Single Image Blind Deconvolution with Higher-Order Texture Statistics[⋆]

Manuel Martinello and Paolo Favaro

Heriot-Watt University
School of EPS, Edinburgh EH14 4AS, UK

Abstract. We present a novel method for solving blind deconvolution, i.e., the task of recovering a sharp image given a blurry one. We focus on blurry images obtained from a coded aperture camera, where both the camera and the scene are static, and allow blur to vary across the image domain. As most methods for blind deconvolution, we solve the problem in two steps: First, we estimate the coded blur scale at each pixel; second, we deconvolve the blurry image given the estimated blur. Our approach is to use linear high-order priors for texture and second-order priors for the blur scale map, i.e., constraints involving two pixels at a time. We show that by incorporating the texture priors in a least-squares energy minimization we can transform the initial blind deconvolution task in a simpler optimization problem. One of the striking features of the simplified optimization problem is that the parameters that define the functional can be learned offline directly from natural images via singular value decomposition. We also show a geometrical interpretation of image blurring and explain our method from this viewpoint. In doing so we devise a novel technique to design optimally coded apertures. Finally, our coded blur identification results in computing convolutions, rather than deconvolutions, which are stable operations. We will demonstrate in several experiments that this additional stability allows the method to deal with large blur. We also compare our method to existing algorithms in the literature and show that we achieve state-of-the-art performance with both synthetic and real data.

Keywords: coded aperture, single image, image deblurring, depth estimation.

1 Introduction

Recently there has been enormous progress in image deblurring from a single image. Perhaps one of the most remarkable results is to have shown that it is possible to extend the depth of field of a camera by modifying the camera optical response [1,2,3,4,5,6,7]. Moreover, techniques based on applying a mask at the lens aperture have demonstrated the ability to recover a coarse depth of the

[⋆] This research was partly supported by SELEX Galileo grant SELEX/HWU/2010/SOW3.

D. Cremers et al. (Eds.): Video Processing and Computational Video, LNCS 7082, pp. 124–151, 2011.

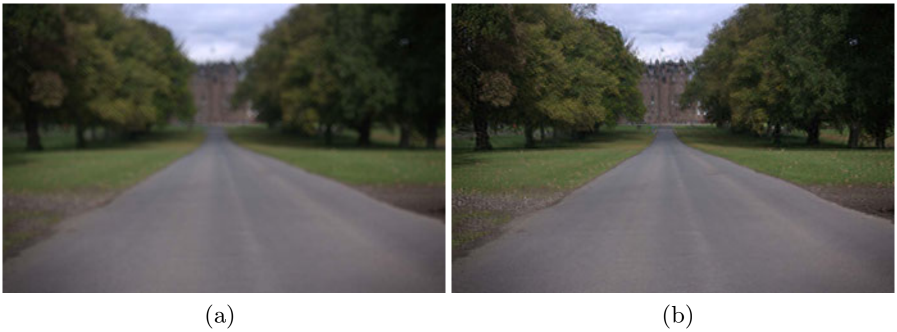


Fig. 1. Results on an outdoor scene [exposure time 1/200s]. (a) Blurry coded image captured with mask b (see Fig. 4). (b) Sharp image reconstructed with our method.

scene [4,5,8]. Depth has then been used for digital refocusing [9] and advanced image editing.

In this paper we present a novel method for image deblurring and demonstrate it on blurred images obtained from a coded aperture camera. Our algorithm uses as input a single blurred image (see Fig. 1 (a)) and automatically returns the corresponding sharp one (see Fig. 1 (b)). Our main contribution is to provide a computationally efficient method that achieves state-of-the-art performance in terms of depth and image reconstruction with coded aperture cameras. We demonstrate experimentally that our algorithm can deal with larger amounts of blur than previous coded aperture methods.

One of the leading approaches in the literature [5] recovers a sharp image by sequentially testing a deconvolution method for several given hypotheses for the blur scale. Then, the blur scale that yields a sharp image that is consistent with both the model and the texture priors is chosen. In contrast, in our approach we show that one can identify the blur scale by computing *convolutions*, rather than deconvolutions, of the blurry image with a finite set of filters. As a consequence, our method is numerically stable especially when dealing with large blur scales. In the next sections, we present all the steps needed to define our algorithm for image deblurring. The task is split in two steps: First the blur scale is identified and second, the coded image is deblurred with the estimated blur scale. We present an algorithm for blur scale identification in section 3.1. Image deblurring is then solved iteratively in section 3.2. A discussion on mask selection is then presented in section 4.1. Comparisons to existing methods are shown in section 5.

1.1 Prior Work

This work relates to several fields ranging from computer vision to image and signal processing, and from optics to astronomy and computer graphics. For simplicity, we group past work based on the technique being employed.

Coded Imaging: Early work in coded imaging appears in the field of astronomy. One of the most interesting pattern designs is the Modified Uniformly Redundant Arrays (MURA) [10] for which a simple coding and decoding procedure was devised (see one such pattern in Fig. 4). In our tests the MURA pattern seems very well behaved, but too sensitive to noise (see Fig. 5). Coded patterns have also been used to design lensless systems, but these systems require either long exposures or are sensitive to noise [11]. More recently, coding of the exposure [12] or of the aperture [4] has been used to preserve high spatial frequencies in blurred images so that deblurring is well-posed. We test the mask proposed in [4] and find that it works well for image deblurring, but not for blur scale identification. A mask that we have tested and has yielded good performance is the four-holes mask of Hiura and Matsuyama [13]. In [13] however, the authors used multiple images. A study on good apertures for deblurring multiple coded images via Wiener filtering has instead led to two novel designs [14,15]. Although the masks were designed to be used together, we have tested each of them independently for comparisons purposes. We found, as predicted by the authors, that the masks are quite robust to noise and quite well designed for image deblurring. Image deblurring and depth estimation with a coded aperture camera has also been demonstrated by Levin et al. [5]. One of their main contributions is the design of an optimal mask. We indeed find this mask quite effective both on synthetic data and real data. However, as already noticed in [16], we have found that the coded aperture technique, if approached as in [5], fails when dealing with large blur amounts. The method we propose in this paper, instead, overtakes this limitation, especially when using the four-hole mask. Finally, a design based on annular masks has also been proposed in [17] and has been exploited for depth estimation in [3]. We also tested this mask in our experiments, but, contrary to our expectations, we did not find its performance superior to the other masks.

3D Point Spread Functions: While there are several techniques to extract depth from images, we briefly mention some recent work by Greengard et al. [18] because their optical design included and exploited diffraction effects. They investigated 3D point spread functions (PSF) whose transverse cross sections rotate as a result of diffraction, and showed that such PSFs yield an order of magnitude increase in the sensitivity with respect to depth variations. The main drawback however, is that the depth range and resolution is limited due to the angular resolution of the reconstructed PSF.

Depth-Invariant Blur: An alternative approach to coded imaging is wavefront coding. The key idea is to use aspheric lenses to render the lens point spread function (PSF) depth-invariant. Then, shift-invariant deblurring with a fixed known blur can be applied to sharpen the image [19,20]. However, while the results are quite promising, the PSF is not fully depth-invariant and artifacts are still present in the reconstructed image. Other techniques based on depth-invariant PSFs exploit the chromatic aberrations of lenses [7] or use diffusion [21]. However, in the first case, as the focal sweep is across the spectrum, the method is mostly designed for grayscale imaging. While the results shown in these recent works are stunning, there are two inherent limitations: 1) Depth is lost in the imaging process; 2) In general, as method based on focal sweep are not exactly depth-invariant, the deblurring performance decays for objects that are too close or too far away from the camera.

Multiple Viewpoint: The extension of the depth of field can also be achieved by using multiple images and/or multiple viewpoints. One technique is to obtain multiple viewpoints by capturing multiple coded images [8,13,22] or by capturing a single image by using a plenoptic camera [9,6,23,24]. These methods however, exploit multiple images or require a more costly optical design (e.g., a calibrated microlens array).

Motion Deblurring and Blind Deconvolution: This work also relates to work in blind deconvolution, and in particular on motion deblurring. There has been a quite steady progress in uniform motion deblurring [25,26,27,28,29] thanks to the modeling and exploitation of texture statistics. Although these methods deal with an unknown and general blur pattern, they assume that blur is not changing across the image domain. More recently, the space-varying case has been studied [30,31,32] albeit with some restrictions on the type of motion or the scene depth structure.

Blurred Face Recognition: Work in the recognition of blurred faces [33] is also related to our method. Their approach extracts features from motion-blurred images of faces and then uses the subspace distance to identify the blur. In contrast, our method can be applied to space-varying blur and our analysis provides a novel method to evaluate (and design) masks.

2 Single Image Blind Deconvolution

Blind deconvolution from a single image is a very challenging problem: We need to recover more unknowns than the available observations. This challenge will be illustrated in the next section, where we present the image formation model of a blurred image obtained from a coded aperture camera. To make the problem feasible and well-behaved, one can introduce additional constraints on the solution. In particular, we constrain the higher-order statistics of sharp texture (sec. 2.2) and impose that the blur scale be piecewise smooth across the image pixels (sec. 2.3).

2.1 Image Model

In the simplest instance, a blurred image of a plane facing the camera can be described via the convolution of a sharp image with the blur kernel. However, the convolutional model breaks down with more general surfaces and, in particular, at occlusion boundaries. In this case, one can describe a blurred image with a linear model. For the sake of notational simplicity, we write images as column vectors, where all pixels are sorted in lexicographical order. Thus, a blurred image with N pixels is a column vector $\boldsymbol{g} \in \mathbb{R}^N$. Similarly, a sharp image with M pixels is a column vector $\boldsymbol{f} \in \mathbb{R}^M$. Then, $\boldsymbol{g}$ satisfies

$$\boldsymbol{g} = \boldsymbol{H_d f}, \tag{1}$$

where the $N \times M$ matrix $\boldsymbol{H_d}$ represents the coded blur. $\boldsymbol{d}$ is a column vector with M pixels and collects the blur scale corresponding to each pixel of $\boldsymbol{f}$. The i-th column of $\boldsymbol{H_d}$ is an image, rearranged as a vector, of the coded blur with scale $\boldsymbol{d}_i$ generated by the i-th pixel of $\boldsymbol{f}$. Notice that this model is indeed a generalization of the convolutional case. In the convolutional model, $\boldsymbol{H_d}$ reduces to a Toeplitz matrix.

Our task is to recover the unknown sharp image $\boldsymbol{f}$ given the blurred image $\boldsymbol{g}$. To achieve this goal it is necessary to recover the blur scale at each pixel $\boldsymbol{d}$. The theory of linear algebra tells us that: If $N = M$ and the equations in eq. (1) are not linearly dependent, and we are given both $\boldsymbol{g}$ and $\boldsymbol{H_d}$, then we can recover the sharp image $\boldsymbol{f}$. However, in our case we are not given the matrix $\boldsymbol{H_d}$ and the blurred image $\boldsymbol{g}$ is affected by noise. This introduces two challenges: First, to obtain $\boldsymbol{H_d}$ we need to retrieve the blur scale $\boldsymbol{d}$; second, because of noise in $\boldsymbol{g}$ and of the ill-conditioning of the linear system in eq. (1), the estimation of $\boldsymbol{f}$ might be unstable. The first challenge implies that we do not have a unique solution. The second challenge implies that even if the solution were unique, its estimation would not be reliable. However, not all is lost. It is possible to add more equations to eq. (1) until a unique reliable solution can be obtained. This technique is based on observing that, typically, one expects the unknown sharp image and blur scale map to have some regularity. For instance, both sharp textures and blur scale maps are not likely to look like noise. In the next two sections we will present and illustrate our sharp image and blur scale priors.

2.2 Sharp Image Prior

Images of the real world exhibit statistical regularities that have been studied intensively in the past 20 years and have been linked to the human visual system and its evolution [34]. For the purpose of image deblurring, the most important aspect of this study is that natural images form a much smaller subset of all possible images. In general, the characterization of the statistical properties of natural images is done by applying a given transform, typically related to a component of human vision. Among the most common statistics used in image processing are the second order statistics, i.e., relations between pairs of pixels. For instance, this category includes the distributions of image gradients [35,36].

However, a more accurate account of the image structure can be captured with high-order statistics, i.e., relations between several pixels. In this work, we consider this general case, but restrict the relations to linear ones of the form

$$\boldsymbol{\Sigma f} \simeq 0 \tag{2}$$

where $\boldsymbol{\Sigma}$ is a rectangular matrix. Eq. (2) implies that all sharp images live approximately on a subspace. Despite their crude simplicity, these linear constraints allow for some flexibility. For example, the case of second-order statistics results in rows of $\boldsymbol{\Sigma}$ with only two nonzero values. Also, by designing $\boldsymbol{\Sigma}$ one can selectively apply the constraints only on some of the pixels. Another example is to choose each row of $\boldsymbol{\Sigma}$ as a Haar feature applied to some pixels. Notice that in our approach we do not make any of these choices. Rather, we estimate $\boldsymbol{\Sigma}$ directly from natural images.

Natural image statistics, such as gradients, typically exhibit a peaked distribution. However, performing inference on such distributions results in minimizations of non convex functionals for which we do not have probably optimal algorithms. Furthermore, we are interested in simplifying the optimization task as much as possible to gain in computational efficiency. This has led us to enforce the linear relation above by minimizing the convex cost

$$\|\boldsymbol{\Sigma f}\|_2^2. \tag{3}$$

As we do not have an analytical expression for $\boldsymbol{\Sigma}$ that satisfies eq. (2), we need to learn it directly from the data. We will see later that this step is necessary only when performing the deconvolution step given the estimated blur. Instead, when estimating the blur scale our method allows us to use $\boldsymbol{\Sigma}$ implicitly, i.e., without ever recovering it.

2.3 Blur Scale Prior

The statistics of range images can be characterized with an approach similar to that for optical images [37]. The study in [37] verified the random collage model, i.e., that a scene is a collection of piecewise constant surfaces. This has been observed in the distributions of Haar filter responses on the logarithm of the range data, which showed strong cusps in the isoprobability contours. Unfortunately, a prior following these distributions faithfully would result in non convex energy minimization. A practical convex solution to enforce the piecewise constant model, is to use total variation [38]. Common choices are the isotropic and anisotropic total variation. In our algorithm we have implemented the latter. We minimize $\|\nabla \boldsymbol{d}\|_1$, i.e., the sum of the absolute value of the components of the gradient of $\boldsymbol{d}$.

3 Blur Scale Identification and Image Deblurring

We can combine the image model introduced in sec. 2.1 with the priors in sec. 2.2 and 2.3 and formulate the following energy minimization problem:

$$\hat{\boldsymbol{d}}, \hat{\boldsymbol{f}} = \underset{\boldsymbol{d},\boldsymbol{f}}{\operatorname{argmin}} \|\boldsymbol{g} - \boldsymbol{H_d f}\|_2^2 + \alpha\|\boldsymbol{\Sigma f}\|_2^2 + \beta\|\nabla \boldsymbol{d}\|_1, \tag{4}$$

where the parameters $\alpha, \beta > 0$ determine the amount of regularization for texture and blur scale respectively. Notice that the formulation above is common to many approaches including, in particular, [5]. Our approach, however, in addition to using a more accurate blur matrix $\boldsymbol{H_d}$, considers different priors and a different depth identification procedure.

Our next step is to notice that, given $\boldsymbol{d}$, the proposed cost is simply a least-squares problem in the unknown sharp texture $\boldsymbol{f}$. Hence, it is possible to compute $\boldsymbol{f}$ in closed-form and plug it back in the cost functional. The result is a much simpler problem to solve. We summarize all the steps in the following Theorem:

Theorem 1. *The set of extrema of the minimization* (4) *coincides with the set of extrema of the minimization*

$$\begin{cases} \hat{\boldsymbol{d}} = \underset{\boldsymbol{d}}{\operatorname{argmin}} \, \|\boldsymbol{H}_{\boldsymbol{d}}^{\perp}\boldsymbol{g}\|_2^2 + \beta\|\nabla\boldsymbol{d}\|_1 \\ \hat{\boldsymbol{f}} = \left(\alpha\boldsymbol{\Sigma}^T\boldsymbol{\Sigma} + \boldsymbol{H}_{\hat{\boldsymbol{d}}}^T\boldsymbol{H}_{\hat{\boldsymbol{d}}}\right)^{-1}\boldsymbol{H}_{\hat{\boldsymbol{d}}}^T\boldsymbol{g} \end{cases} \tag{5}$$

where $\boldsymbol{H}_{\boldsymbol{d}}^{\perp} \doteq \boldsymbol{I} - \boldsymbol{H}_{\boldsymbol{d}}\left(\alpha\boldsymbol{\Sigma}^T\boldsymbol{\Sigma} + \boldsymbol{H}_{\boldsymbol{d}}^T\boldsymbol{H}_{\boldsymbol{d}}\right)^{-1}\boldsymbol{H}_{\boldsymbol{d}}^T$, *and* $\boldsymbol{I}$ *is the identity matrix.*

Proof. See Appendix.

Notice that the new formulation requires the definition of a square and symmetric matrix $\boldsymbol{H}_{\boldsymbol{d}}^{\perp}$. This matrix depends on the parameter α and the prior matrix $\boldsymbol{\Sigma}$, both of which are unknown. However, for the purpose of estimating the unknown blur scale map $\boldsymbol{d}$, it is possible to bypass the estimation of α and $\boldsymbol{\Sigma}$ by learning directly the matrix $\boldsymbol{H}_{\boldsymbol{d}}^{\perp}$ from data.

3.1 Learning Procedure and Blur Scale Identification

We break down the complexity of solving eq. (5) by using local blur uniformity, i.e., by assuming that blur is constant within a small region of pixels. Then, we further simplify the problem by considering only a finite set of L blur sizes $d_1, \dots, d_L$. In practice, we find that both assumptions work well. The local blur uniformity holds reasonably well except at occluding boundaries, which form a small subset of the image domain. At occluding boundaries the solution tends to favor small blur estimates. We also found experimentally that the discretization is not a limiting factor in our method. The number of blur sizes L can be set to a value that matches the level of accuracy of the method without reaching a prohibitive computational load.

Now, by combining the assumptions we find that eq. (5) at one pixel $\mathbf{x}$

$$\hat{\boldsymbol{d}}(\mathbf{x}) = \underset{\boldsymbol{d}(\mathbf{x})}{\operatorname{argmin}} \, \|\boldsymbol{H}_{\boldsymbol{d}}^{\perp}(\mathbf{x})\boldsymbol{g}\|_2^2 + \beta\|\nabla\boldsymbol{d}(\mathbf{x})\|_1 \tag{6}$$

can be approximated by

$$\hat{\boldsymbol{d}}(\mathbf{x}) = \underset{\boldsymbol{d}(\mathbf{x})}{\operatorname{argmin}} \, \|\boldsymbol{H}_{\boldsymbol{d}(\mathbf{x})}^{\perp}\boldsymbol{g}_{\mathbf{x}}\|_2^2 \tag{7}$$

where $\boldsymbol{g}_{\mathbf{x}}$ is a column vector of δ^2 pixels extracted from a $\delta \times \delta$ patch centered at the pixel $\mathbf{x}$ of $\boldsymbol{g}$. Experimentally, we find that the size δ of the patch should not be smaller than the maximum scale of the coded blur in the captured image

$\boldsymbol{g}$. $\boldsymbol{H}^{\perp}_{\boldsymbol{d}(\mathbf{x})}$ is a $\delta^2 \times \delta^2$ matrix that depends on the blur size $\boldsymbol{d}(\mathbf{x}) \in \{d_1, \ldots, d_L\}$. So we assume that $\boldsymbol{H}^{\perp}_{\boldsymbol{d}}(\mathbf{x}, \mathbf{y}) \simeq 0$ for $\mathbf{y}$ such that $\|\mathbf{y} - \mathbf{x}\|_1 > \delta/2$. Notice that the term $\beta\|\nabla \boldsymbol{d}\|_1$ drops because of the local blur uniformity assumption.

The next step is to explicitly compute $\boldsymbol{H}^{\perp}_{\boldsymbol{d}(\mathbf{x})}$. Since the blur size $\boldsymbol{d}(\mathbf{x})$ is one of L values, we only need to compute $\boldsymbol{H}^{\perp}_{d_1}, \ldots, \boldsymbol{H}^{\perp}_{d_L}$ matrices. As each $\boldsymbol{H}^{\perp}_{d_i}$ depends on α and the local $\boldsymbol{\Sigma}$, we propose to learn each $\boldsymbol{H}^{\perp}_{d_i}$ directly from data. Suppose that we are given a set of T column vectors $\boldsymbol{g}_{\mathbf{x}_1}, \ldots, \boldsymbol{g}_{\mathbf{x}_T}$ extracted from blurry images of a plane parallel to the camera image plane. The column vectors will all share the same blur scale d_i. Hence, we can rewrite the cost functional in eq. (7) for all $\mathbf{x}$ as

$$\|\boldsymbol{H}^{\perp}_{d_i}\boldsymbol{G}_i\|_2^2 \tag{8}$$

where $\boldsymbol{G}_i \doteq [\boldsymbol{g}_{\mathbf{x}_1} \ \cdots \ \boldsymbol{g}_{\mathbf{x}_T}]$. By definition of $\boldsymbol{G}_i$, $\|\boldsymbol{H}^{\perp}_{d_i}\boldsymbol{G}_i\|_2^2 = 0$. Hence, we find that $\boldsymbol{H}^{\perp}_{d_i}$ can be computed via the singular value decomposition of $\boldsymbol{G}_i = U_i S_i V_i^T$. If $U_i = [\boldsymbol{U}_{\boldsymbol{d}_i} \ \boldsymbol{Q}_{\boldsymbol{d}_i}]$ where $\boldsymbol{Q}_{\boldsymbol{d}_i}$ corresponds to the singular values of S_i that are zero (or negligible), then $\boldsymbol{H}^{\perp}_{d_i} = \boldsymbol{Q}_{\boldsymbol{d}_i}\boldsymbol{Q}^T_{\boldsymbol{d}_i}$. The procedure is then repeated for each blur scale d_i with $i = 1, \ldots, L$.

Next, we can use the estimated matrices $\boldsymbol{H}^{\perp}_{d_1}, \ldots, \boldsymbol{H}^{\perp}_{d_L}$ on a new image $\boldsymbol{g}$ and optimize with respect to $\boldsymbol{d}$:

$$\hat{\boldsymbol{d}} = \underset{\boldsymbol{d}}{\operatorname{argmin}} \sum_{\mathbf{x}} \|\boldsymbol{H}^{\perp}_{\boldsymbol{d}(\mathbf{x})}\boldsymbol{g}_{\mathbf{x}}\|_2^2 + \beta\|\nabla \boldsymbol{d}(\mathbf{x})\|_1. \tag{9}$$

The first term represents unitary terms, i.e., terms that are defined on single pixels; the second term represents binary terms, i.e., terms that are defined on pairs of pixels. The minimization problem (9) can then be solved efficiently via graph cuts [39].

Notice that the procedure above can be applied to other surfaces as well, so that instead of a collection of parallel planes, one can consider, for example, a collection of quadratic surfaces. Also, notice that there are no restrictions on the size of a patch. In particular, the same procedure can be applied to a patch of the size of the input image. In our experiments for depth estimation, however, we consider only small patches and parallel planes as local surfaces.

3.2 Image Deblurring

In the previous section we have devised a procedure to compute the blur scale at each pixel $\boldsymbol{d}$. In this section we assume that $\boldsymbol{d}$ is given and devise a procedure to compute the image $\boldsymbol{f}$. In principle, one could use the closed-form solution

$$\boldsymbol{f} = \left(\alpha\boldsymbol{\Sigma}^T\boldsymbol{\Sigma} + \boldsymbol{H}^T_{\hat{\boldsymbol{d}}}\boldsymbol{H}_{\hat{\boldsymbol{d}}}\right)^{-1} \boldsymbol{H}^T_{\hat{\boldsymbol{d}}}\boldsymbol{g}. \tag{10}$$

However, notice that computing this equation entails solving a large matrix inversion, which is not practical for moderate image dimensions. A simpler approach is to solve the least squares problem (4) in $\boldsymbol{f}$ via an iterative method. Therefore, we consider solving the problem

$$\hat{\boldsymbol{f}} = \underset{\boldsymbol{f}}{\operatorname{argmin}} \|\boldsymbol{g} - \boldsymbol{H}_{\hat{\boldsymbol{d}}}\boldsymbol{f}\|_2^2 + \alpha\|\boldsymbol{\Sigma}\boldsymbol{f}\|_2^2 \tag{11}$$

by using a least-squares conjugate gradient descent algorithm in $\boldsymbol{f}$ [40]. The main component for the iteration in $\boldsymbol{f}$ is the gradient $\nabla E_{\boldsymbol{f}}$ of the cost (11) with respect to $\boldsymbol{f}$

$$\nabla E_{\boldsymbol{f}} = \left(\alpha \boldsymbol{\Sigma}^T \boldsymbol{\Sigma} + \boldsymbol{H}_{\hat{\boldsymbol{d}}}^T \boldsymbol{H}_{\hat{\boldsymbol{d}}}\right) \boldsymbol{f} - \boldsymbol{H}_{\hat{\boldsymbol{d}}}^T \boldsymbol{g}. \tag{12}$$

The descent algorithm iterates until $\nabla E_{\boldsymbol{f}} \simeq \mathbf{0}$. Because of the convexity of the cost functional with respect to $\boldsymbol{f}$, the solution is also a global minimum.

To compute $\boldsymbol{\Sigma}$ we use a database of sharp images $\boldsymbol{F} = [\boldsymbol{f}_1 \cdots \boldsymbol{f}_T]$ where $\{\boldsymbol{f}_i\}_{i=1,\dots,T}$ are sharp images rearranged as column vectors, and compute the singular value decomposition $\boldsymbol{F} = U_F \Sigma_F V_F^T$. Then, we partition $U_F = [U_{F,1}\ U_{F,2}]$ such that $U_{F,2}$ corresponds to the smallest singular values of Σ_F. The high-order prior is defined as $\boldsymbol{\Sigma} \doteq U_{F,2} U_{F,2}^T$, such that we have $\boldsymbol{\Sigma} \boldsymbol{f}_i \approx 0$. The regularization parameter α is instead manually tuned. The matrix $\boldsymbol{H}_{\hat{\boldsymbol{d}}}$ is computed as described in Section 2.1.

4 A Geometric Viewpoint on Blur Scale Identification

In the previous sections we have seen that the blur scale at each pixel can be obtained by minimizing eq. (9). We search among matrices $\boldsymbol{H}_{d_1}^{\perp}, \dots, \boldsymbol{H}_{d_L}^{\perp}$ the one that yields the minimum ℓ_2 norm when applied to the vector $\boldsymbol{g}_{\mathbf{x}}$. We show that this has a geometrical interpretation: Each matrix $\boldsymbol{H}_{d_i}^{\perp}$ defines a subspace and $\|\boldsymbol{H}_{d_i}^{\perp} \boldsymbol{g}_{\mathbf{x}}\|_2^2$ is the distance of each vector $\boldsymbol{g}_{\mathbf{x}}$ from that subspace.

Recall that $\boldsymbol{H}_{d_i}^{\perp} = \boldsymbol{Q}_{\boldsymbol{d}_i} \boldsymbol{Q}_{\boldsymbol{d}_i}^T$ and that $U_i = [\boldsymbol{U}_{\boldsymbol{d}_i}\ \boldsymbol{Q}_{\boldsymbol{d}_i}]$ is an orthonormal matrix. Then, we obtain that $\|\boldsymbol{H}_{d_i}^{\perp} \boldsymbol{g}_{\mathbf{x}}\|_2^2 = \|\boldsymbol{Q}_{\boldsymbol{d}_i} \boldsymbol{Q}_{\boldsymbol{d}_i}^T \boldsymbol{g}_{\mathbf{x}}\|_2^2 = \|\boldsymbol{Q}_{\boldsymbol{d}_i}^T \boldsymbol{g}_{\mathbf{x}}\|_2^2 = \|\boldsymbol{g}_{\mathbf{x}}\|_2^2 - \|\boldsymbol{U}_{\boldsymbol{d}_i}^T \boldsymbol{g}_{\mathbf{x}}\|_2^2$. If we now divide by the scalar number $\|\boldsymbol{g}_{\mathbf{x}}\|_2^2$, we obtain exactly the square of the subspace distance [41]

$$\mathcal{M}(\boldsymbol{g}, \boldsymbol{U}_{\boldsymbol{d}_i}) = \sqrt{1 - \sum_{j=1}^{K} \left(U_{\boldsymbol{d}_i,j}^T \frac{\boldsymbol{g}}{\|\boldsymbol{g}\|} \right)^2} \tag{13}$$

where K is the rank of the subspace $\boldsymbol{U}_{\boldsymbol{d}_i}$, $\boldsymbol{U}_{\boldsymbol{d}_i} = [U_{\boldsymbol{d}_i,1}\ \dots\ U_{\boldsymbol{d}_i,K}]$, and $U_{\boldsymbol{d}_i,j}$, $j = 1, \cdots, K$ are orthonormal vectors.

The geometrical interpretation brings a fresh look to image blurring and deblurring. Consider the image model (1). Let us take the singular value decomposition of the blur matrix $\boldsymbol{H}_{\boldsymbol{d}}$

$$\boldsymbol{H}_{\boldsymbol{d}} = U_{\boldsymbol{d}} S_{\boldsymbol{d}} V_{\boldsymbol{d}}^T \tag{14}$$

where $S_{\boldsymbol{d}}$ is a diagonal matrix with positive entries, and both $U_{\boldsymbol{d}}$ and $V_{\boldsymbol{d}}$ are orthonormal matrices. Formally, the vector $\boldsymbol{f}$ undergoes a rotation ($V_{\boldsymbol{d}}^T$), then a scaling ($S_{\boldsymbol{d}}$), and then again another rotation ($U_{\boldsymbol{d}}$). This means that if $\boldsymbol{f}$ lives in a subspace, the initial subspace is mapped to another rotated subspace, possibly of smaller dimension (see Fig. 2, middle). Notice that as we change the blur scale, the rotations and scaling are also changing and may result in yet a different subspace (see Fig. 2, right).

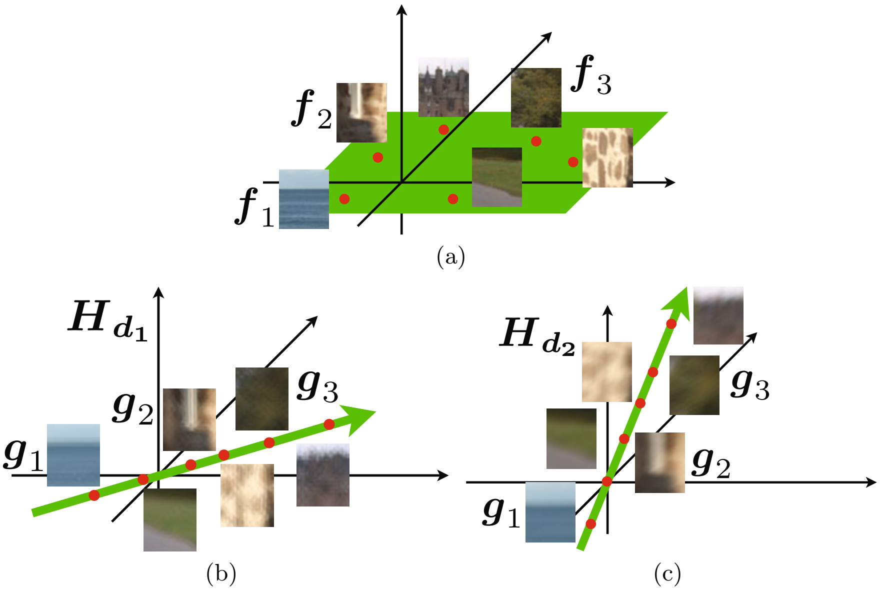


Fig. 2. Coded images subspaces. (a) Image patches on a subspace. (b) Subspace containing images blurred with $\boldsymbol{H}_{\boldsymbol{d}_1}$; blurring has the effect of rotating and possibly reducing the dimensionality of the original subspace. (c) Subspace containing images blurred with $\boldsymbol{H}_{\boldsymbol{d}_2}$.

It is important to understand that rotations of the vector $\boldsymbol{f}$ can result in blurring. To clarify this, consider blurred and sharp images with only 3 pixels (we cannot visualize the case of more than 3 pixels), i.e., $\boldsymbol{g}_1 = [\boldsymbol{g}_{1,x}\ \boldsymbol{g}_{1,y}\ \boldsymbol{g}_{1,z}]^T$ and $\boldsymbol{f}_1 = [\boldsymbol{f}_{1,x}\ \boldsymbol{f}_{1,y}\ \boldsymbol{f}_{1,z}]^T$. Then, we can plot the vectors $\boldsymbol{g}_1$ and $\boldsymbol{f}_1$ as 3D points (see Fig. 2). Let $\|\boldsymbol{g}_1\| = 1$ and $\|\boldsymbol{f}_1\| = 1$. Then, we can rotate $\boldsymbol{f}_1$ about the origin and overlap it exactly on $\boldsymbol{g}_1$. In this case rotation corresponded to blurring. The opposite is also true. We can rotate the vector $\boldsymbol{g}_1$ onto the vector $\boldsymbol{f}_1$ and thus perform deblurring. Furthermore, notice that in this simple example the most blurred images are vectors with identical entries. Such blurred images lie along the diagonal direction $[1\ 1\ 1]^T$. In general, blurry images tend to have entries with similar values and hence tend to cluster around the diagonal direction.

Our ability to discriminate between different blur scales in a blurry image boils down to being able to determine the subspaces where the patches of such blurry image live. If sharp images do not live on a subspace, but uniformly in the entire space, our only way to distinguish the blur size is that the blurring $\boldsymbol{H}_{\boldsymbol{d}}$ scales some dimensions of $\boldsymbol{f}$ to zero and that the scaling varies with blur size. This case has links to the zero-sheet approach in the Fourier domain [42]. However, if the sharp images live on a subspace, the blurring $\boldsymbol{H}_{\boldsymbol{d}}$ may preserve all the directions and blur scale identification is still possible by determining the rotation of the sharp images subspace. This is the principle that we exploit.

```
Input: A single coded image $\boldsymbol{g}$ and a collection of coded images of $L$ planar scenes.
Output: The blur scale map $\boldsymbol{d}$ of the scene.
Preprocessing (offline)
Pick an image patch size larger than twice the maximum blur scale;
for $i = 1, \ldots, L$ do
    Compute the singular value decomposition $U_i S_i V_i^T$ of a collection of image patches coded with blur scale $\boldsymbol{d}_i$ ;
    Calculate the subspace $\boldsymbol{U}_{\boldsymbol{d}_i}$ as the columns of $U_i$ corresponding to nonzero singular values of $S_i$;
end
Blur identification (online)
Solve $\hat{\boldsymbol{d}} = \arg\min_{\boldsymbol{d}\in\{\boldsymbol{d}_1,\cdots,\boldsymbol{d}_L\}} \sum_{\mathbf{x}} \mathcal{M}^2(\boldsymbol{g}_{\mathbf{x}}, \boldsymbol{U}_{\boldsymbol{d}}) + \frac{\beta}{\|\boldsymbol{g}_{\mathbf{x}}\|_2^2}\|\nabla \boldsymbol{d}(\mathbf{x})\|_1$.
```

Algorithm 1. Blur scale identification from a single coded image via the subspace distance method.

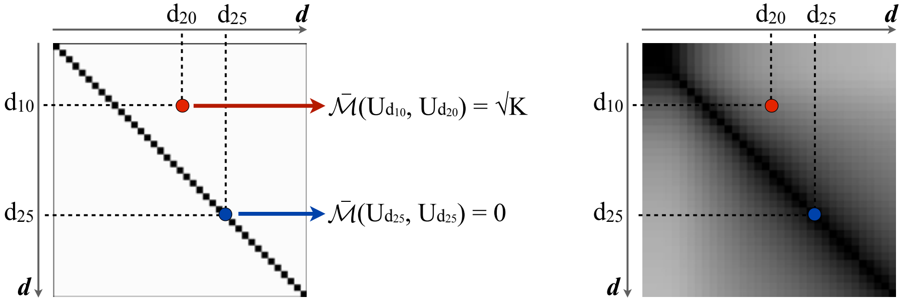
Notice that the evaluation of the subspace distance $\mathcal{M}$ involves the calculation of the inner product between a patch and a column of $\boldsymbol{U}_{\boldsymbol{d}_i}$. Hence, this calculation can be done exactly as the convolution of a column of $\boldsymbol{U}_{\boldsymbol{d}_i}$, rearranged as an image patch, with the whole image $\boldsymbol{g}$. We can conclude that the algorithm requires computing a set of $L \times K$ convolutions with the coded image, which is a stable operation of polynomial computational complexity. As we have shown that minimizing eq. (13) is equivalent to minimizing $\|\boldsymbol{H}_{d_i}^{\perp}\boldsymbol{g}_{\mathbf{x}}\|_2^2$ up to a scalar value, we summarize the blur scale identification procedure in Algorithm 1.

4.1 Coded Aperture Selection

In this section we discuss how to obtain an optimal pattern for the purpose of image deblurring. As pointed out in [19] we identify two main challenges: The first one is that accurate deblurring requires accurate identification of the blur scale; the second one is that accurate deblurring requires little texture loss due to blurring. A first step towards addressing these challenges is to define a metric for blur scale identification and a metric for texture loss. Our metric for blur scale identification can be defined directly from section 4. Indeed, the ability to determine which subspace a coded image patch belongs to can be measured via the distance between the subspaces associated to each blur scale

$$\bar{\mathcal{M}}(\boldsymbol{U}_{\boldsymbol{d}_1}, \boldsymbol{U}_{\boldsymbol{d}_2}) = \sqrt{K - \sum_{i,j}\left(U_{\boldsymbol{d}_1,i}^T U_{\boldsymbol{d}_2,j}\right)^2}. \tag{15}$$

Clearly, the wider apart all the subspaces are, and the less prone to noise the subspace association is. We find that a good visual summary of the "spacing" between all the subspaces is a (symmetric) matrix with distances between any


d20
d25
d
d10
d25
d
$\bar{\mathcal{M}}(U_{d_{10}}, U_{d_{20}}) = \sqrt{K}$
$\bar{\mathcal{M}}(U_{d_{25}}, U_{d_{25}}) = 0$
d20
d25
d
d10
d25
d

(a) Ideal distance matrix

(b) Circular aperture

Fig. 3. Distance matrix computation. The top-left corner of each matrix is the distance between subspaces corresponding to small blur scales, and, vice versa, the bottom-right corner is the distance between subspaces corresponding to large blur scales. Notice that large subspace distances are bright and small subspace distances are dark. The maximum distance ($\sqrt{K}$) is achievable when two subspaces are orthogonal to each other.

two subspaces. We compute such matrix for a conventional camera and show the results in Fig. 3, together with the ideal distance matrix.

In each distance matrix, subspaces associated to blur scales ranging from the smallest to the largest ones are arranged along the rows from left to right and along the columns from top to bottom. Along the diagonal the distance is necessarily 0 as we compare identical subspaces. Also, by definition the metric cannot exceed $\sqrt{K}$, where K is the minimum rank among the subspaces. In Fig. 5 we report the distance matrices computed for each of the apertures we consider in this work (see Fig. 4).

Notice that the subspace distance map for a conventional camera (Fig. 3(b)) is overall darker than the matrices for coded aperture cameras (Fig. 5). This shows the poor blur scale identifiability of the circular aperture and the improvement that can be achieved when using a more elaborate pattern.

The rank K can be used to address the second challenge, i.e., the definition of a metric for texture loss. So far we have seen that blurring can be interpreted as a combination of rotations and scaling. Deblurring can then be interpreted as a combination of rotations and scaling in the opposite direction. However, when blurring scales some directions to 0, part of the texture content has been lost. This suggests that a simple measure for texture loss is the dimension of the coded subspace: The higher the dimension and the more texture content we can restore. As the (coded images) subspace dimension is K, we can immediately conclude that the subspace distance matrix that most closely resembles the ideal distance matrix (see Fig. 3(a)) is the one that simultaneously achieves the best depth identification and the least texture loss. Finally, we propose to use the average L_1 fitting of any distance matrix to the ideal distance matrix scaled of $\sqrt{K}$, i.e., $|\sqrt{K}(\mathbf{1}\mathbf{1}^T - \boldsymbol{I}) - \bar{\mathcal{M}}|$. The fitting yields the values in Table 1. We can also see visually in Fig. 5 that mask 4(b) and mask 4(d) are the coded apertures that we can expect to achieve the best results in texture deblurring.

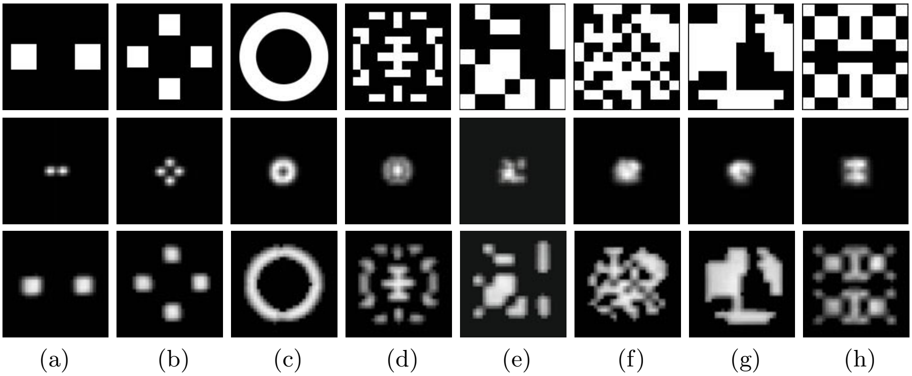


Fig. 4. Coded aperture patterns and PSFs. All the aperture patterns we consider in this work (top row) and their calibrated PSFs for two different blur scales (second and bottom row). (a) and (b) aperture masks used in both [13] and [43]; (c) annular mask used in [17]; (d) pattern proposed by [5]; (e) pattern proposed by [4]; (f) and (g) aperture masks used in [15]; (h) MURA pattern used in [10].

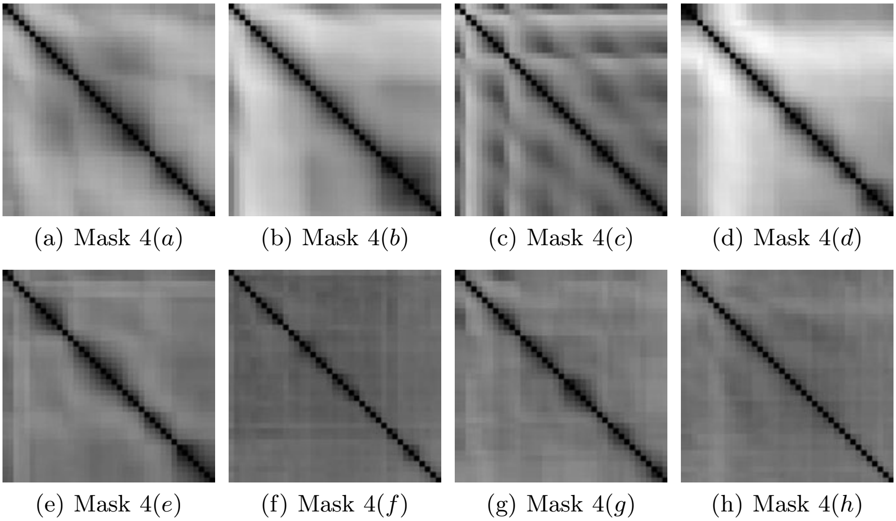

(a) Mask 4(a)
(b) Mask 4(b)
(c) Mask 4(c)
(d) Mask 4(d)
(e) Mask 4(e)
(f) Mask 4(f)
(g) Mask 4(g)
(h) Mask 4(h)

Fig. 5. Subspace distances for the eight masks in Fig. 4. Notice that the subspace rank K determines the maximum distance achievable, and therefore, coded apertures with overall darker subspace distance maps have poor blur scale identifiability (i.e., sensitive to noise).

The quest for the optimal mask is, however, still an open problem. Even if we look for the optimal mask via brute-force search, a single aperture pattern requires the evaluation of eq. (15) and the computation of all the subspaces associated to each blur scale. In particular, the latter process requires about 15 minutes on a QuadCore 2.8GHz with Matlab 7, which makes the evaluation of

Table 1. L_1 **fitting** of any distance matrix to the ideal distance matrix scaled of $\sqrt{K}$

	Masks							
	4(a)	4(b)	4(c)	4(d)	4(e)	4(f)	4(g)	4(h)
L_1 **fitting**	8.24	6.62	8.21	5.63	8.37	16.96	8.17	16.13

a large number of masks unfeasible. Devising a fast procedure to determine the optimal mask will be subject of future work.

5 Experiments

In this section we demonstrate the effectiveness of our approach on both synthetic and real data. We show that the proposed algorithm performs better than previous methods on different coded apertures and different datasets. We also show that the masks proposed in the literature do not always yield the best performance.

5.1 Performance Comparison

Before proceeding with tests on real images, we perform extensive simulations to compare accuracy and robustness of our algorithm with 4 competing methods including the current state-of-the-art approach. The methods are all based on the hypothesis plane deconvolution used by [5] as explained in the Introduction. The main difference among the competing methods is that the deconvolution step is performed either using the Lucy-Richardson method [44], or regularized filtering (i.e., with image gradient smoothness), or Wiener filtering [45], or Levin's procedure [5]. We use the 8 masks shown in Fig. 4. All the patterns have been proposed and used by other researchers [4,5,10,13,15,17]. For each mask and a given blur scale map $\boldsymbol{d}$, we simulate a coded image by using eq. (1), where $\boldsymbol{f}$ is an image of $4,875 \times 125$ pixels with either random texture or a set of patches from natural images (examples of these patches are shown in Fig. 6). Then, for each algorithm we obtain a blur scale map estimate $\hat{\boldsymbol{d}}$ and compute its discrepancy with the ground-truth. The ground-truth blur scale map $\boldsymbol{d}$ that we use is shown in pseudo-colors at the top-left of both Fig. 7 and Fig. 8 and it represents a stair composed of 39 steps at different distances (and thus different blur scales) from the camera. We assume that the focal plane is set to be between the camera and the first object of interest in the scene. With this setting, the bottom part of the blur scale map (small blur sizes) corresponds to points close to the camera, and the top part (large blur sizes) to points far from the camera. Each step of the stair is a square of 125×125 pixels, we have squeezed the actual illustration along the vertical axis to fit in the paper. The size of the blur ranges from 7 to 30 pixels. Notice that in measuring the errors we consider all pixels, including those at the blur scale discontinuities, given by the difference of blur scale between neighboring steps. In Fig. 7 we show, for each mask in Fig. 4, the results of the proposed method (right) together with the results obtained by

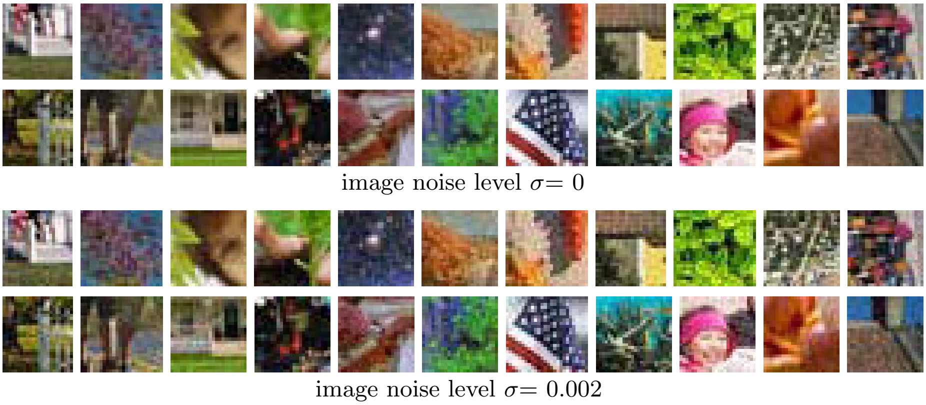


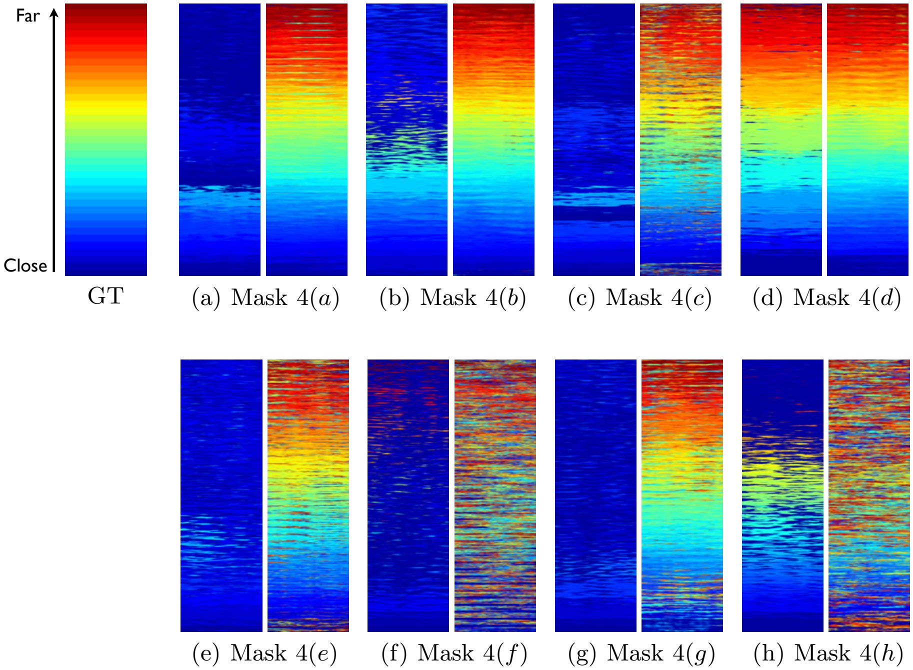
image noise level σ= 0

image noise level σ= 0.002

Fig. 6. Real texture. Some of the patches extracted from real images that have been used in our tests. The same patches are shown with no noise (top part) and when a Gaussian noise is added to them (bottom part).

the current state-of-the-art algorithm (left) on random texture. The same procedure, but with texture from natural images, is reported in Fig. 8. For the three best performing masks (mask $4(a)$, mask $4(b)$, and mask $4(d)$), we report the results with the same graphical layout in Fig. 9, in order to better appreciate the improvement of our method over previous ones, especially for large blur scales. Every plot shows, for each of the 39 steps we consider, the mean and 3 times the standard deviation of the estimated blur scale values (ordinate axis) against the true blur scale level (abscissa axis). The ideal estimate is the diagonal line where each estimated level corresponds to the correct true blur scale level. If there is no bias in the estimation of the blur scale map, the ideal estimate should lie between 3 times the standard deviation about the mean with probability close to 1. Our method performs consistently well with all the masks and at different blur scale levels. In particular, the best performances are observed for mask 4*b* (Fig. 9(b)) and *d* (Fig. 9(c)), while the performance of competing methods rapidly degenerates with increasing pattern scales. This demonstrates that our method has potential for restoring objects at a wider range of blur scales and with higher accuracy than in previous algorithms.

A quantitative comparison among all the methods and masks is given in Table 2 and Table 4 (for random texture) and in Table 3 and Table 5 (for real texture). In each table, the left half reports the average error of the blur scale estimate (measured as $||\boldsymbol{d} - \hat{\boldsymbol{d}}||_1$, where $\boldsymbol{d}$ and $\hat{\boldsymbol{d}}$ are the ground-truth and the estimated blur scale map respectively); the right half reports the error on the reconstructed sharp image $\hat{\boldsymbol{f}}$, measured as $\sqrt{||\boldsymbol{f} - \hat{\boldsymbol{f}}||_2^2 + ||\nabla \boldsymbol{f} - \nabla \hat{\boldsymbol{f}}||_2^2}$, where $\boldsymbol{f}$ is the ground-truth image. The gradient term is added to improve sensitivity to artifacts in the reconstruction. As one can see from Tables 2 - 5, several levels of noise have been considered in the performance comparison: $\sigma = 0$ (Table 2 and



GT (a) Mask 4(a) (b) Mask 4(b) (c) Mask 4(c) (d) Mask 4(d)

(e) Mask 4(e) (f) Mask 4(f) (g) Mask 4(g) (h) Mask 4(h)

Fig. 7. Blur scale estimation - random texture. GT: Ground-truth blur scale map. (a-h) Estimated blur scale maps for all the eight masks we consider in the paper. For each mask, the figure reports the blur scale map estimated with both Levin et al.'s method (left) and our method (right).

Table 3), $\sigma = 0.001$, $\sigma = 0.002$, and $\sigma = 0.005$ (Table 4 and Table 5). The noise level is however adjusted to accommodate the difference in overall incoming light between the masks, i.e., if the mask i has an incoming light of l_i[1], the noise level for that mask is given by:

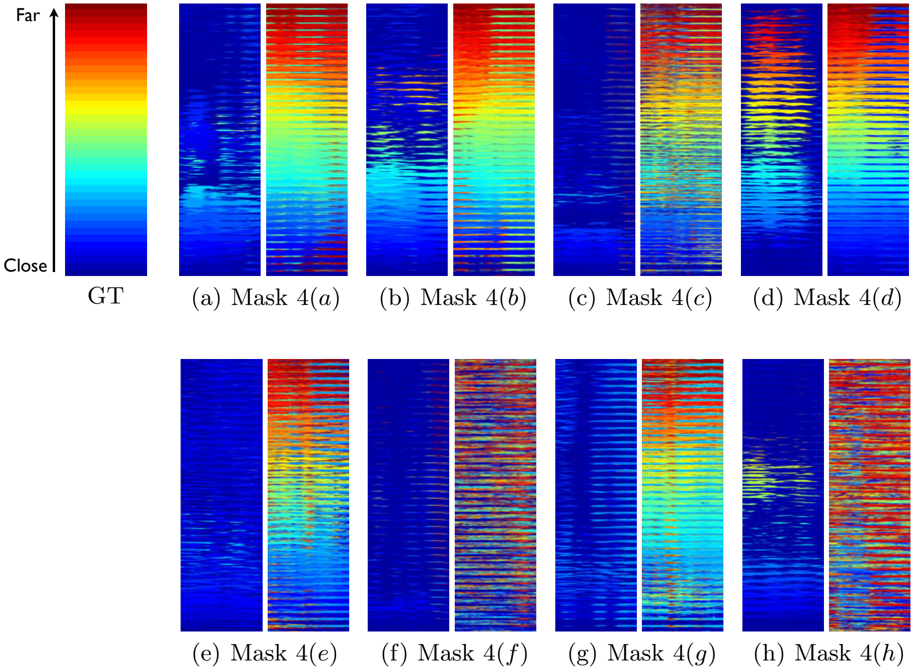
$$\sigma_i = \frac{1}{l_i} * \sigma. \tag{16}$$

Thus, masks such as 4(f), 4(g) and 4(h) are subject to lower noise levels than masks such as 4(a) and 4(b). Our method produces more consistent and accurate blur scale maps than previous methods for both random texture and natural images, and across the 8 masks that it has been tested with.

5.2 Results on Real Data

We now apply the proposed blur scale estimation algorithm to coded aperture images captured by inserting the selected mask into a Canon 50mm $f/1.4$ lens mounted on a Canon EOS-5D DSLR as described in [5,15]. Based on the analysis

[1] The value of l_i represents the quantity of lens aperture that is open: when the lens aperture is totally open, $l_i = 1$; instead, when the mask completely blocks the light, $l_i = 0$.



GT (a) Mask $4(a)$ (b) Mask $4(b)$ (c) Mask $4(c)$ (d) Mask $4(d)$

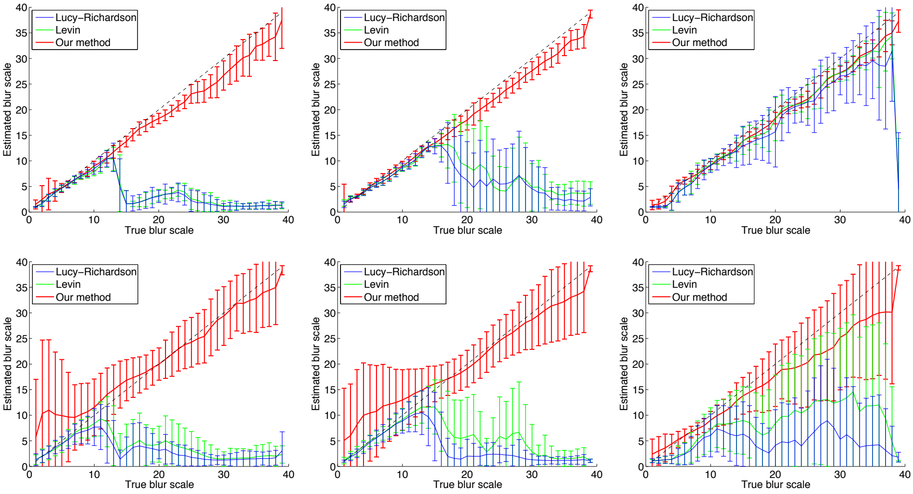
(e) Mask $4(e)$ (f) Mask $4(f)$ (g) Mask $4(g)$ (h) Mask $4(h)$

Fig. 8. Blur scale estimation - real texture. GT: Ground-truth blur scale map. (a-h) Estimated blur scale maps for all the eight masks we consider in the paper. For each mask, the figure reports the blur scale map estimated with both Levin et al.'s method (left) and our method (right).

in section 4.1 we choose mask 4(b) and mask 4(b). Each of the 4 holes in the first mask is $3.5mm$ large, which corresponds to the same overall section of a conventional (circular) aperture with diameter $7.9mm$ ($f/6.3$ in a $50mm$ lens). All indoor images have been captured by setting the shutter speed to $30ms$ (ISO 320-500) while outdoors the exposure has been set to $2ms$ or lower (ISO 100).

Firstly, we need to collect (or synthesize) a sequence of L coded images, where L is the number of blur scale levels we want to distinguish. There are two techniques to acquire these coded images: (1) If the aim is just to estimate the depth map (or blur scale map), one can capture real coded images of a planar surface with sharp natural texture (e.g., a newspaper) at different blur scale levels. (2) If the goal is to reconstruct both depth map and all-in-focus image, one has to capture the PSF of the camera at each depth level, by projecting a grid of bright dots on a plane and using a long exposure; then, coded images are simulated by applying the measured PSFs on sharp natural images collected from the web. In the experiments presented in this paper, we use the latter approach since we estimate both the blur scale map and the all-in-focus image. The PSFs have been captured on a plane at 40 different depths between 60cm and 140cm from the camera. The focal plane of the camera was set at 150cm.

In the first experiments, we show the advantage of our approach over Levin et al.'s method on a scene with blur sizes similar to the ones used in the performance



(a) Mask 4(a) (b) Mask 4(b) (c) Mask 4(d)

Fig. 9. Comparison of the estimated blur scale levels obtained from the 3 best methods using both random (top) and real (bottom) texture. Each graph reports the performance of the algorithms with (a) masks 4(a), (b) masks 4(b), and (c) mask 4(d). Both mean and standard deviation (in the graphs, we show three times the computed standard deviation) of the estimated blur scale are shown in an errorbar with the algorithms performances (solid lines) over the ideal characteristic curve (diagonal dashed line) for 39 blur sizes. Notice how the performance dramatically changes based on the nature of texture (top row vs bottom row). Moreover, in the case of real images the standard deviation of the estimates obtained with our method are more uniform for mask 4(b) than for mask 4(d). In the case of mask 4(d) the performance is reasonably accurate only with small blur scales.

test. The same dataset has been captured by using mask 4(b) (see Fig. 11) and mask 4(d) (see Fig. 12). The size of the blur, especially at the background, is very large; This can be appreciated in Fig. 10(a), which shows the same scenario captured with the same camera setting, but without mask on the lens. For a fair comparison, we do not use any regularization or user intervention to the estimated blur scale maps.

As already seen in the Section 5.1 (especially in Fig. 9), Levin et al.'s method yields an accurate blur scale estimate with mask 4(d) when the size of the blur is small, but it fails with large amounts of blur. The proposed approach overcomes this limitation and yields to a deblurred image that in both cases, Fig. 11(e) and Fig. 12(e), is closer to the ground-truth (Fig. 10(b)). Notice also that our method gives an accurate reconstruction of the blur scale, even without using regularization ($\beta = 0$ in eq. (9)). Some artefacts are still present in the reconstructed all-in-focus images. These are mainly due to the very large size of the blur and to the raw blur-scale map: When adding regularization to the blur-scale map ($\beta > 0$), the deblurring algorithm yields to better results, as one can see in the next examples.

Table 2. Random texture. Performance (mean error) of 5 algorithms in blur scale estimation and image deblurring for the apertures in Fig. 4, assuming there is not noise.

Methods	**Masks** - (image noise level $\sigma = 0$)															
	Blur scale estimation								Image deblurring							
	a	b	c	d	e	f	g	h	a	b	c	d	e	f	g	h
Lucy-Richardson	16.8	14.4	17.2	2.9	17.0	18.1	17.8	15.4	0.22	0.22	0.21	0.22	0.22	0.22	0.22	**0.21**
Regularized filtering	18.4	17.2	18.6	6.8	16.7	12.3	18.8	13.4	0.30	0.32	0.27	0.32	0.25	0.42	0.23	0.25
Wiener filtering	8.8	13.8	14.4	16.6	16.3	15.3	14.1	15.3	0.23	0.29	0.29	0.33	0.31	0.32	0.27	0.30
Levin et al.[5]	16.7	13.7	16.7	1.4	16.6	16.8	17.6	13.3	0.22	0.21	0.22	**0.21**	**0.21**	**0.22**	0.22	**0.21**
Our method	**1.2**	**0.9**	**3.7**	**0.9**	**4.2**	**10.3**	**3.8**	**9.6**	**0.20**	**0.20**	**0.21**	**0.21**	**0.21**	**0.22**	**0.21**	0.22

Table 3. Real texture. Performance (mean error) of 5 algorithms in blur scale estimation and image deblurring for the apertures in Fig. 4, assuming there is not noise.

Methods	**Masks** - (image noise level $\sigma = 0$)															
	Blur scale estimation								Image deblurring							
	a	b	c	d	e	f	g	h	a	b	c	d	e	f	g	h
Lucy-Richardson	17.0	16.4	18.4	15.6	17.9	18.5	18.0	18.3	0.22	0.20	0.22	0.18	0.20	0.20	0.20	**0.20**
Regularized filtering	18.5	16.8	18.2	8.6	16.8	11.4	17.9	15.4	0.51	0.49	0.52	1.08	0.28	0.67	0.28	0.40
Wiener filtering	17.1	16.4	18.2	14.4	17.0	18.0	17.5	17.6	0.25	0.22	0.26	0.21	0.21	0.24	0.23	0.21
Levin et al.[5]	16.3	14.8	17.9	9.9	17.0	18.2	17.6	17.0	0.25	0.21	0.23	0.19	0.20	**0.21**	0.21	**0.20**
Our method	**3.3**	**3.3**	**6.8**	**3.3**	**6.1**	**12.6**	**5.9**	**11.7**	**0.18**	**0.16**	**0.21**	**0.16**	**0.17**	**0.21**	**0.19**	0.21

In Fig. 13 we have the same indoor scenario, but now the items are slightly closer to the focal plane of the camera; then the maximum amount of blur is reduced. Although the background is still very blur in the coded image (Fig. 13(a)), our accurate blur-scale estimation yields to a deblurred image (Fig. 13(b)), where the text of the magazine becomes readable. Since the reconstructed blur-scale map corresponds to the depth map (relative depth) of the scene, we can use it together with the all-in-focus image to generate a *3D image*[2]. This image, when watched with red-cyan glasses, allows one to perceive the depth information extracted with our approach.

All the regularized blur-scale maps in this work are estimated from eq. (9) by setting $\beta = 0.5$; the *raw* maps, instead, are obtained without regularization term ($\beta = 0$).

We have tested our approach on different outdoor scenes: Fig. 15 and Fig. 14. In these scenarios we apply the subspaces we have learned within $150cm$ from the camera to a very large range of depths. Several challenges are present in these scenes, such as occlusions, shadows, and lack of texture. Our method demonstrates robustness to all of them. Notice again that the raw blur-scale maps shown in Fig. 15(c) and Fig. 14(c) are already very close to the maps that include regularization (Fig. 15(d) and Fig. 14(d) respectively). For each dataset, a

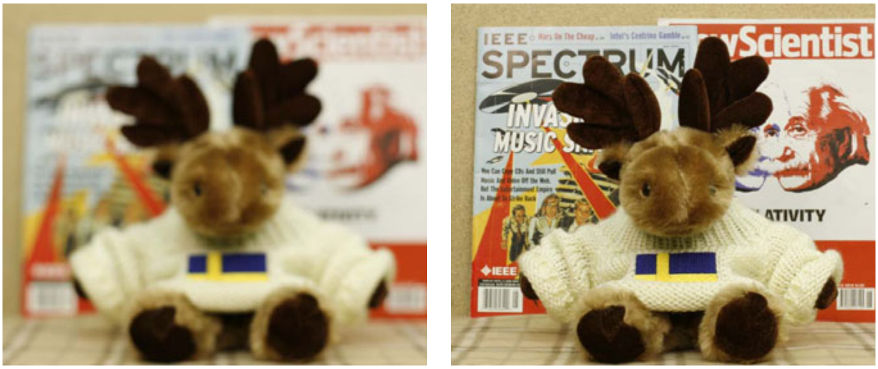
[2] In this work, a *3D image* corresponds to an image captured with a stereo camera, where one lens has a *red* filter and the second lens has a *cyan* filter. When one watches this type of images with red-cyan glasses, each eye will see only one view: The shift between the two views gives the perception of depth.

Table 4. Random texture. Performance (mean error) of 5 algorithms in blur scale estimation and image deblurring for the apertures in Fig. 4, under different levels of noise.

	Masks - (image noise level $\sigma = 0.001$)															
Methods	Blur scale estimation								Image deblurring							
	a	b	c	d	e	f	g	h	a	b	c	d	e	f	g	h
Lucy-Richardson	18.5	17.1	18.2	11.7	16.6	16.2	18.3	17.3	0.39	0.36	0.27	0.28	0.35	0.29	0.26	0.27
Regularized filtering	19.0	17.5	19.0	14.3	16.8	18.3	18.9	15.6	0.88	0.96	0.61	1.03	0.93	0.61	0.61	0.91
Wiener filtering	15.7	16.7	16.8	17.5	17.2	17.6	16.8	17.0	0.35	0.37	0.36	0.39	0.38	0.38	0.35	0.38
Levin et al.	18.4	16.3	18.1	11.0	16.7	17.3	18.3	17.5	0.32	0.31	0.26	0.28	0.30	0.28	0.25	0.26
Our method	**9.6**	**8.7**	**12.7**	**10.1**	**12.5**	**13.2**	**12.9**	**13.9**	**0.20**	**0.21**	**0.22**	**0.22**	**0.21**	**0.23**	**0.21**	**0.23**
Methods	**Masks** - (image noise level $\sigma = 0.002$)															
	Blur scale estimation								Image deblurring							
	a	b	c	d	e	f	g	h	a	b	c	d	e	f	g	h
Lucy-Richardson	18.5	17.1	18.2	12.1	16.6	16.3	18.3	17.3	0.49	0.46	0.31	0.34	0.44	0.33	0.30	0.32
Regularized filtering	18.9	17.4	18.8	12.7	16.7	16.9	18.9	16.9	0.76	0.69	0.47	0.50	0.67	0.46	0.49	0.46
Wiener filtering	15.5	16.4	16.7	17.3	17.1	17.5	16.8	17.0	0.35	0.37	0.37	0.39	0.38	0.39	0.35	0.38
Levin et al.	18.5	16.9	18.0	12.1	16.7	17.6	18.4	17.7	0.39	0.38	0.29	0.34	0.37	0.31	0.28	0.29
Our method	**11.3**	**11.1**	**13.2**	**11.3**	**12.6**	**13.5**	**12.8**	**14.0**	**0.22**	**0.22**	**0.23**	**0.23**	**0.22**	**0.23**	**0.23**	**0.24**
Methods	**Masks** - (image noise level $\sigma = 0.005$)															
	Blur scale estimation								Image deblurring							
	a	b	c	d	e	f	g	h	a	b	c	d	e	f	g	h
Lucy-Richardson	18.4	17.0	18.2	12.6	16.5	16.6	18.4	17.3	0.66	0.62	0.41	0.47	0.61	0.40	0.40	0.43
Regularized filtering	18.9	17.4	18.8	13.1	16.6	17.1	18.8	16.9	1.17	1.04	0.69	0.75	1.03	0.59	0.73	0.68
Wiener filtering	15.4	16.2	16.5	17.3	17.2	17.3	16.7	17.0	0.35	0.37	0.37	0.39	0.38	0.39	0.35	0.38
Levin et al.	18.5	16.9	18.0	12.5	16.7	17.7	18.4	17.7	0.55	0.54	0.37	0.45	0.51	0.37	0.36	0.39
Our method	**12.8**	**12.6**	**13.4**	**12.0**	**12.8**	**13.5**	**13.5**	**14.0**	**0.25**	**0.25**	**0.26**	**0.25**	**0.25**	**0.26**	**0.26**	**0.27**

Table 5. Real texture. Performance (mean error) of 5 algorithms in blur scale estimation and image deblurring for the apertures in Fig. 4, under different levels of noise.

Methods	**Masks** - (image noise level $\sigma = 0.001$)															
	Blur scale estimation								Image deblurring							
	a	b	c	d	e	f	g	h	a	b	c	d	e	f	g	h
Lucy-Richardson	18.5	17.2	18.3	13.7	16.8	17.8	18.4	18.1	0.38	0.35	0.26	0.24	0.32	0.23	0.24	0.25
Regularized filtering	19.0	17.5	19.0	14.0	16.8	17.6	19.0	15.6	0.96	1.05	0.66	1.39	0.94	0.68	0.64	1.02
Wiener filtering	13.8	14.5	14.1	14.6	15.2	14.4	14.8	14.5	0.21	0.23	0.22	0.22	0.23	0.21	0.21	0.23
Levin et al.[5]	18.4	16.8	18.1	10.6	16.7	17.0	18.2	17.8	0.34	0.33	0.27	0.30	0.30	0.27	0.24	0.25
Our method	**8.7**	**7.8**	**11.8**	**7.7**	**11.9**	**13.5**	**11.5**	**13.8**	**0.21**	**0.18**	**0.22**	**0.17**	**0.19**	**0.20**	**0.20**	**0.20**
Methods	**Masks** - (image noise level $\sigma = 0.002$)															
	Blur scale estimation								Image deblurring							
	a	b	c	d	e	f	g	h	a	b	c	d	e	f	g	h
Lucy-Richardson	18.5	17.2	18.3	13.2	16.7	17.5	18.4	17.9	0.47	0.44	0.30	0.29	0.40	0.26	0.27	0.30
Regularized filtering	19.0	17.5	19.0	14.1	16.8	18.1	19.0	15.7	1.26	1.38	0.87	1.72	1.30	0.74	0.87	1.34
Wiener filtering	14.7	15.8	15.2	15.8	16.0	15.1	15.0	15.7	**0.23**	0.25	0.24	0.24	0.25	0.24	0.22	0.25
Levin et al.	18.4	16.8	18.1	11.1	16.7	17.1	18.3	17.7	0.41	0.40	0.30	0.37	0.37	0.30	0.28	0.29
Our method	**10.6**	**9.5**	**12.1**	**9.0**	**12.3**	**13.5**	**12.1**	**14.1**	0.24	**0.19**	**0.23**	**0.17**	**0.19**	**0.20**	**0.21**	**0.20**
Methods	**Masks** - (image noise level $\sigma = 0.005$)															
	Blur scale estimation								Image deblurring							
	a	b	c	d	e	f	g	h	a	b	c	d	e	f	g	h
Lucy-Richardson	18.3	17.1	18.2	12.9	16.6	17.4	18.4	17.9	0.61	0.58	0.39	0.40	0.55	0.34	0.37	0.40
Regularized filtering	19.0	17.5	19.0	14.1	16.8	18.1	18.9	15.7	1.89	2.07	1.31	2.38	2.03	0.88	1.31	2.02
Wiener filtering	15.6	16.5	16.1	16.8	16.7	16.5	16.0	16.7	**0.26**	0.27	0.26	0.27	0.27	0.26	0.24	0.26
Levin et al.	18.5	16.9	18.1	11.3	16.7	17.4	18.4	17.7	0.56	0.55	0.38	0.49	0.51	0.37	0.35	0.39
Our method	**12.2**	**11.8**	**13.3**	**10.8**	**12.7**	**13.7**	**13.4**	**13.7**	**0.26**	**0.22**	**0.24**	**0.19**	**0.21**	**0.22**	**0.22**	**0.25**



(a) Conventional aperture

(b) Ground-truth (pinhole camera)

Fig. 10. (a) Picture taken with the conventional camera without placing the mask on the lens. (b)Image captured by simulating a pinhole camera (f/22.0), which can be used as ground-truth for the image texture.

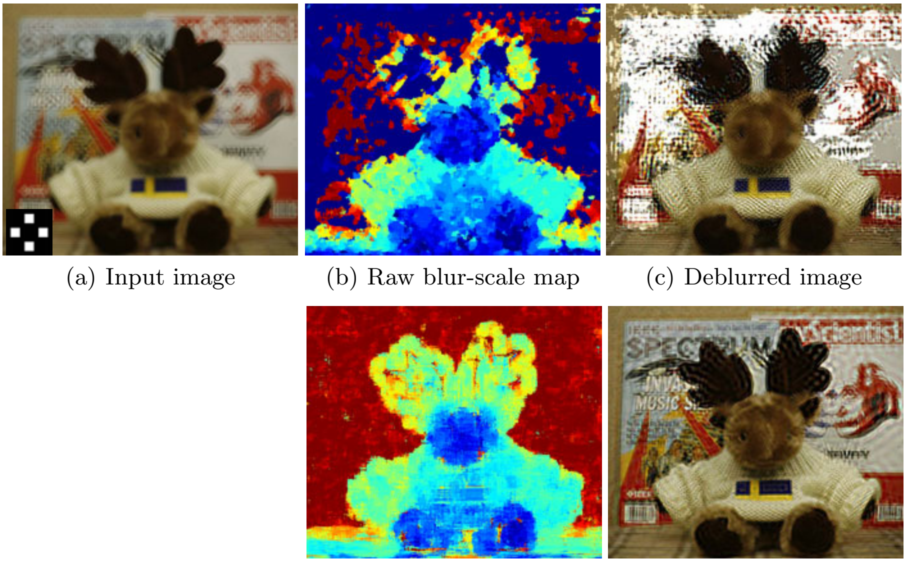
3D image (Fig. 14(e) and Fig. 15(e)) has been generated by using just the output of our method: the deblurred images (b) and the blur-scale maps (d). The ground-truth images have been taken by simulating a pinhole camera (f/22.0).

5.3 Computational Cost

We downsample 4 times the input images from an original resolution of 12,8 megapixel ($4,368 \times 2,912$) and use sub-pixel accuracy, in order to keep the algorithm efficient. We have seen from experiments on real data that the raw blur-scale map is already very close to the regularized map. This means that we can obtain a reasonable blur scale map very efficiently: When $\beta = 0$ the value of the blur scale at one pixel is independent of the other pixels and the calculations can be carried out in parallel. Since the algorithm takes about 5ms for processing 40 blur scale levels at each pixel, it is suitable for real-time applications. We have run the algorithm on a QuadCore 2.8GHz with 16GB memory. The code has been written mainly in Matlab 7. The deblurring procedure, instead, takes about 100s to process the whole image for 40 blur scale levels.

6 Conclusions

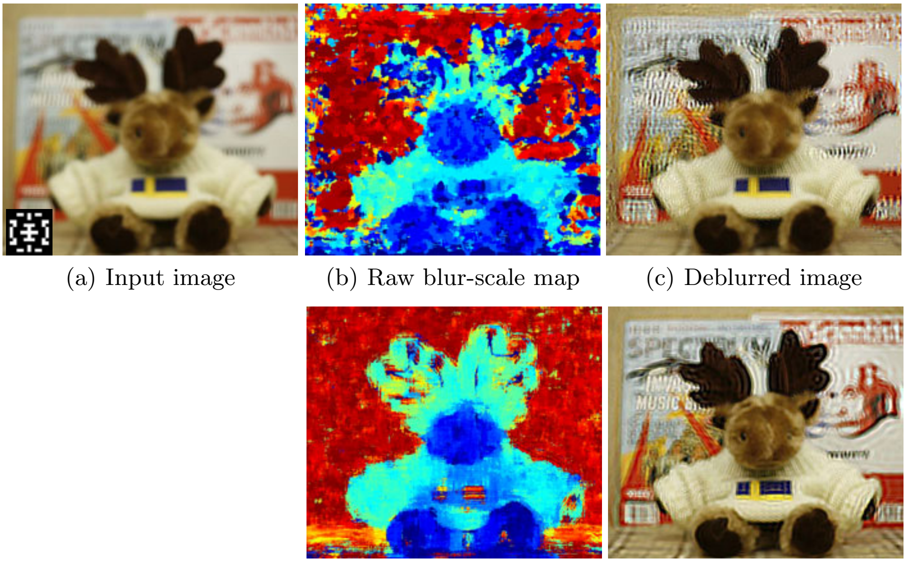
We have presented a novel method to recover the all-in-focus image from a single blurred image captured with a coded aperture camera. The method is split in two steps: A subspace-based blur scale identification approach and an image deblurring algorithm based on conjugate gradient descent. The method is simple, general, and computationally efficient. We have compared our method to existing algorithms in the literature and showed that we achieve state of the art performance in blur scale identification and image deblurring with both synthetic and real data while retaining polynomial time complexity.



(a) Input image (b) Raw blur-scale map (c) Deblurred image

(d) Raw blur-scale map (e) Deblurred image

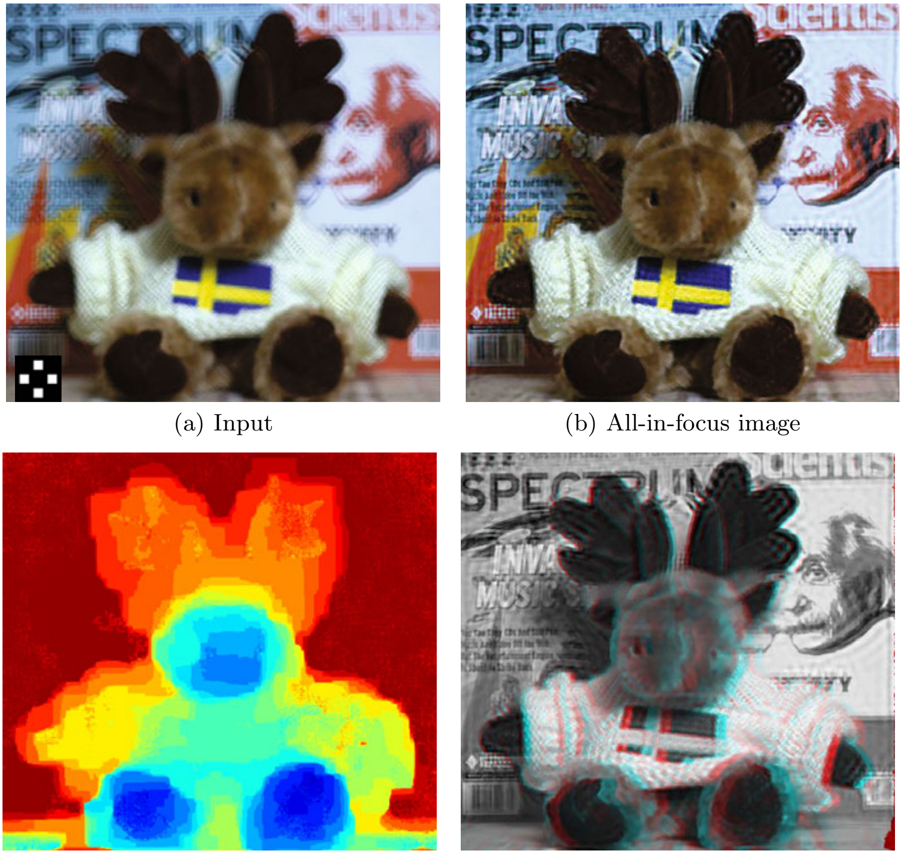
Fig. 11. Comparison on real data - mask 4(b). (a) Input image captured by using mask 4(b). (b-c) Blur-scale map and all-in-focus image reconstructed with Levins et al.'s method [5]; (d-e) Results obtained from our method.



(a) Input image (b) Raw blur-scale map (c) Deblurred image

(d) Raw blur-scale map (e) Deblurred image

Fig. 12. Comparison on real data - mask 4(d). (a) Input image captured by using mask 4(d). (b-c) Blur-scale map and all-in-focus image reconstructed with Levins et al.'s method [5]; (d-e) Results obtained from our method.



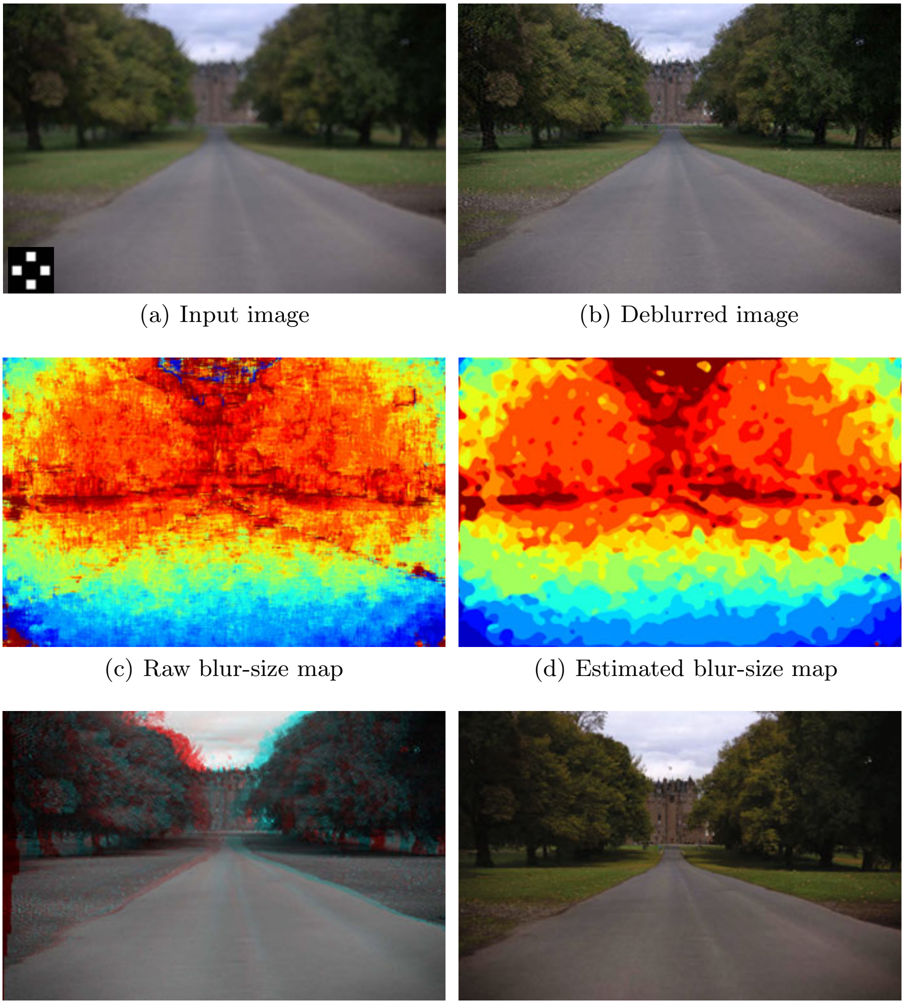
(a) Input

(b) All-in-focus image

(c) Blur-scale map

(d) 3D image

Fig. 13. Close-range indoor scene [exposure time: 1/30s]. (a) coded image captured with mask 4(b); (b) estimated all-in-focus image; (c) estimated blur-scale map; (d) 3D image (to be watched with red-cyan glasses).


(a) Input image

(b) Deblurred image

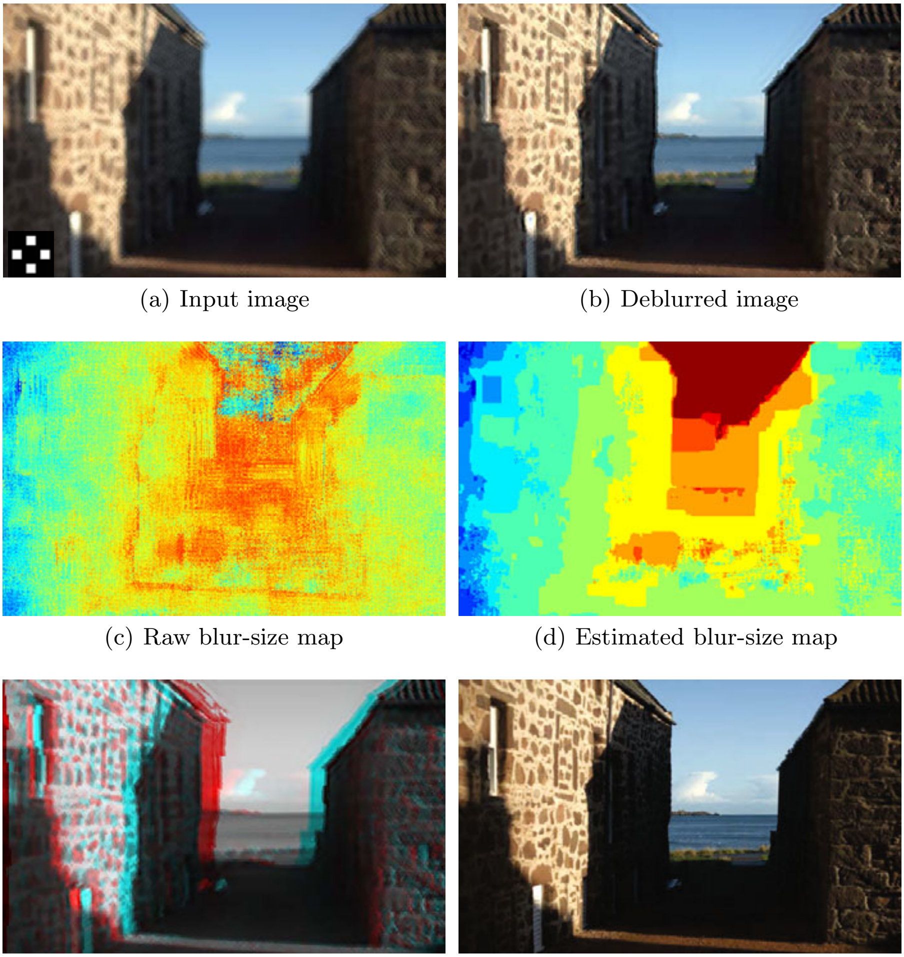
(c) Raw blur-size map

(d) Estimated blur-size map

(e) 3D image

(f) Ground-truth image

Fig. 14. Long-range outdoor scene [exposure time: 1/200s]. (a) coded image captured with mask 4(b); (b) estimated all-in-focus image; (c) raw blur-scale map (without regularization); (d) regularized blur-scale map; (e) 3D image (to be watched with red-cyan glasses); (f) ground-truth image.



(a) Input image

(b) Deblurred image

(c) Raw blur-size map

(d) Estimated blur-size map

(e) 3D image

(f) Ground-truth image

Fig. 15. Mid-range outdoor scene [exposure time: 1/200s]. (a) coded image captured with mask 4(b); (b) estimated all-in-focus image; (c) raw blur-scale map (without regularization); (d) regularized blur-scale map; (e) 3D image (to be watched with red-cyan glasses); (f) ground-truth image.

Appendix

Proof of Theorem 1

To prove the theorem we rewrite the least squares problem in $\boldsymbol{f}$ as

$$\|\boldsymbol{H_d f} - \boldsymbol{g}\|_2^2 + \alpha\|\boldsymbol{\Sigma f}\|_2^2 = \left\| \begin{bmatrix} \boldsymbol{H_d} \\ \sqrt{\alpha}\boldsymbol{\Sigma} \end{bmatrix} \boldsymbol{f} - \begin{bmatrix} \boldsymbol{g} \\ 0 \end{bmatrix} \right\|_2^2 = \|\bar{\boldsymbol{H}}_{\boldsymbol{d}}\boldsymbol{f} - \bar{\boldsymbol{g}}\|_2^2 \tag{17}$$

where we have defined $\bar{\boldsymbol{H}}_{\boldsymbol{d}} = \left[\boldsymbol{H}_{\boldsymbol{d}}^T \; \sqrt{\alpha}\boldsymbol{\Sigma}^T\right]^T$ and $\bar{\boldsymbol{g}} = \left[\boldsymbol{g}^T \; 0^T\right]^T$. Then, we can define the solution in $\boldsymbol{f}$ as $\hat{\boldsymbol{f}} = \left(\bar{\boldsymbol{H}}_{\boldsymbol{d}}^T \bar{\boldsymbol{H}}_{\boldsymbol{d}}\right)^{-1} \bar{\boldsymbol{H}}_{\boldsymbol{d}}^T \bar{\boldsymbol{g}}$. By substituting the solution for $\boldsymbol{f}$ back in the least squares problem, we obtain

$$\|\boldsymbol{H_d f} - \boldsymbol{g}\|_2^2 + \alpha\|\boldsymbol{\Sigma f}\|_2^2 = \|\bar{\boldsymbol{H}}_{\boldsymbol{d}}^{\perp}\bar{\boldsymbol{g}}\|_2^2 \tag{18}$$

where $\bar{\boldsymbol{H}}_{\boldsymbol{d}}^{\perp} = \boldsymbol{I} - \bar{\boldsymbol{H}}_{\boldsymbol{d}} \left(\bar{\boldsymbol{H}}_{\boldsymbol{d}}^T \bar{\boldsymbol{H}}_{\boldsymbol{d}}\right)^{-1} \bar{\boldsymbol{H}}_{\boldsymbol{d}}^T$.

We have shown that we can use $\bar{\boldsymbol{H}}_{\boldsymbol{d}}^{\perp}$ rather than $\boldsymbol{H}_{\boldsymbol{d}}^{\perp}$ and $\bar{\boldsymbol{g}}$ rather than $\boldsymbol{g}$ in the minimization problem (5) without affecting the solution. The rest of the proof then assumes that the energy in eq. (5) is based on $\|\bar{\boldsymbol{H}}_{\boldsymbol{d}}^{\perp}\bar{\boldsymbol{g}}\|_2^2$. The step above is necessary to fully exploit the properties of $\bar{\boldsymbol{H}}_{\boldsymbol{d}}^{\perp}$. $\bar{\boldsymbol{H}}_{\boldsymbol{d}}^{\perp}$ is a *symmetric* matrix (i.e, $(\bar{\boldsymbol{H}}_{\boldsymbol{d}}^{\perp})^T = \bar{\boldsymbol{H}}_{\boldsymbol{d}}^{\perp}$) and is also *idempotent* (i.e, $\bar{\boldsymbol{H}}_{\boldsymbol{d}}^{\perp} = (\bar{\boldsymbol{H}}_{\boldsymbol{d}}^{\perp})^2$). By applying the above properties we can write the argument of the first term of the cost in eq. (5) as

$$\bar{\boldsymbol{g}}^T \bar{\boldsymbol{H}}_{\boldsymbol{d}}^{\perp} \bar{\boldsymbol{g}} = \bar{\boldsymbol{g}}^T (\bar{\boldsymbol{H}}_{\boldsymbol{d}}^{\perp})^T \bar{\boldsymbol{H}}_{\boldsymbol{d}}^{\perp} \bar{\boldsymbol{g}} = ||\bar{\boldsymbol{H}}_{\boldsymbol{d}}^{\perp}\bar{\boldsymbol{g}}||^2 \tag{19}$$

Moreover, from the definition of $\bar{\boldsymbol{H}}_{\boldsymbol{d}}^{\perp}$ we know that

$$\begin{aligned} \bar{\boldsymbol{H}}_{\boldsymbol{d}}^{\perp} &\doteq \boldsymbol{I} - \bar{\boldsymbol{H}}_{\boldsymbol{d}} (\bar{\boldsymbol{H}}_{\boldsymbol{d}}^T \bar{\boldsymbol{H}}_{\boldsymbol{d}})^{-1} \bar{\boldsymbol{H}}_{\boldsymbol{d}}^T \\ &= \boldsymbol{I} - \bar{\boldsymbol{H}}_{\boldsymbol{d}} \bar{\boldsymbol{H}}_{\boldsymbol{d}}^{\dagger} \end{aligned} \tag{20}$$

Thus, the necessary conditions for an extremum of eq. (5) become

$$\begin{cases} \left(\bar{\boldsymbol{g}} - \bar{\boldsymbol{H}}_{\boldsymbol{d}}\bar{\boldsymbol{H}}_{\boldsymbol{d}}^{\dagger}\bar{\boldsymbol{g}}\right)^T \left(\nabla\bar{\boldsymbol{H}}_{\boldsymbol{d}}\bar{\boldsymbol{H}}_{\boldsymbol{d}}^{\dagger} + \bar{\boldsymbol{H}}_{\boldsymbol{d}}\nabla\bar{\boldsymbol{H}}_{\boldsymbol{d}}^{\dagger}\right)\bar{\boldsymbol{g}} = \nabla \cdot \dfrac{\nabla \boldsymbol{d}}{\|\nabla \boldsymbol{d}\|_1} \\ \boldsymbol{f} = \bar{\boldsymbol{H}}_{\boldsymbol{d}}^{\dagger}\bar{\boldsymbol{g}}. \end{cases} \tag{21}$$

where $\nabla\bar{\boldsymbol{H}}_{\boldsymbol{d}}$ is the gradient of $\bar{\boldsymbol{H}}_{\boldsymbol{d}}$ with respect to $\boldsymbol{d}$, and the right hand side of the first equation is the gradient of $\|\nabla \boldsymbol{d}\|_1$ with respect to $\boldsymbol{d}$. Similarly, the necessary conditions for eq. (4) are

$$\begin{cases} \left(\bar{\boldsymbol{g}} - \bar{\boldsymbol{H}}_{\boldsymbol{d}}\boldsymbol{f}\right)^T \nabla\bar{\boldsymbol{H}}_{\boldsymbol{d}}\boldsymbol{f} = \nabla \cdot \dfrac{\nabla \boldsymbol{d}}{\|\nabla \boldsymbol{d}\|_1} \\ \bar{\boldsymbol{H}}_{\boldsymbol{d}}^T \left(\bar{\boldsymbol{g}} - \bar{\boldsymbol{H}}_{\boldsymbol{d}}\boldsymbol{f}\right) = 0. \end{cases} \tag{22}$$

It is now immediate to apply the same derivation as in [46] and demonstrate that the left hand side of the first equation in both system (22) and system (21) are identical. Since the right hand sides are also identical, this implies that the first equations have the same solutions. The second equations in (22) and (21) are instead identical by construction.

References

1. Jones, D., Lamb, D.: Analyzing the visual echo: Passive 3-d imaging with a multiple aperture camera. Technical report, McGill University (1993)
2. Dowski, E.R., Cathey, T.W.: Extended depth of field through wave-front coding. Applied Optics 34, 1859–1866 (1995)
3. Farid, H.: Range Estimation by Optical Differentiation. PhD thesis, University of Pennsylvania (1997)
4. Veeraraghavan, A., Raskar, R., Agrawal, A., Mohan, A., Tumblin, J.: Dappled photography: mask enhanced cameras for heterodyned light fields and coded aperture refocusing. ACM Trans. Graph 26, 69 (2007)
5. Levin, A., Fergus, R., Durand, F., Freeman, W.T.: Image and depth from a conventional camera with a coded aperture. ACM Trans. Graph 26, 70 (2007)
6. Bishop, T., Zanetti, S., Favaro, P.: Light field superresolution. In: ICCP (2009)
7. Cossairt, O., Nayar, S.: Spectral focal sweep: Extended depth of field from chromatic aberrations. In: ICCP (2010)
8. Liang, C.K., Lin, T.H., Wong, B.Y., Liu, C., Chen, H.: Programmable aperture photography: Multiplexed light field acquisition. ACM Trans. Graph 27, 55:1–55:10 (2008)
9. Ng, R., Levoy, M., Brédif, M., Duval, G., Horowitz, M., Hanrahan, P.: Light field photography with a hand-held plenoptic camera. Technical Report CSTR 2005-02, Stanford University CS (2005)
10. Gottesman, S.R., Fenimore, E.E.: New family of binary arrays for coded aperture imaging. Applied Optics 28, 4344–4352 (1989)
11. Zomet, A., Nayar, S.K.: Lensless imaging with a controllable aperture. In: CVPR, vol. 1, pp. 339–346 (2006)
12. Raskar, R., Agrawal, A.K., Tumblin, J.: Coded exposure photography: Motion deblurring using fluttered shutter. ACM Trans. Graph 25, 795–804 (2006)
13. Hiura, S., Matsuyama, T.: Depth measurement by the multi-focus camera. In: CVPR, vol. 2, pp. 953–961 (1998)
14. Zhou, C., Lin, S., Nayar, S.K.: Coded aperture pairs for depth from defocus. In: ICCV (2009)
15. Zhou, C., Nayar, S.: What are good apertures for defocus deblurring? In: IEEE ICCP (2009)
16. Levin, A., Hasinoff, S., Green, P., Durand, F., Freeman., W.T.: 4d frequency analysis of computational cameras for depth of field extension. ACM Trans. Graph 28 (2009)
17. McLean, D.: The improvement of images obtained with annular apertures. Royal Society of London 263, 545–551 (1961)
18. Greengard, A., Schechner, Y.Y., Piestun, R.: Depth from diffracted rotation. Optics Letters 31, 181–183 (2006)
19. Dowski, E.R., Cathey, T.W.: Single-lens single-image incoherent passive-ranging systems. Applied Optics 33, 6762–6773 (1994)
20. Johnson, G.E., Dowski, E.R., Cathey, W.T.: Passive ranging through wave-front coding: Information and application. Applied Optics 39, 1700–1710 (2000)
21. Cossairt, O., Zhou, C., Nayar, S.K.: Diffusion coding photography for extended depth of field. ACM Trans. Graph (2010)
22. Dou, Q., Favaro, P.: Off-axis aperture camera: 3d shape reconstruction and image restoration. In: CVPR (2008)

23. Georgiev, T., Zheng, K., Curless, B., Salesin, D., Nayar, S., Intawala, C.: Spatio-angular resolution tradeoffs in integral photography. In: Eurographics Workshop on Rendering, pp. 263–272 (2006)
24. Levoy, M., Ng, R., Adams, A., Footer, M., Horowitz, M.: Light field microscopy. ACM Trans. Graph 25, 924–934 (2006)
25. Fergus, R., Singh, B., Hertzmann, A., Roweis, S., Freeman, W.: Removing camera shake from a single photograph. ACM Trans. Graph 25, 787–794 (2006)
26. Shan, Q., Jia, J., Agarwala, A.: High-quality motion deblurring from a single image. ACM Trans. Graph (2008)
27. Levin, A., Weiss, Y., Durand, F., Freeman., W.T.: Understanding and evaluating blind deconvolution algorithms. In: CVPR, pp. 1964–1971 (2009)
28. Cho, S., Lee, S.: Fast motion deblurring. Siggraph Asia 28 (2009)
29. Xu, L., Jia, J.: Two-phase kernel estimation for robust motion deblurring. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6311, pp. 157–170. Springer, Heidelberg (2010)
30. Shan, Q., Xiong, W., Jia, J.: Rotational motion deblurring of a rigid object from a single image. In: ICCV, pp. 1–8 (2007)
31. Whyte, O., Sivic, J.: Zisserman, A., Ponce, J.: Non-uniform deblurring for shaken images. In: CVPR, pp. 491–498 (2010)
32. Gupta, A., Joshi, N., Zitnick, C., Cohen, M., Curless, B.: Single image deblurring using motion density functions. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010. LNCS, vol. 6311, pp. 171–184. Springer, Heidelberg (2010)
33. Nishiyama, M., Hadid, A., Takeshima, H., Shotton, J., Kozakaya, T., Yamaguchi, O.: Facial deblur inference using subspace analysis for recognition of blurred faces. IEEE T.PAMI 33, 1–8 (2011)
34. Pouli, T., Cunningham, D.W., Reinhard, E.: Image statistics and their applications in computer graphics. Eurographics, State of the Art Report (2010)
35. Ruderman, D.L.: The statistics of natural images. Network: Computation in Neural Systems 5, 517–548 (1994)
36. Huang, J., Mumford, D.: Statistics of natural images and models. In: CVPR, vol. 1, pp. 1541–1548 (1999)
37. Huang, J., Lee, A., Mumford, D.: Statistics of range images. In: CVPR, pp. 324–331 (2000)
38. Rudin, L., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D 60, 259–268 (1992)
39. Kolmogorov, V., Zabih, R.: Multi-camera scene reconstruction via graph cuts. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2352, pp. 82–96. Springer, Heidelberg (2002)
40. Press, W., Flannery, B., Teukolsky, S., Vetterling, W.: Numerical Recipes in C. Cambridge University Press, Cambridge (1988)
41. Sun, X., Cheng, Q.: On subspace distance. In: Image Analysis and Recognition, pp. 81–89 (2006)
42. Premaratne, P., Ko, C.: Zero sheet separation of blurred images with symmetrical point spread functions. Signals, Systems, and Computers, 1297–1299 (1999)
43. Martinello, M., Bishop, T.E., Favaro, P.: A bayesian approach to shape from coded aperture. In: ICIP (2010)
44. Snyder, D., Schulz, T., O'Sullivan, J.: Deblurring subject to nonnegativity constraints. IEEE Trans. on Signal Processing 40(5), 1143–1150 (1992)
45. Bertero, M., Boccacci, P.: Introduction to inverse problems in imaging. Institute of Physics Publishing, Bristol (1998)
46. Favaro, P., Soatto, S.: A geometric approach to shape from defocus. TPAMI 27, 406–417 (2005)