Contents

List of contributors *page* ii

6 Removing camera shake in smartphones without hardware stabilization
F. Šroubek and J. Flusser 1

6.1 Introduction 1
6.2 Image acquisition model 1
6.2.1 Space-invariant model 2
6.2.2 Space-variant model 2
6.3 Inverse problem 4
6.3.1 MAP and beyond 4
6.3.2 Getting more prior information 6
6.3.3 Patch-based 9
6.4 Pinhole camera model 10
6.5 Smartphone application 13
6.5.1 Space-invariant implementation 13
6.5.2 Space-variant implementation 15
6.6 Evaluation 17
6.7 Conclusions 20

List of contributors

Filip Šroubek
Institute of Information Theory and Automation, Czech Academy of Sciences

Jan Flusser
Institute of Information Theory and Automation, Czech Academy of Sciences

6 Removing camera shake in smartphones without hardware stabilization

Filip Šroubek and Jan Flusser

6.1 Introduction

Processing images becomes an every day practice in a wide range of applications in science and technology and we rely on our images with ever growing emphasis. Our understanding of the world is however limited by measuring devices that we use to acquire images. Inadequate measuring conditions together with technological limitations of the measuring devices result in acquired images that represent a degraded version of the "true" image.

Blur induced by camera motion is a frequent problem in photography mainly when the light conditions are poor. As the exposure time increases, involuntary camera motion has a growing effect on the acquired image. Image stabilization (IS) devices that help to reduce the motion blur by moving the camera sensor in the opposite direction are becoming more common. However, such hardware remedy has its limitations as it can compensate only for motion of a very small extent and speed. Deblurring the image offline using mathematical algorithms is usually the only choice we have to obtain a sharp image. Motion blur can be modeled by convolution and the deblurring process is called deconvolution, which a well-known ill-posed problem. In general, the situation is even more complicated, since we usually have no or limited information about the blur shape.

We can divide the deconvolution methods into two categories: methods that estimate the blur and the sharp image directly from the acquired image (blind deconvolution) and methods that use information from other sensors to estimate the blur (semi-blind deconvolution).

The main contribution of this chapter is to illustrate that blur estimation with built-in inertial sensors is possible and to implement image deblurring on a smartphone, which works in practical situations and is relatively fast to be acceptable for a general user.

6.2 Image acquisition model

In order to perform successful image deblurring, it is important to correctly model the acquisition process. A commonly used model is the following. Let

$u(\mathbf{x}) : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$, $\mathbf{x} = [x, y]$, be an original *latent* image describing a real scene, where Ω is a rectangular image support. A first-order approximation of degradation acting on u is twofold: degradation linear operator H and additive noise n. Then the output of acquisition is a degraded image g given by a relation

$$g = Hu + n\,. \tag{6.1}$$

The difficulty with H is that it is ill-conditioned, which means that during inversion noise n gets amplified and the solution is unstable. We face an *ill-posed inverse* problem that requires special handling.

6.2.1 Space-invariant model

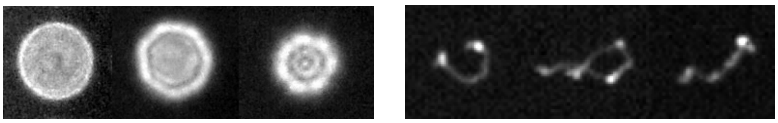
The most common type of degradation, which is considered in this chapter, is *convolution*:

$$[Hu](x, y) = h * u = \int_\Omega u(s, t)h(x - s, y - t)\mathrm{d}s\mathrm{d}t\,, \tag{6.2}$$

This definition extends to any number of dimensions and not just $\mathbb{R}^2$. For example, in confocal microscopy convolution is in $\mathbb{R}^3$. Function h is a convolution kernel (or simply blur) and defines the behavior of the convolution operator. It is also called a *point spread function* (PSF), because h is an image the device would acquire after measuring an ideal point source $\delta(\mathbf{x})$ (delta function). Image blur due to camera motion or improper camera focus setting can be in simple cases modeled by convolution. The PSF size influences the degree of blurring and the PSF shape reflects the physical nature of blurring. For example, out-of-focus camera lens causes PSFs of various shapes depending on lens aperture, while camera motion causes curvy PSF, where the curve shape is related to the trajectory of the motion; see Fig. 6.1. There are a wide range of imaging devices in which the acquisition process can be modeled by convolution. Apart from devices with classical optical systems, such as digital cameras, optical microscopes or telescopes, convolution degradation occurs also in atomic force microscopy (AFM) or scanning tunneling microscopy (STM), where the PSF shape is related to the measuring tip shape. Media turbulence (e.g. atmosphere for terrestrial telescopes) can cause blurring that can be modeled by convolution, and there are many more examples.

6.2.2 Space-variant model

To make convolution more general, it is often necessary to allow the PSF to change over the image. This is called *space-variant* convolution, though strictly speaking it is not mathematical convolution any more. Using space-variant convolution we can model more general degradations, such as blur induced by complex camera motion and rotation, out-of-focus blur in a wide-depth scene, or blur due to hot-air turbulence.



(a) out-of-focus blurs (b) camera shake blurs

Figure 6.1 Examples of real camera blurs: (a) blur caused by out-of-focus lens with different lens parameters (focal length and aperture size), notice polygonal shape clearly visible in the central image, which corresponds to the aperture opening of 7-blade diaphragm; (b) blurs caused by camera motion during exposure.

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

$$[Hu](x,y) = \int_{\Omega} u(s,t)h(x-s,y-t,s,t)\mathrm{d}s\mathrm{d}t \tag{6.3}$$

with h now dependent on the position (x,y) in the image. Convolution in (6.2) is a special case of (6.3) with $h(s,t,x,y) = h(s,t)$ for any position (x,y). It is valid to refer to $h(s,t,x,y)$ as a space-variant PSF (or kernel), for the function $h(s,t,x,y)$ taken as a function of two variables with fixed x and y, describes how a delta function at (x,y) spreads its energy in space after degradation by H. In many real scenarios h changes slowly with respect to (x,y) as discussed in Sec. 6.3.3. In this case the operator H can be locally approximated by convolution with $\tilde{h}(s,t) = h(s,t,x,y)$ in a neighborhood of the point (x,y).

It is worth noting that we can interpret the physical meaning of the PSF in two different ways. The standard view is that the PSF tells us how a point source is spread in space. The blurred image is obtained by spreading every pixel and summing contribution of all pixels. The other interpretation is that the PSF is a weighted window. The blurred image is calculated by performing weighted averaging around every pixel. In the case of space-invariant convolution both views are equivalent. However in the case of space-variant convolution, this is not true and we assume the first interpretation, which describes the physical nature of the blurring phenomenon.

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 in (6.3) becomes a sum, operator H corresponds to a matrix and u to a vector obtained by concatenating columns of the image into one long vector. Using the vector-matrix notation, (6.1) writes as

$$\mathbf{g} = \mathbf{H}\mathbf{u} + \mathbf{n}\,. \tag{6.4}$$

In the space-invariant case, $\mathbf{H}$ is a convolution matrix (block-Toeplitz matrix with Toeplitz blocks) and each column of $\mathbf{H}$ 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 $h(s,t,x,y)$.

6.3 Inverse problem

A standard formulation of the image deblurring problem is a stochastic one [1] assuming that images and PSFs are random vector fields [2] with known prior probability density functions $p(u)$ and $p(h)$, respectively. The Bayesian paradigm dictates that the inference on the latent image and PSF should be based on the posterior probability

$$p(u,h|g) \propto p(g|u,h)p(u)p(h)\,, \tag{6.5}$$

where u and h are assumed to be independent. The conditional density $p(g|u,h)$ is given by our model (6.1) and is equal to the probability density function of noise n, which is typically normally distributed, $p(n) = N(0,\sigma^2)$. The prior $p(u)$ forces some type of image statistical characteristics, typically we assume sparse distribution (e.g. Laplacian) of image derivatives [3]. The prior $p(h)$ can be similar to $p(u)$, but it is often rectified to force positivity. Estimating the pair $(\hat{u},\hat{h})$ is equivalent to maximizing the posterior $p(u,h|g)$, which is commonly referred to as the *maximum a posteriori* (MAP) approach. Note that maximization of the posterior is equivalent to minimization of $-\log p(u,h|g)$.

6.3.1 MAP and beyond

The classical solution to the blind deconvolution problem is maximizing the posterior probability $p(u,h|g)$ with respect to both u and h (denoted as $MAP_{u,h}$ approach). However, the posterior has very uneven shape with many local peaks and alternating maximization often returns an incorrect solution.

Typically the priors are defined in some other domain than the image domain and are assumed to be independent and sparse, $p(u) = \Pi_i p(\phi_i(u))$, where ϕ_i is a function mapping the image into the other domain. A common choice of ϕ is an image gradient with p favoring sparse distributions, i.e., $\log p(u) = -\sum_i \frac{1}{2\gamma}|\nabla u_i|^p + C$ for $0 \le p \le 1$. In the vector matrix notation this can be expressed as $\log p(u) = -\frac{1}{2\gamma}\mathbf{u}^T\mathbf{L}_u^p\mathbf{u}$, where $\mathbf{L}_u^p$ is a symmetrical sparse matrix with elements related to ∇u. In the case of $p = 2$ (Gaussian prior), $\mathbf{L}_u^2$ is independent of u and is equal to the Laplacian, which we denote simply as $\mathbf{L}$. A detailed derivation of these relations is provided in [4].

The most common method to find $MAP_{u,h}$ is to alternate between minimization of $-\log p(u,h|g)$ with respect to u and h. It is a well-known fact that this leads to the so-called "no-blur" solution, i.e., the most probable solution is that h is a delta function and $u = g$, which is clearly a solution we better want to avoid. It is important to note that the posterior $p(u,h|g)$ has a local extreme for the true u and h, which can be reached if we initialize the alternating minimization close to the true solution. Many authors [5, 6, 7, 8, 9] tried to alleviate this problem by searching for more sophisticated priors that would better match the natural image statistics. Unfortunately, mainly due to the fact that blur reduces image variance, most of these priors still favor the blurred image over the sharp one as

pointed out by Levin *et al.* [9]. Levin further demonstrated in [10] that a proper estimator matters more than the shape of priors. Marginalizing the posterior with respect to the latent image u and maximizing thus $p(h|g) = \int p(u, h|g)du$ leads to the correct solution. This is referred to as MAP_h approach. Once we estimate $\hat{h}$, we can estimate the sharp image as

$$u^{MAP} = \underset{u}{\operatorname{argmax}}\, p(u, h = \hat{h}|g). \tag{6.6}$$

The marginalized probability $p(h|g)$ can be expressed in a closed form only for simple priors that are not sparse. Otherwise approximation methods such as variational Bayes [11, 6] or the Laplace approximation [12] must be used. For example, using the Laplace approximation

$$\log p(h|g) \approx \log p(u = u^{MAP}, h|g) - \frac{1}{2}\log|A| + C\,, \tag{6.7}$$

where $|A|$ denotes determinant of the variance of $p(u = u^{MAP}, h|g)$ with respect to u. If we assume a simple Gaussian prior, $\log p(u) = -\sum_i \frac{1}{2\gamma}|\phi_i(u)|^2 + C = -\frac{1}{2\gamma}\mathbf{u}^T\mathbf{L}\mathbf{u} + C$ then the Laplace approximation is exact with $\mathbf{A} = \frac{\mathbf{H}^T\mathbf{H}}{\sigma^2} + \frac{\mathbf{L}}{\gamma}$, u^{MAP} has a closed-form solution and so does $p(h|g)$. In a general case of sparse priors, we can use the above equation as an update equation for $\hat{h} = \operatorname{argmax}_h p(h|g)$ and iteratively update also u^{MAP} in (6.6). The variational Bayes factorizes the posterior $p(u, h|g)$ and approximates it by $q(u)q(h)$, which can be then solved by alternating maximization. As derived for example in [5], the variational Bayes leads to a very similar update equation (6.7) for h, where the second term $\frac{1}{2}\log|A|$ is replaced by $\frac{1}{2}\mathbf{h}^T\mathbf{cov}(q(u))\mathbf{h}$. In the case of the simple Gaussian prior, $\mathbf{cov}(q(u)) = \mathbf{A}^{-1}$. Both in the Laplace approximation (6.7) and variational Bayes the second term plays a crucial role, since it favors kernels with more blur and penalizes the "no-blur" solution. This brings us to an interesting conclusion that the iterative algorithms of the approximation techniques are very similar to the classical $MAP_{u,h}$ approach with an additional term that penalizes blurs close to delta functions. This conclusion suggests that even $MAP_{u,h}$ should work if a better numerical method is applied.

There are several other modifications (or maybe better to say tweaks) used in blind deconvolution methods, which are often only briefly mentioned in the literature but we believe play a key role and largely improve results in practice. The first one is that in the blur estimation step we use ∇u instead of u. From the perspective of the data model this formulation is equivalent, since $\nabla g = h * \nabla u$. One possible reason for the advantage of derivatives is that it fits better with our model that the image prior is a product of independently and identically distributed variables. An algebraic explanation is that the deconvolution system solved in every iteration is better conditioned. The second modification is that the current blind deconvolution methods, such as [13], start the estimation process with a very high noise variance (σ^2) and slowly decrease it to the correct level. This idea is similar to what is used in simulated annealing and has an effect that low frequencies (overall shape) of blurs are estimated first and then

the high frequencies (details). The third modification is that the blur and image are estimated in a coarse-to-fine framework, using pyramidal representation of the input blurred image. This has similar effect as the second modification. However numerically, the coarser levels have lower dimension and are thus easier to solve. It is thus possible that a combination of the coarse-to-fine approach and gradual variance decrease makes the whole system more robust in practice. Last but not least, it is necessary to constrain h to avoid local minima. Recent methods propose to use prior $p(h)$ such that only positive values are allowed and sometimes support constraints are also added. Blur kernels are in general much smaller then the image and it is therefore wise to constrain the blur support and effectively decrease the dimensionality of the problem.

6.3.2 Getting more prior information

There are various ways to get additional information about the degradation process so that the blind deconvolution problem addressed in the previous section 6.3.1 is better conditioned and thus easier to solve.

One approach is to use multiple degraded images of the same scene. This is called *multichannel blind deconvolution*. The estimation of blurs from multiple observed images is based on a simple but fundamental idea. Let us assume that we have $K > 1$ images g_k that represent the original image u according to our space-invariant model

$$g_k = h_k * u + n\,. \tag{6.8}$$

The PSF h_k is different for every k. Let $\hat{h}_k$ denote our estimated PSFs. In the case of no noise holds that for every pair of images g_i, g_j, $i \neq j$

$$g_i * \hat{h}_j - g_j * \hat{h}_i = 0\,, \tag{6.9}$$

if $\hat{h}_k = h_k$. This simple relation, which exists only in the multichannel case, give us means to estimate PSFs directly from blurred images and forms the key idea of every multichannel blind method discussed below.

One of the earliest intrinsic multichannel (MC) blind deconvolution methods [14] was designed particularly for images blurred by atmospheric turbulence. Harikumar *et al.* [15] proposed an indirect algorithm, which first estimates the blur functions (published earlier for 1D signals in [16]) and then recovers the original image by standard nonblind methods.

Giannakis *et al.* [17] developed similar algorithm based on Bezout's identity of coprime polynomials which finds restoration filters and by convolving the filters with the observed images recovers the original image. Pai *et al.* [18] suggested two MC restoration algorithms that estimate directly the original image from the null space or from the range of a special matrix. Another direct method based on the greatest common divisor was proposed in [19]. Interesting approaches based on the ARMA (autoregressive moving average) model are given in [20]. MC blind deconvolution based on the Bussgang algorithm was proposed in [21],

which performs well on spatially uncorrelated data, such as binary text images and spiky images. Most of the algorithms lack the necessary robustness since they do not include any noise assumptions in their derivation and miss regularization terms. Šroubek *et al.* proposed an iterative MC algorithm [22] that performs well even on noisy images. It is based on least-squares deconvolution by anisotropic regularization of the image and between-channel regularization of the blurs. Another drawback of the MC methods is that the observed images must be spatially aligned, which is seldom true. A first attempt in this direction was done by Šroubek *et al.* in [4], where they proposed MC blind deconvolution of images, which are mutually shifted by unknown vectors. The same team extended this idea to superresolution in [23]. In superresolution, the physical resolution of the image is increased, which is equivalent to considering both convolution and downsampling in the degradation operator H.

Another possibility is to acquire a pair of images: one correctly exposed but blurred and one underexposed (noisy) but sharp image. Then we can apply the above MC blind deconvolution methods, which are even better posed, as was demonstrated in [24, 25, 26].

Blind deconvolution in the MC framework is in general a well-posed inverse problem due to existence of the relation (6.9). However, in many practical situations we do not have multiple observations of the same scene, which would differ only by the convolution kernel, and we must revert to the single-channel methods in Sec. 6.3.1 or try *parametric methods*.

Parametric methods assume a certain set of plausible blurs that are fully described by a few parameters. Inference is done on the parameters and not on the blur function itself, which would be an intractable problem since the dimensionality of the blur functions is too high. A classical example is the blur caused by camera motion, which 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 estimate the depth map, which makes the algorithm complicated and time consuming. Nevertheless, there are algorithms that work satisfactorily, assuming certain additional constraints on the camera motion. We refer the interested readers to [27, 28, 29] and references therein.

Here we describe a more practical approach that exploits the fact that certain types of camera motion can be neglected in practice. The most common assumption is that all camera translations can be neglected, making the blur independent of scene depth. Validity of this assumption is discussed in Sec. 6.4. Many devices, such as modern smartphones, are now equipped with inertial sensors (gyroscopes and accelerometers) that can give us a very accurate information about camera motion. If we are able to reconstruct camera path then we can recover blur and perform nonblind image deblurring. This idea was recently described in [30] using an expensive measuring apparatus consisting of a DSLR camera and a set of inertial sensors, where image deblurring is performed offline

on a computer. Another possibility is to attach an auxiliary high-speed camera of lower resolution to estimate the PSF using optical flow techniques as discussed in [31, 32]. Later in Sec. 6.5 we demonstrate that image deblurring is feasible on modern smartphones and not requiring any other devices.

A versatile approach applied recently in [33] is to express the operator of blurring in a specially chosen set of PSF basis B_i

$$Hu = \sum_i d_i B_i u \tag{6.10}$$

which allows us to work with spatially varying blur in a manner similar to common convolution. In this case, B_i are operators (in the discrete case matrices) that perform image warping such that images $B_i u$ 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 set of weights d_i, which are our unknown parameters, are referred to as kernels or motion density functions.

Whyte et al. [34] consider rotations about three axes up to several degrees and describe blurring by the corresponding three-dimensional kernel. For blind deconvolution, it uses a straightforward analogy of the well-known blind deconvolution algorithm [6] based on marginalization over the latent sharp image. The only difference is that it uses (6.10) instead of convolution. For deblurring, following the kernel estimation phase, it uses the corresponding modification of the Richardson-Lucy algorithm.

Gupta et al. [35] 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 [6] and estimate the corresponding sharp image patches. In the second step, they estimate the kernels d_i from (6.10) 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 (6.10) is that it is very robust with respect to local non-existence of texture as well as local inaccuracies of the used model, such as sensor saturation for example. On the other hand, it may be considerably more time consuming than algorithms based on local approximation by convolution described in the next section. Another disadvantage is that the

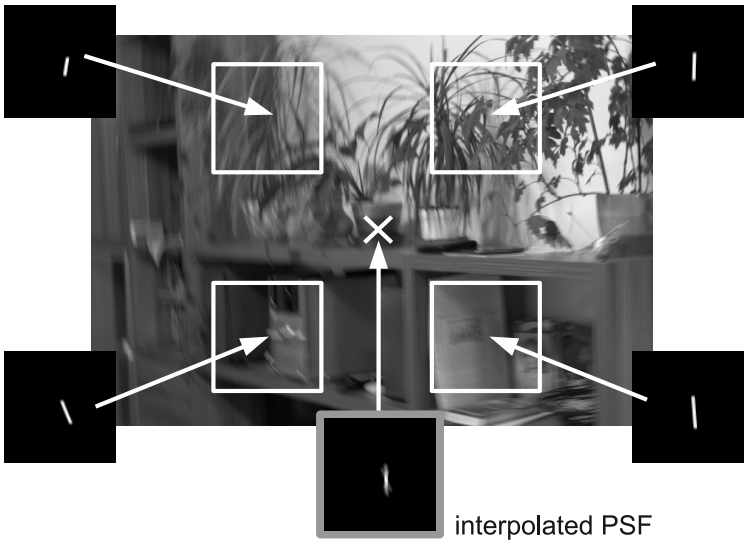


Figure 6.2 Kernel interpolation: If image blur varies slowly, we can estimate convolution kernels on a grid of positions and approximate kernels in the rest of the image by interpolation of four adjacent kernels.

actual motion may be more complicated and it is difficult to combine (6.10) with other models.

6.3.3 Patch-based

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 out-of-focus blur it holds only for approximately planar scenes.

The kernels are usually estimated at a set of regularly spaced positions. For estimation we can use one of many available blind deconvolution methods, working either with a single image [6, 13], 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 [24, 25, 26].

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 PSF. The patch size is chosen such that the patches overlap and the reconstructed image is generated by blending the reconstructed patches. Blending must be tuned to minimize blocking artifacts that tend to appear around patch boundaries.

A more elaborate but only slightly more computationally demanding method

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. 6.2. 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) [36, 37]. 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

$$Hu = \sum_i (w_i u) * h_i \,. \tag{6.11}$$

For linear interpolation, the weight functions $w_i(x, y)$ satisfy the constraint $\sum_i w_i(x, y) = 1$ for an arbitrary 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 [26], the authors 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 [38] is shown in [39].

If the blur changes faster across the image, linear interpolation is not realistic and produces false PSFs, which consequently leads to artifacts in the reconstructed image. More advanced techniques, such as shape-preserving PSF morphing proposed in [40], should be used instead.

6.4 Pinhole camera model

Let us consider the image a camera captures during its exposure window. Light from a scene point $\mathbf{x}_w = [x_w, y_w, z_w]^T$ projects on the image plane at a location $\mathbf{x} = [x, y]^T$; see Fig. 6.3. Using homogeneous coordinates in the image plane $\bar{\mathbf{x}} = [d\mathbf{x}^T, d]^T$, the relation to $\mathbf{x}_w$ is given by

$$\bar{\mathbf{x}} = \mathbf{K}[\mathbf{R}\mathbf{x}_w + \mathbf{t}] \,, \tag{6.12}$$

where $\mathbf{R}$ (3×3) and $\mathbf{t}$ (3×1) are the camera rotation matrix and translation vector, respectively. Upper triangular matrix $\mathbf{K}$ (3×3) is the camera intrinsic matrix. The third element d of the homogeneous coordinates corresponds to distance. The camera intrinsic matrix is typically of the form

$$\mathbf{K} = \begin{bmatrix} fs_x & 0 & c_x \\ 0 & fs_y & c_y \\ 0 & 0 & 1 \end{bmatrix} , \tag{6.13}$$

where f is the camera focal length, $[c_x, c_y]$ is the image center and $[s_x, s_y]$ is the pixel scale.

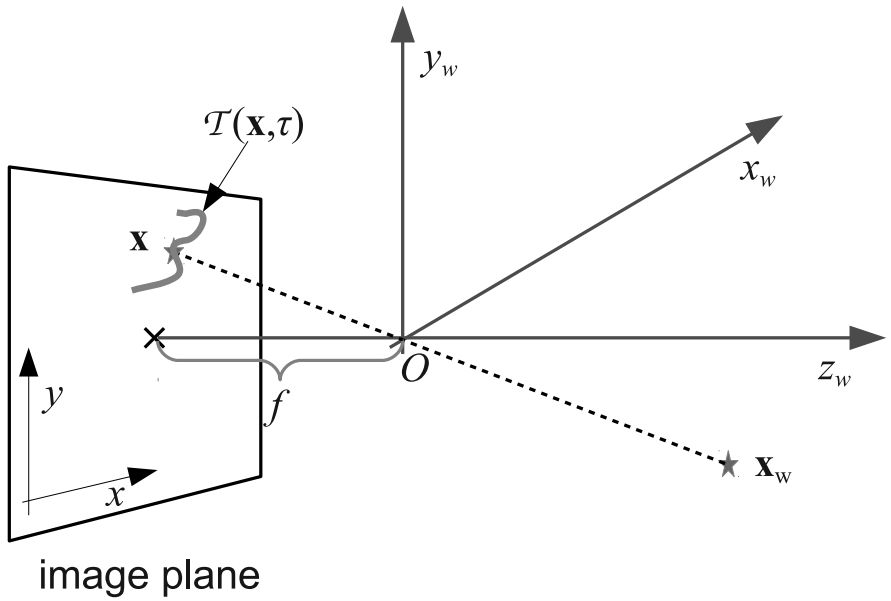


Figure 6.3 Pinhole camera model: A 3D point $\mathbf{x}_w$ in the world coordinate system x_w, y_w, z_w projects into the image plane x, y at the position $\mathbf{x}$. The projection is defined by a camera intrinsic matrix, which is primarily a function of the camera focal length f. Due to camera motion during exposure the projected point draws a trace (2D curve) $\mathcal{T}(\mathbf{x}, \tau)$, which is a function of time τ.

During the exposure window the camera position and orientation may change and therefore the extrinsic parameters $\mathbf{R}$ and $\mathbf{t}$ are function of time τ. The projected point $\mathbf{x}$ moves along a curve parametrized by τ, which we denote as $\mathcal{T}(\mathbf{x}, \tau)$ and call it a point trace:

$$\mathcal{T}(\mathbf{x}, \tau): \quad \bar{\mathbf{x}}(\tau) = \mathbf{K}[\mathbf{R}(\tau)\mathbf{K}^{-1}\bar{\mathbf{x}}_0 + \frac{1}{d_0}\mathbf{t}(\tau)]\,, \tag{6.14}$$

where $\bar{\mathbf{x}}_0 = [\mathbf{x}^T, 1]^T = [x, y, 1]$ is the initial location of the point in the image plane using normalized homogenous coordinates and d_0 is the initial distance of the corresponding 3D point $\mathbf{x}_w$ from the camera. Refer to Fig. 6.3 for illustration of the point trace. Assuming a constant illuminance over the exposure period, the light energy emitted from the point is distributed evenly (with respect to time) over the trace $\mathcal{T}$. This effectively gives us a time parametrization of a point-spread function for a given point $\mathbf{x}$, which forms the blur operator H. The space-variant PSF $h(s, t, \mathbf{X})$ corresponds precisely to the rendered trace $\mathcal{T}(\mathbf{x}, \tau)$. The point trace as defined in (6.14) is in homogenous coordinates $\mathcal{T}(\mathbf{x}, \tau) = [d(\tau)\mathbf{x}(\tau), d(\tau)]^T$. The space-variant PSF $h(s, t, \mathbf{x})$ can be thus expressed as follows:

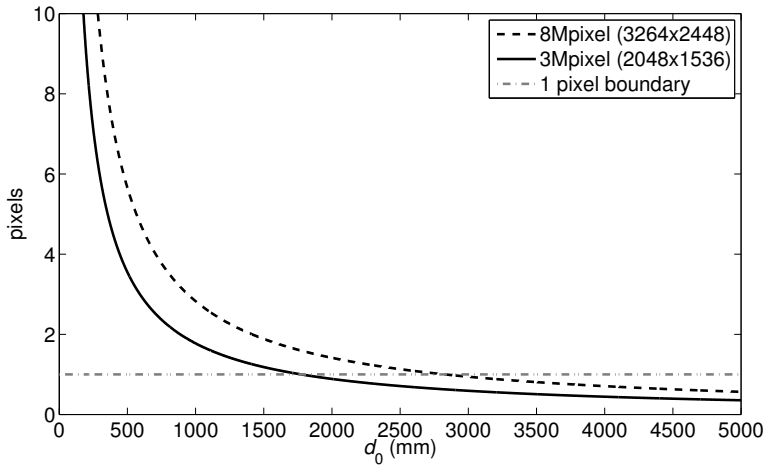


Figure 6.4 Influence of 1mm translation in x and y axes as a function of object distance from the camera with 28mm focal length (35mm equivalent). Even for relatively large sensors of 8 Mpixels, if the objects are 3 meters away from the camera, 1 mm translation in the plane perpendicular to the optical axis produces a shift of less than 1 pixel in the image plane.

$$h(s,t,\mathbf{x}) = \frac{1}{T}\int \delta\Big(\begin{bmatrix}s\\t\end{bmatrix} - \mathbf{x}(\tau)\Big)\mathrm{d}\tau\,, \tag{6.15}$$

where T is the exposure time. The point trace in (6.14) depends on camera intrinsic parameters $\mathbf{K}$ and distance d_0. Unlike the camera parameters (focal length, image center and pixel scale), which are usually known or can be easily estimated for any camera model, the distance of 3D points (depth map) are not readily available. We cannot assume that the depth map is known in practise. However, only camera translation $\mathbf{t}(\tau)$ is influenced by the distance and rotation $\mathbf{R}(\tau)$ is independent. The translation term in (6.14) is $\frac{1}{d_0}\mathbf{K}\mathbf{t}(\tau)$ and after substitution for K we see that the translation vector is multiplied by a factor $\frac{f}{d_0}$ in the x and y direction, and by a factor $\frac{1}{d_0}$ in the z direction. Typical phone cameras have focal length of several millimeters and objects that we shoot are usually at least few meters away, thus both factors are small numbers and the influence of camera translation is negligible. Fig. 6.4 illustrates influence of 1 mm camera translation in x or y direction as a function of the distance of the object from the camera. For typical phone cameras, a scene projected into the camera image plane moves by less than a pixel if the objects are more than 2 m away.

The above discussion allows us to focus on purely rotational motion of the camera and this provides us with one additional benefit. Smartphones are equipped both with accelerometers and gyroscopes. Accelerometers measure translation acceleration, which is the second derivative of $\mathbf{t}(\tau)$, and gyroscopes measure an-

gular velocity, which is proportional to the first derivative of $\mathbf{R}(\tau)$. To estimate translation vector $\mathbf{t}$ one has to perform double integration of accelerometer data along time. On the other hand, estimation of the rotation matrix $\mathbf{R}$ is done by single integration of gyroscope data along time. In double integration, accelerometer noise quickly amplifies and precision deteriorates shortly afterwards. This is not the case for the gyroscope data, which can give a relatively precise estimation of rotational angles throughout the camera exposure time. Since we assume only rotation in our camera motion model, the approximated point trace $\mathcal{T}'$ is given by

$$\mathcal{T}'(\mathbf{x},\tau): \quad \bar{\mathbf{x}}(\tau) = \mathbf{K}\mathbf{R}(\tau)\mathbf{K}^{-1}\bar{\mathbf{x}}_0\,. \tag{6.16}$$

The rotation matrix $\mathbf{R}$ is a function of three angles $\phi_x(\tau)$, $\phi_y(\tau)$ and $\phi_z(\tau)$ and these angles are estimated from gyroscope data, three angular speeds $\omega_x(\tau)$, $\omega_y(\tau)$ and $\omega_z(\tau)$, by

$$\phi_x(\tau) = \int_0^\tau \omega_x(t)\mathrm{d}t\,, \quad \phi_y(\tau) = \int_0^\tau \omega_y(t)\mathrm{d}t\,, \quad \phi_z(\tau) = \int_0^\tau \omega_z(t)\mathrm{d}t\,. \tag{6.17}$$

Knowing the camera intrinsic parameters ($\mathbf{K}$) and having the data stream from gyroscopes ($\omega_x(\tau)$, $\omega_y(\tau)$ and $\omega_z(\tau)$), we can generate PSFs $h(s,t,\mathbf{x})$ at any position $\mathbf{x}$ in the image plane using (6.15), (6.16), and (6.17).

6.5 Smartphone application

This section illustrates that blur estimation with built-in inertial sensors is possible and we developed an implementation of image deblurring on a smartphone, which works in practical situations and is relatively fast to be acceptable for a general user.

As a testing platform, we have chosen a *Samsung Galaxy S II* smartphone with Android OS. It is equipped with all the apparatus needed for our experiments; namely, a relatively high-quality camera, motion sensors, a fast CPU, and enough RAM to perform computations. A block diagram of the deblurring application is shown in Fig. 6.5.

We first discuss the space-invariant implementation, which assumes a single PSF in the entire image. Then we show a simple patch-based extension for the space-variant implementation, which applies the space-invariant method in a patch-wise manner which is still manageable on the tested smartphone platform.

6.5.1 Space-invariant implementation

During the photo acquisition, samples of angular velocity are recorded using the embedded gyroscopes, which are afterwards trimmed to fit the exposure period.

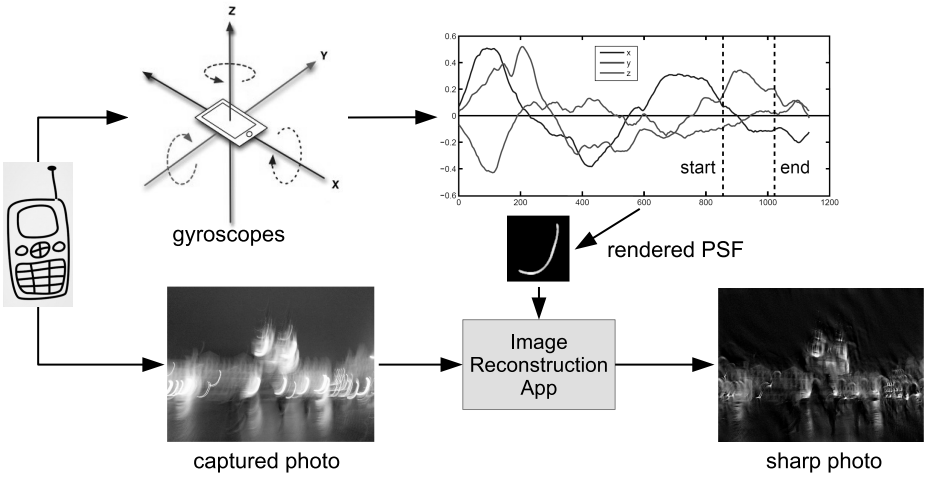


Figure 6.5 The block diagram of the smartphone application. During camera exposure, the application records data from the built-in gyroscopes. The data is processed and PSFs are estimated. The captured photo is divided into overlapping patches, Wiener deconvolution is performed on every patch and the reconstructed patches are blended to generate the sharp photo. The whole process for a 3Mpixel photo, entirely done on the smartphone, takes around 6 seconds.

An estimation of the PSF is rendered by integrating the curve position from the recorded gyroscope data using (6.15) and (6.16) for X in the center of the image.

State-of-the-art non-blind deconvolution methods use sparse image priors and the solution is usually found by some iterative minimization algorithms, such as in [8]. However the limited computational power of the smartphone prevents us from implementing these sophisticated deconvolution methods. We thus use a simple and fast *Wiener filter* in the form

$$\hat{U} = G\frac{H^*}{|H|^2 + \Phi}\,, \tag{6.18}$$

where Φ is an estimation of the inverse signal-to-noise ratio (SNR), and G, H and $\hat{U}$ are discrete Fourier transforms of the observed image g, PSF h and the estimated latent image $\hat{u}$, respectively.

Filtering in the frequency domain treats the image as a periodic function, which causes ringing artifacts around image borders. To overcome this problem, several techniques were proposed in the literature [41, 42]. We have found it sufficient to preprocess the input image g by blending the opposite image borders at the width of the PSF, which creates a smooth transition and eliminates the artifacts.

The intensity values of the output image $\hat{u}$ sometimes lie outside the 8-bit range (0-255), therefore we added optional normalization with clipping of outliers. The normalization is especially useful in the case of larger blurs and scenes with high illumination.

To convert images between the frequency and spatial domains, we use the FFT algorithm implemented in the FFTW library. Utilizing a fast ARM Cortex-A9 CPU with two cores and support for a SIMD instruction set (NEON), FFTW proved to be remarkably fast on the tested smartphone; see Tab. 6.1.

resolution	no NEON, no hardware FPU	NEON, 1 core	NEON, 2 cores
1536×1152	2900	185	110
2048×1536	5300	330	195
2050×1538	—	1000	540
3264×2448	21200	1450	800

Table 6.1 Computational time (in milliseconds) of FFT transform of gray-scale images with different sizes and different CPU settings.

Images are acquired with the native camera resolution of 3264×2448 which we then rescale to 2048×1536 to take the advantage of better performance of FFTW when the image size is a factor of small primes. Image downsampling has a negligible effect on the image quality, because native camera resolution is unnecessarily high. The optical system of the camera has a very small aperture, which, because of diffraction and optical aberrations, limits the number of pixels that can be effectively captured by the image sensor.

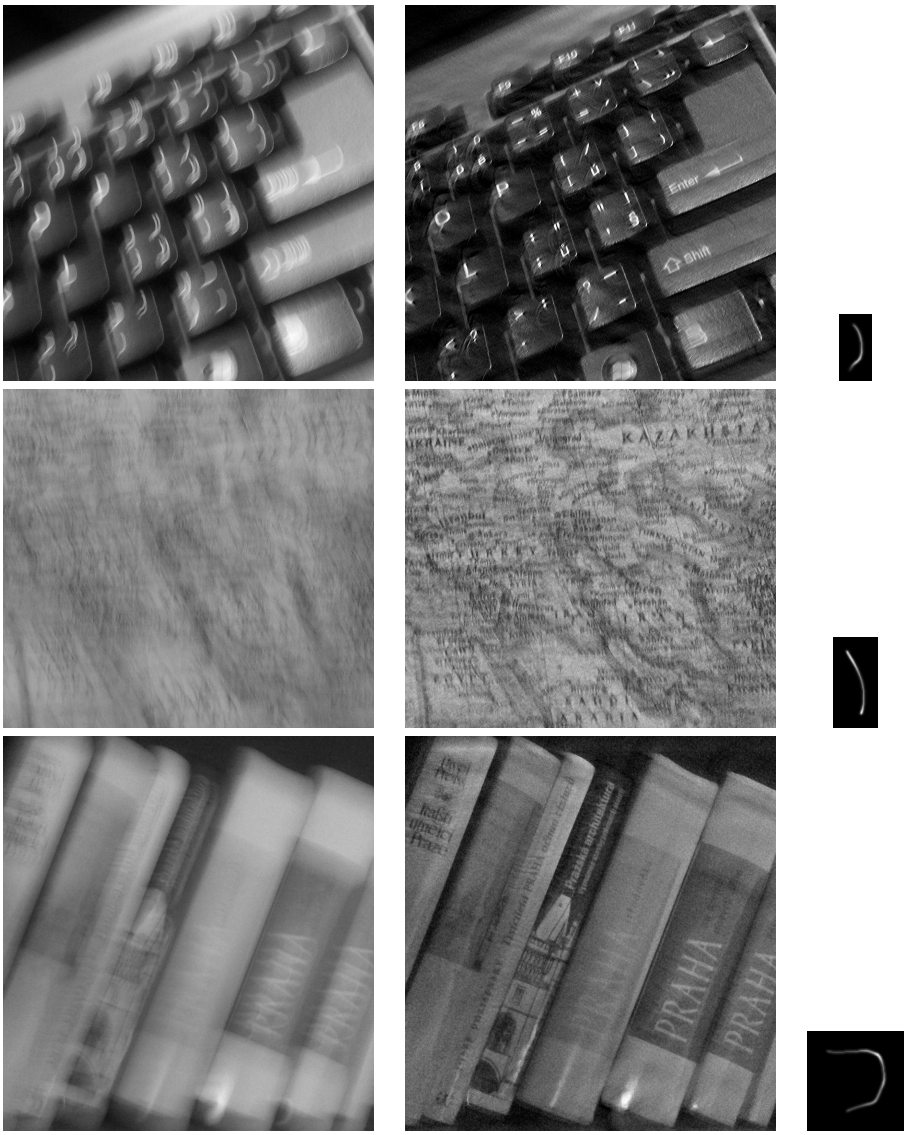
To perform Wiener filtering, the FFT must be applied several times: once for the PSF and twice (forward and backward) for each color channel. That yields a total of 7 FFT operations. With some overhead of bitmap transfers, the deconvolution phase for the image resolution 2048×1536 takes about 2.6 seconds. The whole process starting from the moment the camera shutter closes takes a little over 6 seconds. This includes image resizing, PSF estimation, compressing and saving the original and deblurred image files.

In Fig. 6.6 we display several of our results together with the PSFs calculated from the gyroscope data. All results were computed with the signal-to-noise parameter Φ set to 0.01. This value was determined experimentally to provide the best looking results. The original intention was to set Φ proportionally to ISO value extracted from EXIF data of a photo, which should determine the amount of noise present in the image. However, we found the dependency of Φ on ISO negligible. This behavior can be explained by the denoising step performed internally by the camera module.

6.5.2 Space-variant implementation

Performance of the space-invariant implementation (as seen in Fig. 6.6) is by no means consistent. Deconvolved images are impaired by visual anomalies worsening its appearance. Most often these anomalies manifest as ringing artifacts surrounding sharp edges in the picture, as demonstrated in Fig. 6.7.

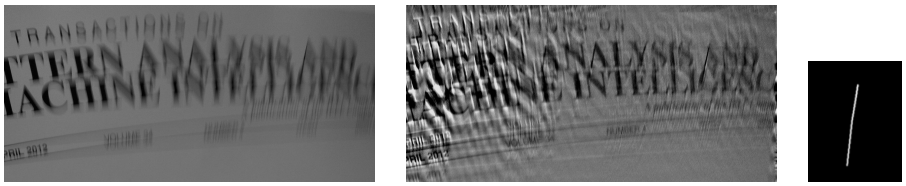
The main cause is the space-variant nature of the blur that was not considered so far. The space-variant PSFs are particularly noticeable when rotation around



(a) captured image (b) reconstructed image (c) PSF

Figure 6.6 Examples of space-invariant deconvolution on the smartphone.

the z axis is significant. Smartphones are equipped with wide-angle lenses with a field of view of approximately 60°. This means that camera projection causes rotation around the x and y axes to produce strongly space-variant blur in parts close to image borders. Using relationships in (6.15) and (6.16) allows us to render a correct PSF at every pixel of the image. However to perform image deblurring with space-variant PSFs would be computationally expensive since we cannot use the Wiener filter. Instead we break the image into overlapping



(a) captured image (b) reconstructed image (c) PSF

Figure 6.7 An example of reconstruction artifacts in the space-invariant implementation due to space-variant nature of degradation.

patches and generate one PSF in every patch using (6.16) at the center of the patch. An example of dividing the image into 6×8 patches with 25% overlap and generating the corresponding PSFs is in Fig. 6.9. In every patch we apply the Wiener filter (including edge tapering and normalization) as described in the space-invariant implementation in Sec. 6.5.1. Due to patch overlaps, we blend the reconstructed patches by weighting them with appropriate Hamming windows which produces the final image without noticeable seams between patches; see Fig. 6.12(d).

Another reason for space-variant blur, which is not connected with camera motion, but inherent to the camera hardware design, is the shutter mechanism. Contrary to systems with a mechanical shutter, CMOS sensors in cheep cameras are continuously illuminated. The readout from the CMOS sensor takes several tens of milliseconds, which results in a picture not taken at a single moment, but with a slight time delay between the first and last pixel row. This process, called *rolling shutter*, is therefore another cause of the blur variance as the PSF depends on the vertical position in the image. The correct approach to PSF estimation is thus shifting the inertial sensor data in time according to the vertical position in the image. An example in Fig. 6.8 illustrates the rolling shutter effect. We took a snapshot of a LCD screen displaying a grid of white points on black background. Due to camera motion, the points appear as streaks on the captured image. To accurately model the PSFs at every position, it is necessary to shift the exposure-time window, in which the gyroscope data are fetched. Let us assume that we know the precise moment when the camera exposure starts. At the top row of the image there is no delay between the exposure start and the time when we start reading gyroscope data. As we move down towards the bottom rows, the delay increases as these rows are exposed later. As illustrated in Fig. 6.8 for the particular testing device, the delay between reading the first row and the last row of the image is around 60ms.

6.6 Evaluation

Removal of camera shake based on gyroscope data as proposed in the previous section should perform better than fully blind deconvolution methods discussed

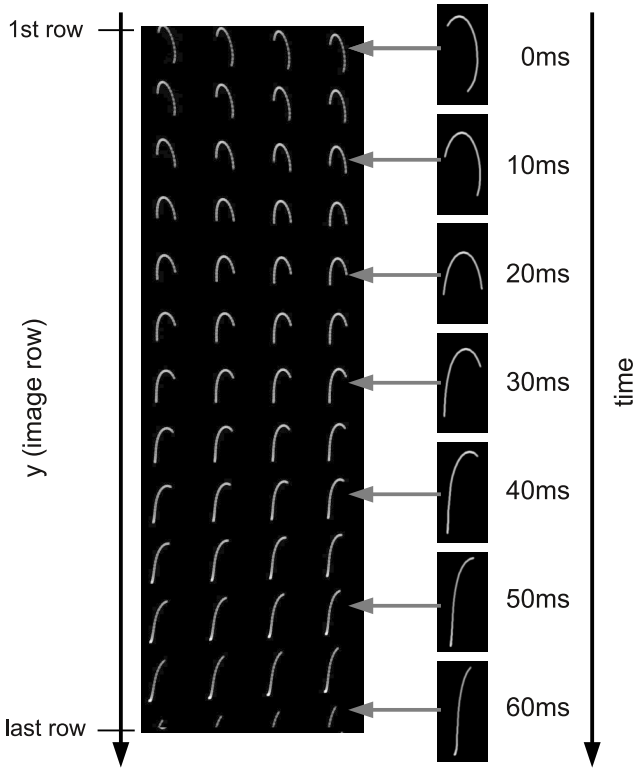


Figure 6.8 A snapshot (exposure time 1/14s) of point grid displayed on a LCD screen showing the rolling shutter effect. The middle column shows a series of blur kernels rendered using data from the gyroscope sensor shifted in time. Blurs were created from sensor data starting 0–60 ms after a synchronization timestamp.

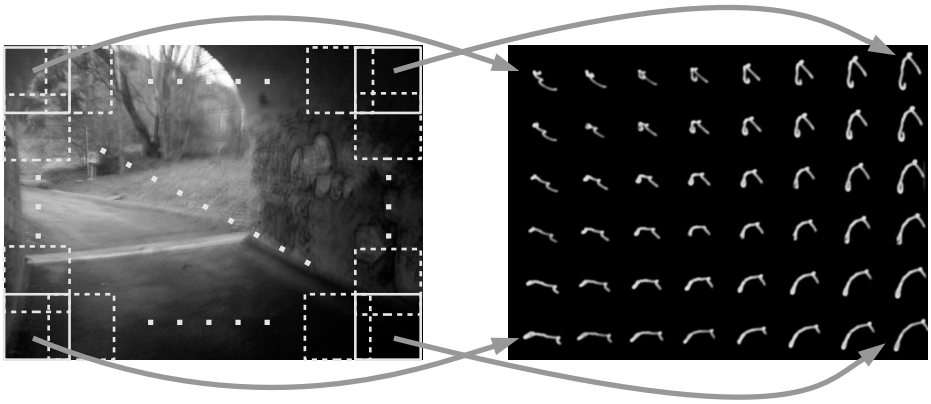
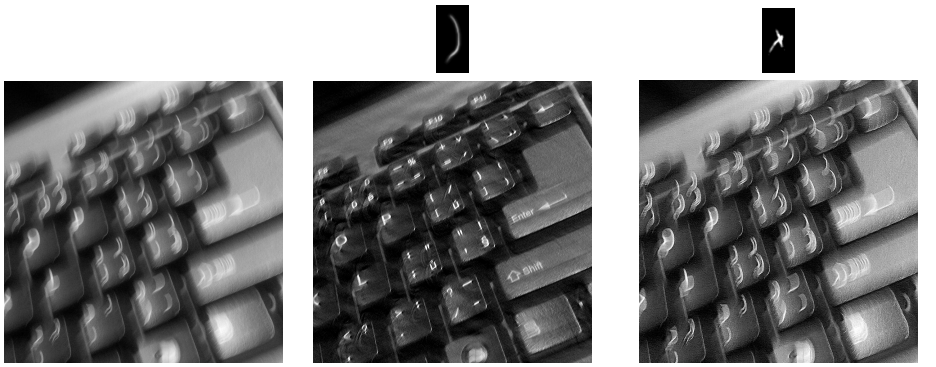


Figure 6.9 Patch-based space-variant reconstruction: The input image is divided into 6×8 patches with 25% overlap. In the center of every patch, we estimate the corresponding PSF using data from smartphone gyroscopes.



(a) captured image (b) recon. with gyroscope (c) recon. with blind method

Figure 6.10 Comparison of reconstruction methods: (a) input blurred image; (b) estimated sharp image and PSF with our smartphone application using gyroscope data; (c) state-of-the-art blind deconvolution method proposed by Xu *et al.* in [13].

in Sec. 6.3.1. To illustrate algorithm's performance, we compare the outcome of the proposed smartphone application in Fig. 6.10(b) with the state-of-the-art single-channel blind deconvolution in Fig. 6.10(c) proposed by Xu *et al.* in [13]. Xu's method apart from being far more computationally demanding, is unable to estimate the correct blurs in most of the test images we have taken with the smartphone.

The angular speed estimated by gyroscopes is obviously not absolutely precise. To quantify the effect of gyroscope additive noise, we have conducted the following synthetic experiment. A test image was synthetically convolved with a PSF that was rendered using some gyroscope samples recorded previously with our mobile application and an additive noise (SNR = 40 dB) was added to the image. The generated blurred image imitated a photo that one typically acquires with the smartphone. We then tried to deblur the image using the same gyroscope samples but this time corrupted by additive noise to simulate errors in sensor measurement. PSNR $= 10\log_{10}(|\Omega|255^2/\|u-\hat{u}\|^2)$(dB) of the results as a function of noise standard deviation (rad/s) in gyroscope samples is summarized in Fig. 6.11. The graph clearly shows that the performance of deblurring quickly deteriorates as the noise level increases. Note that a fall of 1 dB in PSNR has a dramatic visual effect on the image quality. The gyroscope noise level typically encountered in motion sensors inside mobile devices (in our case Samsung Galaxy S II) is 0.007 rad/s for our sampling rate, which is depicted by the vertical dashed line in the graph. We can see that this noise level is associated with a relatively small drop in PSNR, which is one of the reasons for satisfactory performance of the proposed smartphone application.

In Fig. 6.12 we demonstrate the performance of the space-variant implementation from Sec. 6.5.2. The smartphone implementation divides the acquired image in Fig. 6.12(a) into 6×8 patches of size 384×384 pixels with 25% overlap. PSFs

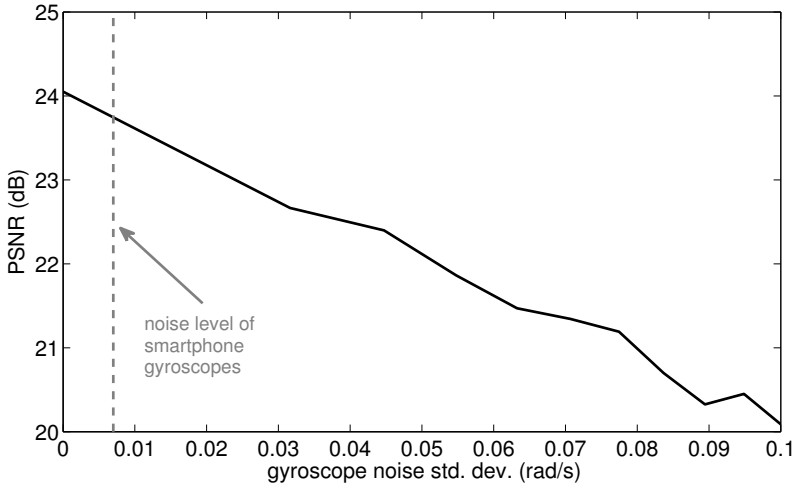


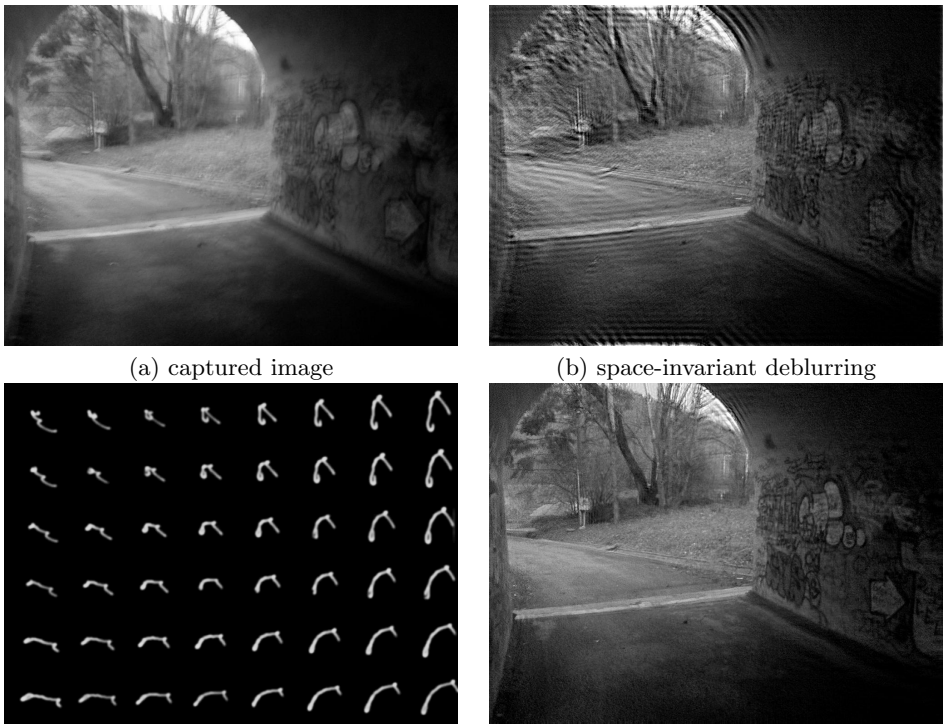
Figure 6.11 PSNR of the original and deblurred image as a function of additive gyroscope noise: Gaussian noise of standard deviation 0 to 0.1 rad/s was added to gyroscope samples (angular velocity in rad/s). The deconvolution algorithm was then applied using computed blur kernels based on these altered gyroscope measurements. The graph shows mean values of 50 realizations for each of the noise levels.

generated from gyroscope data in every patch are illustrated in Fig. 6.12(c). The reconstruction of the sharp image (Fig. 6.12(d)) takes about 7 seconds on the testing device. For comparison we also show the space-invariant implementation in Fig. 6.12(b), which assumes only one PSF generated for the center of the image. In this particular case, the space-variant nature of PSFs was profound and therefore the image shows noticeable ringing artifacts.

6.7 Conclusions

The first half of this chapter gives an overview of current trends in blind deconvolution methodology using Bayesian inference. Image deblurring is inherently an ill-posed problem and we need to search for techniques that make it better posed. In general, there are two ways to achieve this. We must include priors, which penalize inappropriate images and incorrect blur kernels, or we use additional information from other sources, such as multiple images of the same scene, parametric models or information from inertial sensors. Our main attention is blur caused by camera shake, which is very common in real life and even though space-variant it can be easily parametrized.

The second half of the chapter presents an image deblurring method that can effectively remove blur caused by camera motion using information from inertial sensors – gyroscopes. The proposed method is fully implemented on a smart-



(a) captured image (b) space-invariant deblurring

(c) gyroscope generated blurs (d) space-variant deblurring

Figure 6.12 An example of space-variant reconstruction: Due to space-variant nature of the blur, which is clearly visible in (c), the space-invariant deblurring in (b) produces reconstruction artifacts. On the other hand, performing deconvolution patch-wise with correct blurs gives satisfactory results.

phone device, which is to our knowledge the first attempt in this direction and renders the method particularly appealing for end users. We have demonstrated that the space-invariant simplification for certain camera motions is plausible, but simultaneously we have uncovered intrinsic sources of space-variant blur. The space-variant implementation of the deblurring algorithm solves the complex camera motion and rolling shutter issues, which both can be modeled by space-variant blur.

There are several topics of future research. Implementing smarter deblurring algorithms that avoid ringing artifacts is viable and clearly a tempting improvement. Gyroscope data are not precise and one can use the calculated PSFs as an initial estimate and apply modified blind deconvolution methods to improve their accuracy.

Acknowledgment

We would like to thank Ondřej Šindelář for implementing the smartphone application and generating results for this chapter. This work was supported in part by the Grant Agency of the Czech Republic under the project 13-29225S and by the Academy of Sciences of the Czech Republic under the project M100751201.

References

[1] Campisi P, Egiazarian K, editors. Blind Image Deconvolution, Theory and Application. CRC Press; 2007.

[2] Rosenfeld A, Kak AC. Digital Picture Processing. 2nd ed. Orlando, FL, USA: Academic Press, Inc.; 1982.

[3] Rudin LI, Osher S, Fatemi E. Nonlinear total variation based noise removal algorithms. Physica D. 1992;60:259–268.

[4] Sroubek F, Flusser J. Multichannel blind deconvolution of spatially misaligned images. IEEE Transactions on Image Processing. 2005 Jul;14(7):874–883.

[5] Molina R, Mateos J, Katsaggelos AK. Blind Deconvolution Using a Variational Approach to Parameter, Image, and Blur Estimation. IEEE Transactions on Image Processing. 2006 Dec;15(12):3715–3727.

[6] Fergus R, Singh B, Hertzmann A, Roweis ST, Freeman WT. Removing camera shake from a single photograph. In: SIGGRAPH '06: ACM SIGGRAPH 2006 Papers. New York, NY, USA: ACM; 2006. p. 787–794.

[7] Jia J. Single Image Motion Deblurring Using Transparency. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition CVPR '07; 2007. p. 1–8.

[8] Shan Q, Jia J, Agarwala A. High-quality motion deblurring from a single image. In: SIGGRAPH '08: ACM SIGGRAPH 2008 papers. New York, NY, USA: ACM; 2008. p. 1–10.

[9] Levin A, Weiss Y, Durand F, Freeman WT. Understanding and evaluating blind deconvolution algorithms. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition CVPR '09; 2009. p. 1964–1971.

[10] Levin A, Weiss Y, Durand F, Freeman WT. Understanding Blind Deconvolution Algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2011;33(12):2354–2367.

[11] Miskin J, MacKay DJC. Ensemble Learning for Blind Image Separation and Deconvolution. In: Girolani M, editor. Advances in Independent Component Analysis. Springer-Verlag; 2000. p. 123–142.

[12] Galatsanos NP, Mesarovic VZ, Molina R, Katsaggelos AK. Hierarchical Bayesian image restoration from partially known blurs. IEEE Transactions on Image Processing. 2000;9(10):1784–1797.

[13] Xu L, Jia J. Two-phase kernel estimation for robust motion deblurring. In: Proceedings of the 11th European conference on Computer vision: Part I. ECCV'10. Berlin, Heidelberg: Springer-Verlag; 2010. p. 157–170.

[14] Schulz TJ. Multiframe Blind Deconvolution of Astronomical Images. J Opt Soc Am A. 1993 May;10(5):1064–1073.

[15] Harikumar G, Bresler Y. Perfect Blind Restoration of Images Blurred by Multiple Filters: Theory and Efficient Algorithms. IEEE Trans Image Processing. 1999 Feb;8(2):202–219.

[16] Gurelli MI, Nikias CL. EVAM: An eigenvector based algorithm for multichannel blind deconvolution of input colored signals. IEEE Trans Acous, Speech, Signal Processing. 1995;43:134–149.

[17] Giannakis GB, Heath RW. Blind identification of Multichannel FIR Blurs and Perfect Image Restoration. IEEE Trans Image Processing. 2000 Nov;9(11):1877–1896.

[18] Pai HT, Bovik AC. On eigenstructure-based direct multichannel blind image restoration. IEEE Trans Image Processing. 2001 Oct;10(10):1434–1446.

[19] Pillai SU, Liang B. Blind Image Deconvolution Using a Robust GCD Approach. IEEE Trans Image Processing. 1999 Feb;8(2):295–301.

[20] Haindl M, Šimberová S. Model-Based Restoration of Short-Exposure Solar Images. In: Jain LC, Howlett RJ, editors. Frontiers in Artificial Intelligence and Applications. vol. 87 of Knowledge-Based Intelligent Engeneering Systems. Amsterdam: Publisher IOS Press; 2002. p. 697–706.

[21] Panci G, Campisi P, Colonnese S, Scarano G. Multichannel Blind Image Deconvolution Using the Bussgang algorithm: Spatial and Multiresolution Approaches. IEEE Trans Image Processing. 2003 Nov;12(11):1324–1337.

[22] Šroubek F, Flusser J. Multichannel Blind Iterative Image Restoration. IEEE Trans Image Processing. 2003 Sep;12(9):1094–1106.

[23] Šroubek F, Cristóbal G, Flusser J. A unified approach to superresolution and multichannel blind deconvolution. IEEE Trans Image Processing. 2007 Sep;16(9):2322–2332.

[24] Tico M, Trimeche M, Vehvilainen M. Motion Blur Identification Based on Differently Exposed Images. In: Proc. IEEE International Conference on Image Processing; 2006. p. 2021–2024.

[25] Yuan L, Sun J, Quan L, Shum HY. Image deblurring with blurred/noisy image pairs. In: SIGGRAPH '07: ACM SIGGRAPH 2007 papers. New York, NY, USA: ACM; 2007. p. 1.

[26] Šorel M, Šroubek F. Space-variant deblurring using one blurred and one underexposed image. In: Proceedings of the IEEE 16th International Conference on Image Processing ICIP 2009. IEEE; 2009. p. 157–160.

[27] Šorel M, Flusser J. Space-Variant Restoration of Images Degraded by Camera Motion Blur. IEEE Transactions on Image Processing. 2008 Feb;17(2):105–116.

[28] Šorel M, Šroubek F, Flusser J. Towards super-resolution in the presence of spatially varying blur. In: Peyman M, editor. Super-resolution imaging. CRC Press; 2010. p. 187–217.

[29] Favaro P, Burger M, Soatto S. Scene and Motion Reconstruction from Defocus and Motion-Blurred Images via Anisothropic Diffusion. In: Pajdla T, Matas J, editors. ECCV 2004, LNCS 3021, Springer Verlag, Berlin Heidelberg; 2004. p. 257–269.

[30] Joshi N, Kang SB, Zitnick CL, Szeliski R. Image deblurring using inertial measurement sensors. ACM Trans Graph. 2010 July;29:30:1–30:9.

[31] Ben-Ezra M, Nayar SK. Motion-Based Motion Deblurring. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2004 Jun;26(6):689–698.

[32] Tai YW, Du H, Brown MS, Lin S. Correction of Spatially Varying Image and Video Motion Blur Using a Hybrid Camera. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2010;32:1012–1028.

[33] Hirsch M, Schuler CJ, Harmeling S, Scholkopf B. Fast removal of non-uniform camera shake. In: Proc. IEEE Int Computer Vision (ICCV) Conf; 2011. p. 463–470.

[34] Whyte O, Sivic J, Zisserman A, Ponce J. Non-uniform deblurring for shaken images. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on; 2010. p. 491 –498.

[35] Gupta A, Joshi N, Zitnick CL, Cohen M, Curless B. Single image deblurring using motion density functions. In: Proceedings of the 11th European conference on Computer vision: Part I. ECCV'10. Berlin, Heidelberg: Springer-Verlag; 2010. p. 171–184.

[36] Stockham TG 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: ACM; 1966. p. 229–233.

[37] Nagy JG, O'Leary DP. Restoring Images Degraded by Spatially Variant Blur. SIAM JOURNAL ON SCIENTIFIC COMPUTING. 1998;19(4):1063–1082.

[38] Cho S, Lee S. Fast Motion Deblurring. ACM Transactions on Graphics (SIGGRAPH ASIA 2009). 2009;28(5):article no. 145.

[39] Harmeling S, Michael H, Scholkopf B. Space-Variant Single-Image Blind Deconvolution for Removing Camera Shake. In: Lafferty J, Williams CKI, Shawe-Taylor J, Zemel RS, Culotta A, editors. Advances in Neural Information Processing Systems 23; 2010. p. 829–837.

[40] Šroubek F, Šorel M, Horackova I, Flusser J. Patch-based Blind Deconvolution with Parametric Interpolation of Convolution Kernels. In: Proc. IEEE International Conference on Image Processing; 2013. p. 1–4.

[41] Liu R, Jia J. Reducing boundary artifacts in image deconvolution. In: Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on; 2008. p. 505 –508.

[42] Šorel M. Removing Boundary Artifacts for Real-Time Iterated Shrinkage Deconvolution. IEEE Transactions on Image Processing. 2012 Apr;21(4):2329 –2334.