterative approach, solving iteratively a sequence of linear subproblems, as described for example in [32]. In this case, the decimation operator  $D$  and masking operator  $M$  are identities for both images. Blurring operator  $H$  is identity for the noisy image. The geometric deformation is removed in the registration step. Note that the blurring operator can be speeded up by Fourier transform computed separately on each square corresponding to the neighborhood of four adjacent PSF's [18].

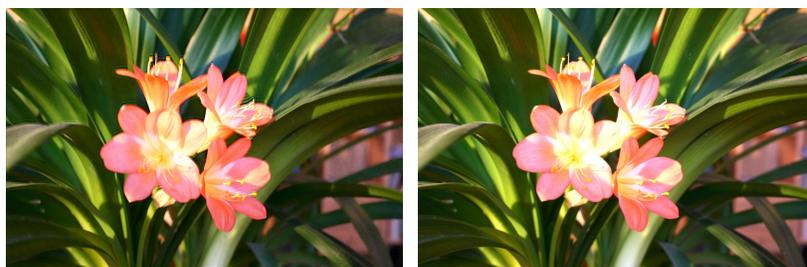
To help reader recognize differences in quite a large photograph ( $1154 \times 1736$  pixels), we show details of the result in Fig. 1.8. Details of the algorithm can be found in [33].

### 1.5.3 Depth-dependent blur

In this section, we demonstrate algorithms working for PSF's that depend on the depth, which implies that besides the restored image we must estimate also an unknown depth map. This includes the blur caused by a camera motion and defocus. Similarly to the previous section, there are no published algorithms that actually increase the physical resolution. On the other hand, a considerable work has been devoted to deblurring.



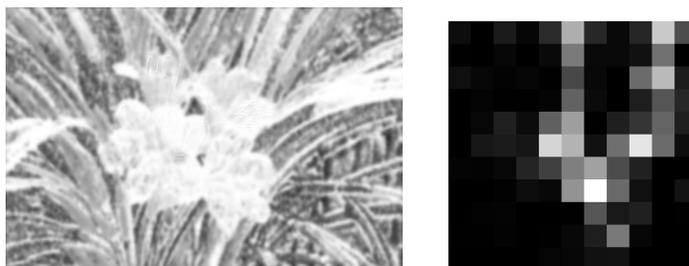
(a) Two motion blurred images with small depth of focus



(b) Result of the algorithm (left) and ground truth (right)

**FIGURE 1.9**

Removing motion blur from images degraded simultaneously by motion blur and defocus by the algorithm described in Sec. 1.5.3.



**FIGURE 1.10**

Depth map corresponding to images in Fig. 1.9 and the PSF estimated locally around the flowers close to the center of the left input image.

In the case of scenes with significant depth variations, the methods requiring PSF's without discontinuities are not suitable. Artifacts would appear especially at the edges of objects. For this case, so far, the only approach that seems to give relatively precise results is based on the

MAP approach, which estimates simultaneously an unknown image and depth map by minimization of a functional in the form (1.10). The main assumption of these algorithms is that the relation between the PSF and the depth is known. One exception is [32], where this relation is estimated for a camera motion constrained to movement in one plane and without rotation. This result is described later in this section.

First this approach appeared in the context of out-of-focus images in [20] proposing to use simulated annealing to minimize the corresponding cost functional. This guarantees global convergence, but in practice, it is prohibitively slow. Later, this approach was adopted by Favaro et al. [8] who modeled the camera motion blur by a Gaussian PSF, locally deformed according to the direction and extent of blur. To make the minimization feasible, they take advantage of special properties of Gaussian PSF's as to view the corresponding blur as an anisotropic diffusion. This model can be appropriate for small blurs corresponding to short locally linear translations. An extension of [8] proposed in [9] segments moving objects but it keeps the limitations of the original paper concerning the shape of the PSF. Other papers related to this type of variational problems can be found also in the context of optical flow estimation, such as [30].

We start our discussion with a difficult case of the blur caused by an unconstrained camera motion. If the cameras' motion and parameters (focal length, resolution of the sensor, initial relative position of cameras) are known, we can, at least in theory, compute the PSF as a function of depth map and solve the MAP problem (1.10) for an unknown image  $u$  and a parameter set  $\{\theta_k\}$  corresponding now to a depth map for one of observed images  $g_k$ . An issue arises from the fact that the PSF is a function of not only depth but also of coordinates  $(x, y)$ . In other words, different points of the scene draw different apparent curves during the motion even if they are of the same depth. In addition, the depth map is no longer common for all the images and must be transformed to a common coordinate system before computing  $H_k$  using (1.24) and (1.25). The numerical integration of the velocity field is unfortunately quite time-consuming. A solution could be to precompute the PSF for every possible combination of coordinates  $(x, y)$  and depth values. As it is hardly possible, a reasonable solution seems to store them at least on a grid of positions and compute the rest by interpolation. The density of this grid would depend on application.

In [32] we show that obstacles of the general case described above can be avoided by constraining camera motion to only one plane without rotations. This corresponds to vibrations of a camera fixed for example to an engine or machine tool.

A nice property of this case is that the PSF actually changes only its



(a) Two out-of-focus images taken with aperture  $F/5.0$  and  $F/6.3$



(b) Results of the algorithm (left) and ground truth (right).

### FIGURE 1.11

Removing out-of-focus blur by the algorithm described in Sec. 1.5.3. The extent of blur increases from front to back.

scale proportionally to inverse depth (see Sec. 1.3.2). As a consequence, if we estimate the PSF for one depth, we know the whole relation between the PSF and depth (1.27). In addition, the depth map is common for all images.

The algorithm works in three steps:

1. PSF estimation at a fixed depth using the blind deconvolution algorithm [24]. A region where the PSF is estimated is specified by user (depth variations must be negligible). This region must be in focus, otherwise we would not be able to separate motion and out-of-focus blur.
2. Rough depth map estimation using a simpler method assuming

that the blur is space-invariant in a neighborhood of each pixel (also described in [32])

3. Minimization of the functional (1.10) to get a sharp image and a more precise depth map.

What concerns the degradation operators  $G_k$  in the functional (1.10), the operators  $D_k$  and  $M_k$  are identities and we work only with a blurring. The minimization proceeds alternately by conjugate gradients in the image subspace and steepest descent in the depth map subspace. We chose total variation for image regularization and Tikhonov regularization for the depth map. Note that the depth map we estimate is relative to the distance of the object on which we estimated the PSF in the first step.

An example in Fig. 1.9 illustrates the performance of the algorithm compared to ground truth. Besides the motion blur the photographs contain also defocus but the defocus is common for both images and is not to be removed. Figure 1.10 shows the convolution kernel estimated in the first step of the algorithm (right) and the recovered depth map (left).

Figure 1.11 shows a result of the above described algorithm [32] modified to remove defocus (there is no motion blur in the images). It assumes that the PSF of the lens can be model by a pillbox function as a function of depth according to relation (1.13). For minimization of the corresponding functional, we use the same method as in [32]. Details are given in [31].

---

## 1.6 Conclusion

Bringing this all together, for the present, the restoration of images blurred by spatially varying blur is not resolved satisfactorily for most cases. In this chapter, we went through the special cases where at least a partial solution is known and we explained the basic principles published algorithms are based on. We showed that from Bayesian perspective it is usefull to consider deblurring and super-resolution in one framework.

Many open questions and unresolved problems remain. A large number of blur parameters we need to estimate brings significant errors to the solution and for the present there is no analysis of super-resolution limits for these cases. It may turn out that in many cases super-resolution does not bring much more than mere deblurring. We have shown several algorithms that estimated space-variant blur considering only the

deblurring problem. It will be interesting to see if the extension to true super-resolution really works. Especially difficult is the situation when the changes in the PSF are not continuous, e.g. several independently moving objects (motion blur) or even worse, if the PSF depends on the depth of scene (defocus, camera motion).



---

## ***Bibliography***

- [1] Amit Agrawal, Yi Xu, and Ramesh Raskar. Invertible motion blur in video. *ACM Trans. Graph.*, 28(3):1–8, 2009.
- [2] M. Ahmed and A. Farag. Non-metric calibration of camera lens distortion. In *Image Processing, 2001. Proceedings. 2001 International Conference on*, volume 2, pages 157–160 vol.2, Oct 2001.
- [3] Mark R. Banham and Aggelos K. Katsaggelos. Digital image restoration. *IEEE Signal Process. Mag.*, 14(2):24–41, March 1997.
- [4] Leah Bar, Nir A. Sochen, and Nahum Kiryati. Restoration of images with piecewise space-variant blur. In *SSVM*, pages 533–544, 2007.
- [5] Moshe Ben-Ezra, Zhouchen Lin, and Bennett Wilburn. Penrose pixels super-resolution in the detector layout domain. In *Proc. IEEE Int. Conf. Computer Vision*, pages 1–8, 2007.
- [6] M. Born and E. Wolf. *Principles of Optics*. Pergamon Press, Oxford, 1980.
- [7] Dmitry Datsenko and Michael Elad. Example-based single document image super-resolution: a global map approach with outlier rejection. *Multidimensional Syst. Signal Process.*, 18(2-3):103–121, 2007.
- [8] Paolo Favaro, Martin Burger, and Stefano Soatto. Scene and motion reconstruction from defocus and motion-blurred images via anisotropic diffusion. In Tomáš Pajdla and Jiří Matas, editors, *ECCV 2004, LNCS 3021, Springer Verlag, Berlin Heidelberg*, pages 257–269, 2004.
- [9] Paolo Favaro and Stefano Soatto. A variational approach to scene reconstruction and image segmentation from motion-blur cues. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, volume 1, pages 631–637, 2004.
- [10] Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T. Roweis, and William T. Freeman. Removing camera shake from a single photograph. *ACM Trans. Graph.*, 25(3):787–794, 2006.

- [11] William T. Freeman, Thouis R. Jones, and Egon C Pasztor. Example-based super-resolution. *IEEE Comput. Graph. Appl.*, 22(2):56–65, 2002.
- [12] D. J. Heeger and A. D. Jepson. Subspace methods for recovering rigid motion. *International Journal of Computer Vision*, 7(2):95–117, 1992.
- [13] Kwang In Kim and Younghee Kwon. Example-based learning for single-image super-resolution. In *Proceedings of the 30th DAGM symposium on Pattern Recognition*, pages 456–465, Berlin, Heidelberg, 2008. Springer-Verlag.
- [14] A. Levin. Blind motion deblurring using image statistics. In *NIPS*, pages 841–848, 2006.
- [15] Anat Levin, Robert Fergus, Frédo Durand, and William T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM Trans. Graph.*, 26(3):70, 2007.
- [16] Anat Levin, Peter Sand, Taeg Sang Cho, Frédo Durand, and William T. Freeman. Motion-invariant photography. In *SIGGRAPH '08: ACM SIGGRAPH 2008 papers*, pages 1–9, New York, NY, USA, 2008. ACM.
- [17] S. H. Lim and D. A. Silverstein. Method for deblurring an image. US Patent Application, Pub. No. US2006/0187308 A1, Aug 24 2006.
- [18] James G. Nagy and Dianne P. O’Leary. Restoring images degraded by spatially variant blur. *SIAM J. Sci. Comput.*, 19(4):1063–1082, 1998.
- [19] T. Q. Pham. *Spatiotonal adaptivity in super-resolution of under-sampled image sequences*. PhD thesis, TU Delft, 2006.
- [20] A. N. Rajagopalan and S. Chaudhuri. An MRF model-based approach to simultaneous recovery of depth and restoration from defocused images. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(7), July 1999.
- [21] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.
- [22] Huanfeng Shen, Liangpei Zhang, Bo Huang, and Pingxiang Li. A MAP approach for joint motion estimation, segmentation, and super resolution. *IEEE Trans. Image Process.*, 16(2):479–490, February 2007.

- [23] F. Šroubek, G. Cristobal, and J. Flusser. A Unified Approach to Superresolution and Multichannel Blind Deconvolution. *IEEE Transactions on Image Processing*, 16:2322–2332, September 2007.
- [24] Filip Šroubek and Jan Flusser. Multichannel blind iterative image restoration. *IEEE Trans. Image Process.*, 12(9):1094–1106, September 2003.
- [25] Filip Šroubek and Jan Flusser. Multichannel blind deconvolution of spatially misaligned images. *IEEE Trans. Image Process.*, 14(7):874–883, July 2005.
- [26] Muralidhara Subbarao and G. Surya. Depth from defocus: a spatial domain approach. *International Journal of Computer Vision*, 3(13):271–294, 1994.
- [27] M. Tico, M. Trimeche, and M. Vehvilainen. Motion blur identification based on differently exposed images. In *Proc. IEEE Int. Conf. Image Processing*, pages 2021–2024, 2006.
- [28] D. Tschumperlé and R. Deriche. Diffusion PDE’s on vector-valued images. *IEEE Signal Process. Mag.*, 19(5):16–25, September 2002.
- [29] David Tschumperlé and Rachid Deriche. Vector-valued image regularization with pdes: A common framework for different applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):506–517, 2005.
- [30] Damon L. Tull and Aggelos K. Katsaggelos. Regularized blur-assisted displacement field estimation. In *Proc. Int. Conf. Image Processing*, volume 3, pages 85–88, September 1996.
- [31] Michal Šorel. *Multichannel Blind Restoration of Images with Space-Variant Degradations*. PhD thesis, Charles University in Prague, 2007.
- [32] Michal Šorel and Jan Flusser. Space-variant restoration of images degraded by camera motion blur. *IEEE Trans. Image Process.*, 17(2):105–116, February 2008.
- [33] Michal Šorel and Filip Šroubek. Space-variant deblurring using one blurred and one underexposed image. In *Proc. Int. Conf. Image Processing*, 2009.
- [34] Filip Šroubek, Jan Flusser, and Michal Šorel. Superresolution and blind deconvolution of video. In *Proc. Int. Conf. on Pattern Recognition*, pages 1–4, 2008.

- [35] Masahiro Watanabe and Shree K. Nayar. Rational filters for passive depth from defocus. *International Journal of Computer Vision*, 3(27):203–225, 1998.
- [36] Wonpil Yu. An embedded camera lens distortion correction method for mobile computing applications. *Consumer Electronics, IEEE Transactions on*, 49(4):894–901, Nov. 2003.
- [37] Lu Yuan, Jian Sun, Long Quan, and Heung-Yeung Shum. Image deblurring with blurred/noisy image pairs. In *SIGGRAPH '07: ACM SIGGRAPH 2007 papers*, page 1, New York, NY, USA, 2007. ACM.
- [38] Barbara Zitová and Jan Flusser. Image registration methods: a survey. *Image and Vision Computing*, 11(21):977–1000, 2003.