

# Simultaneous recovery of scene structure and blind restoration of defocused images

Michal Šorel and Jan Flusser

Institute of Information Theory and Automation  
Academy of Sciences of the Czech Republic  
Pod vodárenskou věží 4, 182 08 Prague 8, Czech Republic

**Abstract** We present an algorithm that uses two or more defocused images of the same scene for recovery of scene structure and simultaneous restoration of sharp image.

Defocusing is modeled by convolution with arbitrary known spatially-variant mask unlike the vast majority of published algorithms that used fixed cylindrical or Gaussian mask shapes. In this way it is able to deal with aberrations present in real optical systems. The mask can be given analytically or by a table obtained from physical measurements or generated by a ray tracing algorithm.

For simplicity, we apply additional constraint that points of the same depth produce the same mask regardless of their position in the field of view. This assumption can be nevertheless relaxed at the cost of higher time or memory requirements.

Algorithm can be easily parallelized and has a potential to be used in real time applications.

## 1 Introduction

Defocusing, as well as many other types of simple degradations, can be described by linear relation

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

where  $\mathbf{u}$  is a sharp image,  $\mathbf{h}$  is called *point-spread function* (PSF) or *mask* and  $\mathbf{z}$  is the blurred image. For real optical systems, the PSF depends on the distance of object projecting to the point  $(x, y)$  as well as camera parameters and coordinates  $(x, y)$  themselves. If we assume simple Gaussian optics model and circular aperture, the PSF is a cylinder with radius directly proportional to the reciprocal of the object distance and we speak about *blur circle* or *circle of confusion*. In many cases, the PSF can be better approximated by two-dimensional Gaussian function with variance again related to the object distance.

*Depth from defocus* (DFD) can be defined as the task to recover the distance of image points from the camera when we know  $\mathbf{z}$  and the relation between the PSF  $\mathbf{h}$  and the distance. The opposite problem to find the sharp image  $\mathbf{u}$  when we know  $\mathbf{z}$  and possibly  $\mathbf{h}$  is called *deconvolution* or *image restoration*. If even the PSF is not known, we speak about *blind* deconvolution. Since both problems are mostly too

difficult to solve from just one single image, it is often assumed that we have at least two observations of the same scene taken with different camera settings and we speak about *multichannel* (MC) deconvolution.

Now, we give a short overview of relevant literature for both DFD and deconvolution problems.

Early DFD methods such as [8, 4] are based on the idea of sliding-window, meaning that the amount of defocus is assumed to be constant over some fixed rectangular neighborhood of the given point. Among them, we point out the method of Subbarao and Surya [12], who assumed the Gaussian mask shape, approximated image function by 3rd-order polynomial and derived an elegant expression for relative blur

$$\sigma_2^2 - \sigma_1^2 = 4 \frac{\mathbf{z}_2 - \mathbf{z}_1}{\nabla^2 \left( \frac{\mathbf{z}_1 + \mathbf{z}_2}{2} \right)}, \quad (2)$$

which can be used to estimate distance. Here  $\mathbf{z}_1, \mathbf{z}_2$  are defocused images,  $\sigma_1^2$  and  $\sigma_2^2$  denote variances of mask shapes taken as probability distributions of two-dimensional random quantities and  $\nabla^2$  denotes Laplacian operator. We anticipate that in our algorithm we use this extremely fast method as a reliable initial estimate of the scene structure.

Later, a number of filter-based DFD methods were proposed to achieve better precision [14, 13] or to incorporate for instance image registration [3].

Now we move our attention to restoration methods. The restoration from a single image degraded by known shift-invariant blur can be solved by a multitude of shift-invariant *single channel* (SC) deconvolution techniques [1]. Many of them are formulated as quadratic minimization problems, some others including important anisotropic regularization techniques [10, 2] can be reduced to a sequence of quadratic problems. The vast majority of these techniques can be used in shift-variant situations as well.

Blind restoration requires more complicated algorithms as we need to estimate the unknown degradation. In connection with our algorithm we are interested in shift-variant case, when the PSF can change from point to point.

Just few general results have been reported on this subject, all of them of very limited application. They mostly follow the idea of sliding-window [7, 15].

In the case of optical imaging we know much more about the image formation and the number of unknowns can be strongly reduced, especially if we assume that PSF is a

(single-valued) function of distance. Indeed, an obvious approach to blind restoration is to apply non-blind restoration on the result of a DFD method.

An alternative approach is to do both, DFD and restoration, simultaneously. Rajagopalan and Chaudhuri [9] proposed a MRF-based approach, which is equivalent to minimization of discrepancy between the physical model and measurements assuming two images and Gaussian mask shape. To minimize the corresponding cost functional they used simulated annealing which has a nice property of global convergence, but is too slow to be used in practice. Another view of the same minimization problem was given by Favaro et al. [5] who modeled defocusation as anisotropic diffusion process and solved the corresponding PDE. In order to bypass the deblurring phase of minimization, Favaro and Soatto [6] derived projection operators that yield directly the minimum value of cost functional for given depth map.

Basically, we followed the same approach as the methods from the previous paragraph [9, 5, 6] and focused on the design of a system which could be used in practice. As the PSF of a real lens system significantly differs from Gaussian or cylindrical shapes, we allowed for arbitrary PSF shape. Moreover, our optimization procedure works efficiently also in situations when PSF is not given analytically but for instance by a table. This can be useful as it is easy to get the PSF of a particular lens system by a raytracing algorithm and difficult to express it explicitly by an equation. The whole algorithm consists of a sequence of “shift-variant convolutions” with the consequence that if we have a hardware that is able to carry out this operation efficiently, the whole algorithm can be accelerated and possibly achieve real time performance.

## 2 Optics

As we mentioned in the introduction, in Gaussian optics model if the aperture is assumed to be circular, the PSF has a cylindrical shape with radius  $r$  being the function of point distance  $l$ , namely

$$r = \rho\zeta \left( \frac{1}{\zeta} + \frac{1}{l} - \frac{1}{f} \right) = \frac{1}{l}\rho\zeta + \rho\zeta \left( \frac{1}{\zeta} - \frac{1}{f} \right), \quad (3)$$

where  $f$  is the focal length,  $\rho$  the aperture radius and  $\zeta$  is the distance of the image plane from the lens. This formula can be simply derived from the similarity of triangles.

In the rest of this paper function  $\mathbf{r}(x, y)$  denotes the radius  $r$  corresponding to the distance of point  $(x, y)$  according to (3). We refer to this function as *blur map* analogously to a similar quantity that maps the real distance and is mostly called *depth map* in literature. If we allow for negative  $r$ , (3) gives a one-to-one correspondence between  $r$  and  $l$  values with the exception of  $l = 0$  case. Thus, we can use  $\mathbf{r}(x, y)$  to represent the scene structure instead of the actual distance.

Now, suppose we have another image of the same scene taken with different camera settings. As the distance is the same for all pairs of corresponding points, we get

$$\mathbf{r}_2(x, y) = \alpha\mathbf{r}_1(x, y) + \beta, \quad \alpha, \beta \in \mathbf{R}, \quad (4)$$

where  $\alpha$  and  $\beta$  can be trivially computed from (3). The proposed algorithm assumes  $\alpha$  and  $\beta$  are known.

## 3 Algorithm

To simplify notation, we generalize the concept of convolution to cover the shift-variant case and define two types of *shift-variant convolution* as follows:

$$\begin{aligned} \mathbf{u} *_{d} \mathbf{h} [x, y] &= \int_{\Omega} \mathbf{u}(x-s, y-t) \mathbf{h}(x-s, y-t; s, t) ds dt, \\ \mathbf{u} *_{g} \mathbf{h} [x, y] &= \int_{\Omega} \mathbf{u}(x-s, y-t) \mathbf{h}(x, y; -s, -t) ds dt. \end{aligned} \quad (6)$$

Obviously, relation (1) can be rewritten as  $\mathbf{z} = \mathbf{u} *_{d} \mathbf{h}$ . Note that in the shift-invariant case, when  $\mathbf{h}(x, y; s, t) = \mathbf{h}(s, t)$ , it is exactly the standard convolution.

As we mentioned in the previous section, if we know the actual camera settings, the scene structure can be represented by blur map  $\mathbf{r}_1(x, y)$  related to the first image, since according to (4) blur maps  $\mathbf{r}_p$  related to the other channels are a known function of  $\mathbf{r}_1$ . An advantage of this representation is that we do not have to know actual camera parameters to carry out restoration. If the images differ just in aperture setting,  $\beta = 0$  and the relation between channels is given by just one number  $\alpha$ . Moreover, the MSE in  $\mathbf{r}$  is a lens independent measure of achieved precision. Actually, the proposed algorithm does not work with the real object distance at all and if we are interested, we can get it from the inverse of relation (3).

Consequently, the assumption that mask is given only by the corresponding distance, can be restated as that it is a function of  $\mathbf{r}_1$ . The mask in question will be denoted as  $h_p(\mathbf{r}_p(x, y))$ .

Now, we take advantage of the notation introduced in the previous section and define

$$\mathbf{e}_p = \mathbf{u} *_{d} h_p(\mathbf{r}_p) - \mathbf{z}_p, \quad (7)$$

which is essentially nothing else than a matrix of error at individual points of the image  $p$ .

The proposed algorithm can be described as a minimization of functional (8) with respect to image  $\mathbf{u}$  and blur map  $\mathbf{r}_1$ .

$$E(\mathbf{u}, \mathbf{r}_1) = \frac{1}{2} \sum_{p=1}^P \|\mathbf{e}_p\|^2 + \lambda_u Q(\mathbf{u}) + \lambda_r R(\mathbf{r}_1), \quad (8)$$

where according to (4)

$$\mathbf{r}_p = \alpha_p \mathbf{r}_1 + \beta_p \text{ for images } p > 1. \quad (9)$$

The first *residual* term, is a measure of difference between the model, given by an ideal image  $\mathbf{u}$  and blur map  $\mathbf{r}_1$  (which represents scene structure), and measurements given by blurred images  $\mathbf{z}_p$ . The residual can be written as  $\Phi = \sum_{p=1}^P \Phi_p$ , where  $\Phi_p = \frac{1}{2} \|\mathbf{e}_p\|^2 = \frac{1}{2} \int_{\mathcal{D}} \mathbf{e}_p^2(x, y)$ .

$R(\mathbf{r}_1)$  is a blur map regularization term which can be chosen to represent properly the expected scene structure. A quadratic term such as  $R_2(\mathbf{r}) = \int \|\nabla \mathbf{r}\|^2$  behaves well for smooth surfaces, while total variation  $R_{TV}(r) = \int \|\nabla \mathbf{r}\|$ ,

proposed by Rudin et al. [10], is more appropriate for scenes containing abrupt changes in depth. The same applies to the image regularization term  $Q(\mathbf{u})$  and its relation to the expected image function.

To minimize (8) we will need its gradient, which obviously equals the sum of the gradients of individual terms. First, the gradients of regularization terms are

$$\frac{\partial R_2}{\partial \mathbf{r}} = -\operatorname{div} \nabla \mathbf{r} = -\nabla^2 \mathbf{r}, \quad (10)$$

$$\frac{\partial R_{TV}}{\partial \mathbf{r}} = -\operatorname{div} \left( \frac{\nabla \mathbf{r}}{\|\nabla \mathbf{r}\|} \right), \quad (11)$$

where the symbol  $\nabla^2$  denotes Laplacian operator and  $\operatorname{div}$  is the divergence operator. The gradient of  $Q(\mathbf{u})$  we get from (10) and (11) by simply replacing  $\mathbf{r}$  with  $\mathbf{u}$ .

The gradient of the residual term  $\Phi$  can be expressed as

$$\frac{\partial \Phi}{\partial \mathbf{u}} = \sum_{p=1}^P \mathbf{e}_p *_{\mathbf{g}} h_p(\mathbf{r}_p), \quad (12)$$

$$\frac{\partial \Phi}{\partial \mathbf{r}_1} = \mathbf{u} * \sum_{p=1}^P \alpha_p \left[ \mathbf{e}_p *_{\mathbf{g}} \frac{\partial h_p(\mathbf{r}_p)}{\partial \mathbf{r}_p} \right], \quad (13)$$

where  $\frac{\partial h_p(\mathbf{r}_p)}{\partial \mathbf{r}_p}[x, y; s, t]$  is the derivative of the mask related to image point  $(x, y)$  with respect to the value of  $\mathbf{r}_p(x, y)$ . The  $*_{\mathbf{g}}$  operator, borrowed from Matlab, denotes simple point to point multiplication of functions. The proof to appear in [11].

The minimization of the cost functional  $E$  is a highly nonlinear problem, especially in the subspace corresponding to variable  $\mathbf{r}_1$ , and as a consequence the right choice of initial state is essential to prevent the algorithm from trapping in a local minimum. As the initial blur map estimate we used already mentioned DFD method [12], which can be described by simple expression (2). It provides noisy and inaccurate depth estimates, especially if the actual PSF differs significantly from a Gaussian function but helps to prevent the algorithm from getting stuck in a local minimum and speeds up the minimization considerably.

For subsequent minimization we use a sort of alternating minimization (AM) algorithm, which basically iterates through minimizations in subspaces corresponding to unknown matrices  $\mathbf{u}$  and  $\mathbf{r}_1$ .

Minimization of  $E$  with respect to  $\mathbf{u}$  is the well examined problem of non-blind deconvolution [1]. In the  $Q_2$  case, the whole problem is quadratic and we use simple and relatively fast conjugate gradients method. In case of anisotropic  $Q_{TV}$  we have chosen the algorithm [2] reducing the problem to a sequence of quadratic problems. The idea is as follows.

Let  $\mathbf{u}_n$  be the current estimate of the image minimizing the cost functional (8). We will replace the term  $Q = Q_{TV} = \int \|\nabla(\mathbf{u})\|$  by quadratic term

$$\frac{1}{2} \int_{\mathcal{D}} \frac{1}{\|\nabla \mathbf{u}_n\|} \|\nabla \mathbf{u}\|^2 + \|\nabla \mathbf{u}_n\|. \quad (14)$$

Obviously, it has the same value as  $Q_{TV}$  in  $\mathbf{u}_n$  and it can be shown that it has the same gradient as well. The right term of (14) is constant for now and consequently it does not take

part in actual minimization. We have got a “close” quadratic problem

$$\mathbf{u}_{n+1} = \arg \min_{\mathbf{u}} \frac{1}{2} \sum_{p=1}^P \|\mathbf{e}_p\|^2 + \lambda_u \int_{\mathcal{D}} \frac{1}{2\|\nabla \mathbf{u}_n\|} \|\nabla \mathbf{u}\|^2, \quad (15)$$

solution of which becomes a new estimate  $\mathbf{u}_{n+1}$ . For numerical reasons we take  $\max(\varepsilon, \|\nabla \mathbf{u}_n\|)$  in place of  $\|\nabla \mathbf{u}_n\|$  in (15). The minimization is not very sensitive to the choice of  $\varepsilon$  and for typical images with values in the interval  $(0, 1)$  can be set to something between 0.001 and 0.01. The proof of convergence can be found in [2].

In the subspace corresponding to unknown blur map  $\mathbf{r}_1$  we use the simple steepest descent algorithm.

To specify the number and order of iterations, we use the following notation. For example iteration scheme  $50 \times (8 + 10)$  means that in one step of the algorithm we carry out 8 iterations of CG method over  $\mathbf{u}$ , then 10 iterations of steepest descent over  $\mathbf{r}_1$ , and the whole process is repeated 50 times. The particular number of iterations 8 and 10 seems to be a good choice for wide range of images. We should stress, that CG method in the image subspace is crucial for success of the minimization. At the end, the restored image can be even sharpened by additional say 100 iterations of CG minimization over the image subspace and, for the above example, we describe the whole minimization as  $50 \times (8 + 10) + 100$ .

## 4 Experiments

To demonstrate the performance of the proposed algorithm we present a set of three simulated experiments.

First, let us look at the figure of historical map Fig. 1(a) used as the original image for the experiments. It contains areas of very complex texture but we can also find places of almost constant image function. Since proposed algorithm behaves locally in the sense that the solution depends mainly on points in close neighborhood of the given point (one step of minimization depends only on the neighborhood of size corresponding to blur mask support), it enables us to get an accurate picture of behavior of the algorithm on different types of scenes.

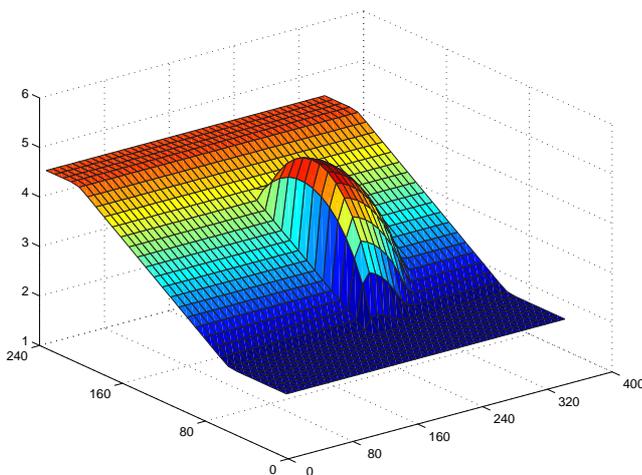
To represent the scene structure we used depth map Fig. 1(b). Again, the scene was designed to show behavior of the algorithms on various types of surfaces - there are areas of constant depth (lower and upper parts of the image), slanted plane, steep edge and spherical surface.

To simulate how the PSF changes with the distance of corresponding objects, we assumed that it keeps its shape and just stretches to have the same support it would have if it was the cylinder of radius (3). It enabled us to generate masks of arbitrary size from one prototype Fig. 1(c) as

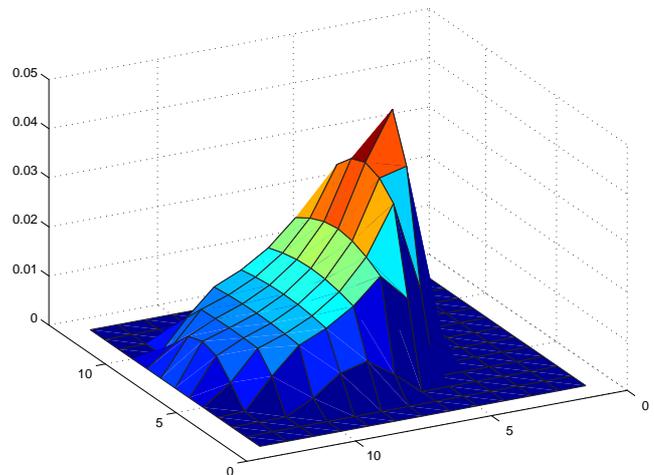
$$h(\mathbf{r})[x, y; s, t] = \frac{1}{\mathbf{r}^2(x, y)} \mathbf{h}\left(\frac{s}{\mathbf{r}(x, y)}, \frac{t}{\mathbf{r}(x, y)}\right). \quad (16)$$

For simplicity, we supposed that the prototype PSF is the same for both images.

We generated two channels (two images) and supposed they were captured with the same camera settings except of the aperture, which was considered 1.2 times larger for the

(a) original image,  $245 \times 356$  pixels

(b) artificial depth map



(c) PSF 13x13

**Figure 1:** Original image, artificial depth map and prototype PSF used for simulated experiments.  $Z$ -coordinate of the depth map actually indicates the radius of related PSF, i. e. approximately half of its support.

second image, i. e.  $\alpha_2 = 1.2$ ,  $\beta_2 = 0$ . The result of blurring can be seen in Fig. 2. Note that the most blurred lower part of the image corresponds to “rear” part of blur map Fig. 1(b).

Note that from (16) we can compute the mask gradients analytically. Instead, we precomputed a table with precision  $1/100$  of pixel (together 600 masks) and in the algorithm we used linear interpolation to get intermediate values, with the intention to simulate the situation, when only some discrete set of masks is available.

The experiment was run at three noise levels – zero (SNR =  $\infty$ ), low (40 dB), moderate (20 dB). Results are arranged in two-column table Fig. 3 with each line corresponding to certain noise level.

Since we know the corresponding ground true Fig. 1, all the figures of reconstructed images and scene structures contain the related value of mean square error (MSE). For images it is given in grey levels per pixel from 256 possible values. The error of depth estimate is given indirectly as the error of related blur map in pixels since it has no meaning to measure directly the error of distance, which depends on camera settings such as aperture diameter or focal length.

All experiments were run several times for different in-

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

## 5 Conclusion

We have presented an algorithm for simultaneous recovery of scene structure and reconstruction of sharp image from two or more defocused images. The PSF was assumed to be a known function of depth constraining the points of the same depth to produce the same mask regardless of their position in the field of view. This assumption can be relaxed at the cost of higher time or memory requirements.

Experiments have shown that the algorithm is robust to noise and for up to moderate noise levels (SNR = 20 dB) gives very satisfactory results. If the noise level is low, the restored image is almost visually undistinguishable from the original except of places of abrupt depth changes. Also the precision of the recovered scene structure approaching



(a) MSE = 17.21 levels



(b) MSE = 19.19 levels

**Figure 2:** To simulate defocusation, we blurred image Fig. 1(a) using depth map Fig. 1(b) and PSFs generated from prototype Fig. 1(c). The largest PSF support (in the lower part of the left image) is about  $11 \times 11$  pixels. Amount of blur in the second channel (right image) is 1.2 times larger than in the first channel (left image), i. e.  $\alpha_2 = 1.2$ .

0.15 pixels is very satisfying and probably it is not possible to achieve much better results in principle.

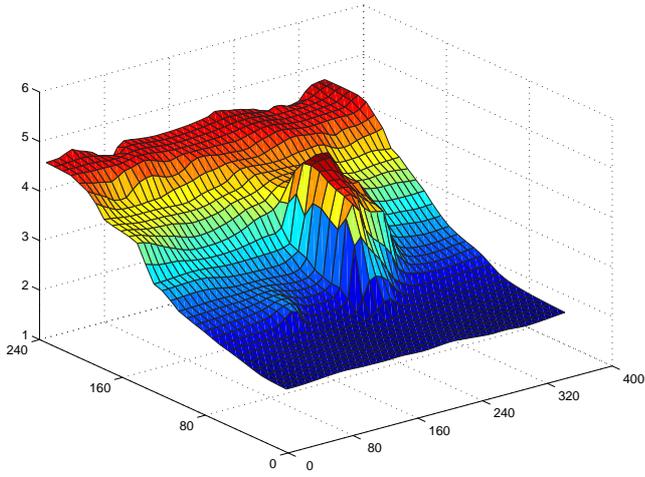
Further research aims at situations of unknown PSF.

## Acknowledgement

This work has been supported by the Ministry of Education of the Czech Republic under the project No. 1M6798555601 (Research Center DAR) and by the Grant Agency of the Czech Republic under the grant No. 102/04/0155.

## References

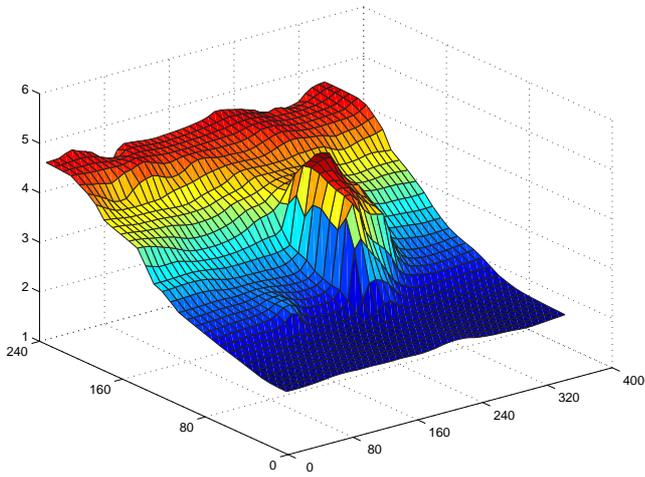
- [1] M.R. Banham and A.K. Katsaggelos. Digital image restoration. *IEEE Signal Processing Mag.*, 14(2):24–41, Mar. 1997.
- [2] A. Chambolle and P. Lions. Image recovery via total variation minimization and related problems. *Numer. Math.*, 76(2):167–188, Apr. 1997.
- [3] Francois Deschênes, Djemel Ziou, and Philippe Fuchs. Simultaneous computation of defocus blur and apparent shifts in spatial domain. In *Proc. Int. Conf. on Vision Interface*, page 236, 2002.
- [4] J. Ens and P. Lawrence. An investigation of methods for determining depth from focus. *IEEE Trans. Pattern Anal. Machine Intell.*, 15(2):97–108, Feb. 1993.
- [5] Paolo Favaro, Martin Burger, and Stefano Soatto. Scene and motion reconstruction from defocus and motion-blurred images via anisotropic diffusion. In Tomáš Pajdla and Jan Matas, editors, *ECCV 2004, LNCS 3021, Springer Verlag, Berlin Heidelberg*, pages 257–269, 2004.
- [6] Paolo Favaro and Stefano Soatto. A geometric approach to shape from defocus. *IEEE Trans. Pattern Anal. Machine Intell.*, 27(3), Mar. 2005.
- [7] R. L. Lagendijk and J. Biemond. Block-adaptive image identification and restoration. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing*, pages 2497–2500, May 1991.
- [8] A. P. Pentland. A new sense for depth of field. *IEEE Trans. Pattern Anal. Machine Intell.*, 9(4):523–531, July 1987.
- [9] A. N. Rajagopalan and S. Chaudhuri. An mrf model-based approach to simultaneous recovery of depth and restoration from defocused images. *IEEE Trans. Pattern Anal. Machine Intell.*, 21(7), July 1999.
- [10] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.
- [11] Michal Šorel. *Multichannel blind restoration of images with space-variant degradations*. PhD thesis, Faculty of Mathematics and Physics, Charles University, Prague, 2006.
- [12] Muralidhara Subbarao and G. Surya. Depth from defocus: a spatial domain approach. *International Journal of Computer Vision*, 3(13):271–294, 1994.
- [13] Masahiro Watanabe and Shree K. Nayar. Rational filters for passive depth from defocus. *International Journal of Computer Vision*, 3(27):203–225, 1998.
- [14] Yalin Xiong and Steven A. Schafer. Moment and hypergeometric filters for high precision computation of focus, stereo and optical flow. Technical Report CMU-RI-TR-94-28, Carnegie Mellon University, Sept. 1994.
- [15] Yu-Li You and Mostafa Kaveh. Blind image restoration by anisotropic regularization. *IEEE Trans. Image Processing*, 8(3), Mar. 1999.



(a) SNR =  $\infty$ , MSE = 0.15 pixels



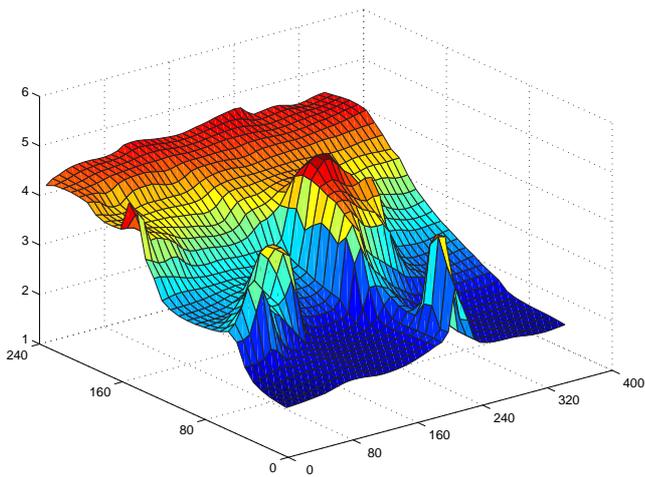
(b) SNR =  $\infty$ , MSE = 6.12 levels



(c) SNR = 40 dB, MSE = 0.15 pixels



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



(e) SNR = 20 dB, MSE = 0.31 pixels



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

**Figure 3:** Recovered scene structure (left column) and corresponding image restoration (right column). Up to moderate noise levels (SNR = 20 dB) gives very good results. Iteration scheme  $50 \times (8 + 10) + 100$ .