Recovering Thin Structures via Nonlocal-Means Regularization with Application to Depth from Defocus

Paolo Favaro
Heriot-Watt University, Edinburgh, UK
p.favaro@hw.ac.uk

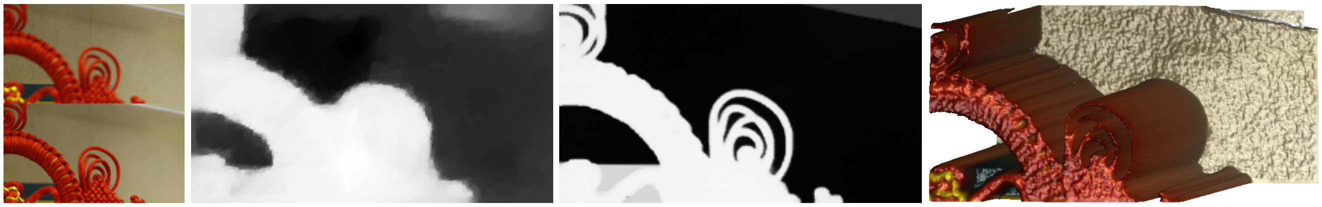

Figure 1. From left to right: Detail of two 150×274 pixels defocused images; depth map recovered without and with nonlocal-means regularization; 3D rendering with texture mapping of the depth map recovered with the proposed algorithm.

Abstract

We propose a novel scheme to recover depth maps containing thin structures based on nonlocal-means filtering regularization. The scheme imposes a distributed smoothness constraint by relying on the assumption that pixels with similar colors are likely to belong to the same surface, and therefore can be used jointly to obtain a robust estimate of their depth. This scheme can be used to solve shape-from-X problems and we demonstrate its use in the case of depth from defocus. We cast the problem in a variational framework and solve it by linearizing the corresponding Euler-Lagrange equations. The linearized system is then inverted by using efficient numerical methods such as successive overrelaxations or more general methods such as conjugate gradient when the system is not diagonally dominant. One of the main benefits of this formulation is that it can handle the regularization of highly fragmented surfaces, which require large neighborhood structures typically difficult to solve efficiently with graph-based methods. We compare the performance of the proposed algorithm with methods recently proposed in the literature that are analogous to neighborhood filters. Finally, experimental results are shown on synthetic and real data.

1. Introduction

In this paper we focus on the task of recovering 3D surfaces from images with particular attention to thin structures and accurate contour estimation. The main challenge in dealing with thin structures is that corresponding pixels across the input images lie in an unknown elongated domain. On the one hand pixel-based correspondence is extremely unreliable and results in noisy estimates; on the other hand, region-based correspondence results in wider object contours that may completely wipe out thin structures (see Figure 1). We propose to use a novel class of nonparametric surface smoothness priors based on the nonlocal-means framework and show that very accurate contours and thin structures can be recovered.

Our method is based on the principles of neighborhood filtering, and in particular nonlocal-means filtering, which has been successfully applied to image restoration and denoising [6]. In neighborhood filtering pixels that share similar colors are averaged together to remove noise. One of the most important properties of these filters is that they accurately preserve edges and texture unlike Gaussian blur denoising. This suggests that one could use a neighborhood filter strategy for recovering thin structures and define a regularization term to penalize depth discontinuities by averaging corresponding pixels. Such strategy however, establishes correspondences by using pixel by pixel comparisons, which might not be always reliable especially in the presence of noise. In our approach instead we propose to determine correspondences by using region-based comparisons, as done in nonlocal-means filtering. Using extended regions however, might lead to finding few good correspondences in the case of thin structures. Therefore, to collect

978-1-4244-6985-7/10/$26.00 ©2010 IEEE

as many valid correspondences as possible we do not use the common square or circular regions, but rather regions with elongated shapes (ellipsoids) and test them for a finite set of directions. The resulting algorithm finds large sets of reliable correspondences between pixels.

While large sets of correspondences provide a useful smoothness constraint, they also render the 3D surface estimation task quite challenging. One approach is to simplify the correspondences and to keep only the most relevant ones in a discrete graph [20]. This however, results in smoothness terms that are too weak at thin structures (see section 2.2). Hence, we propose a solution that keeps all the correspondences at all times. We formulate the 3D surface estimation task as the minimization of a cost functional in the continuous domain. Necessary conditions for a minimum can be written in the form of Euler-Lagrange equations which we solve with an iterative linearization [4]. The resulting linear system is then inverted efficiently by employing successive overrelaxations [28].

We illustrate this novel regularization scheme in the case of depth from defocus, where one exploits changes in the lens settings of finite aperture cameras [18, 11, 9, 13, 25], although other shape-from-X problems could be used as well. Notice that although we impose a smoothness constraint where depths of pixels with similar colors are averaged, additional regularization is needed at pixels without correspondence to make the depth estimation problem well-posed. We employ total variation [7, 4, 28] which tends to favor piecewise constant functions, and, therefore, yields smooth surfaces while allowing for sharp discontinuities.

Contributions

1. We introduce a novel depth smoothness constraint based on nonlocal-means filtering, *i.e.*, pixels whose intensities match within windows should share similar depth values; unlike previous approaches, the proposed nonlocal-means works on thin structures by using directional elongated windows;

2. In contrast to [20], one of the top-performing algorithms in stereopsis that retains only a sparse set of dominant matches established by pixel-to-pixel comparisons, we propose a numerically efficient method based on iterated linearization that uses all correspondences;

3. We demonstrate the proposed strategy in depth from defocus and obtain performances that compare favorably with the state-of-the-art.

2. Regularized Shape Estimation

The family of problems that we consider take the form of an energy minimization with three terms: a data fidelity term, a depth smoothness regularization term, and a neighborhood regularization term:

$$\hat{s} = \arg\min_s E[s] \doteq \arg\min_s E_{data}[s] + \alpha E_{tv}[s] + \beta E_n[s]. \tag{1}$$

where s denotes the unknown depth (or disparity) map and α and β are two positive constants. In this paper we consider the data fidelity term $E_{data}[s]$ given in the case of depth from defocus (see section 3) and pay more attention to the regularization terms. In particular, we consider isotropic total variation (or its regularized version) so that solutions are constrained to be piecewise constant [8]

$$E_{tv}[s] = \int \|\nabla s(\mathbf{y})\| \, d\mathbf{y}. \tag{2}$$

Implementation details of E_{tv} will be explained in section 4. This term alone tends to yield sharp boundaries whose support is broader than the true one. Furthermore, it tends to remove thin surfaces (see Figure 1). To contrast this behavior we design a neighborhood regularization term E_n. As mentioned in the previous sections, the main idea is to link the depth values of pixels sharing similar color (or texture). In the next sections we show how to do so by using neighborhood filtering.

2.1. Pixel Similarity and Neighborhood Filtering

The idea of correlating pixels with similar color or texture has been shown to be particularly effective in preserving accurate edges in stereopsis [3, 14, 23, 17, 20] as well as image denoising [6, 27, 24]. In the case of thin structures this strategy is essential. The computation of the energy terms in eq. (1) requires combining values at multiple pixels. If these pixels do not belong to the same surface then values obtained from their combination might be highly incorrect. For this reason the piecewise smoothness energy term E_{tv} tends to misplace the edge location and to blend background with foreground at thin surfaces (see Figure 1). In this section we briefly review and analyze how neighborhood filtering methods establish pixel correspondence so that we can devise a sensible strategy for thin structures. A detailed account on neighborhood methods can be found in [6].

The neighborhood and nonlocal-means filters are extremely effective in removing noise from images while preserving edges and texture structure. These filters satisfy the *noise to noise* principle, *i.e.*, when given white noise as input they return the white noise, are statistically optimal (for a given noise model), and can yield (with proper tuning) unstructured method noise. We begin with the simplest filtering method that one can consider: Gaussian blur. Its filtering strategy applies to most image filtering operations (*e.g.*, gradients), where pixel similarity is entirely based on how close two pixels are in the spatial domain. Given a noisy image I, the Gaussian blur filter returns

$$\hat{I}(\mathbf{y}) = \frac{1}{\pi\tau^2}\int_{\Omega\subset\mathbb{R}^2} e^{-\frac{|\mathbf{y}-\mathbf{x}|^2}{\tau}} I(\mathbf{x})d\mathbf{x} \qquad (3)$$

where τ is a bandwidth parameter determining the size of the spatial filter. This filter averages together pixels that might not be related to each other, thus resulting in blurred edges. A better method is a technique called sigma-filter [27, 15], where averaging is done only between pixels with the same color

$$\hat{I}(\mathbf{y}) = \frac{1}{C(\mathbf{y})}\int_{B(\mathbf{x})} e^{-\frac{|I(\mathbf{y})-I(\mathbf{x})|^2}{\tau}} I(\mathbf{x})d\mathbf{x} \qquad (4)$$

and $C(\mathbf{y})$ is the normalization factor. Similarly, the bilateral filter [24] and SUSAN [21] combine the above two filters to obtain a localized sigma-filter

$$\hat{I}(\mathbf{y}) = \frac{1}{C(\mathbf{y})}\int_{\Omega} e^{-\frac{|\mathbf{y}-\mathbf{x}|^2}{\tau_1}} e^{-\frac{|I(\mathbf{y})-I(\mathbf{x})|^2}{\tau_2}} I(\mathbf{x})d\mathbf{x} \qquad (5)$$

where again $C(\mathbf{y})$ is the normalization factor and τ_1 and τ_2 are the bandwidth parameters in the spatial and color domains respectively. These filters however, create irregularities at edges and leave some residual noise in uniform regions [6]. Such artifacts are due to the pixel-based matching that might be susceptible to noise. To be less sensitive to noise one could use region-based matching as in the *nonlocal-means* filter

$$\hat{I}(\mathbf{y}) = \frac{1}{C(\mathbf{y})}\int_{\Omega} e^{-\frac{G_\rho * |I(\mathbf{y})-I(\mathbf{x})|^2(0)}{\tau}} I(\mathbf{x})d\mathbf{x} \qquad (6)$$

where G is an isotropic Gaussian kernel with variance ρ such that

$$G_\rho * |I(\mathbf{y})-I(\mathbf{x})|^2(0) = \int_{\mathbb{R}^2} G_\rho(\mathbf{t})|I(\mathbf{y}+\mathbf{t})-I(\mathbf{x}+\mathbf{t})|^2 d\mathbf{t}. \qquad (7)$$

2.2. Directional Nonlocal-Means Regularization

The nonlocal-means filter allows one to establish reliable correspondences between pixels. However, in the case of thin structures matching square or circular regions may lead to few useful correspondences. The first step towards dealing with thin structures is to extend the nonlocal-means filter by changing square regions with elongated regions, for instance by using a non isotropic Gaussian with variance ρ in the direction $\mathbf{v} \doteq [\cos(\theta)\ \sin(\theta)]^T$ defined by the angle θ, and variance approximately 0 along the orthogonal axis

$$G_{\rho,\theta} * |I(\mathbf{y})-I(\mathbf{x})|^2(0) \doteq \int_{\mathbb{R}} G_\rho(t)|I(\mathbf{y}+t\mathbf{v})-I(\mathbf{x}+t\mathbf{v})|^2 dt. \qquad (8)$$

Then, we can look for the best such region at each pixel and use it for the pixel matching, *i.e.*,

$$\hat{I}(\mathbf{y}) = \frac{1}{C^*(\mathbf{y})}\int_{\Omega} e^{-\min_\theta \frac{G_{\rho,\theta} * |I(\mathbf{y})-I(\mathbf{x})|^2(0)}{\tau}} I(\mathbf{x})d\mathbf{x} \qquad (9)$$

where $C^*(\mathbf{y})$ is the normalization factor corresponding to the selected θ at each $\mathbf{y}$. For notational simplicity, let us define the filtering weights

$$\mathcal{W}(\mathbf{x},\mathbf{y}) \doteq e^{-\min_\theta \frac{G_{\rho,\theta} * |I(\mathbf{y})-I(\mathbf{x})|^2}{\tau}}. \qquad (10)$$

Now, we can use the pixel correspondence strategy not only to denoise images, but also to regularize depth maps. We define the neighborhood regularization term so that pixels with similar colors are encouraged to have similar depth values, *i.e.*,

$$E_n[s] = \int \mathcal{W}(\mathbf{x},\mathbf{y})\left(s(\mathbf{y})-s(\mathbf{x})\right)^2 d\mathbf{x}d\mathbf{y}. \qquad (11)$$

If we evaluate the Euler-Lagrange equation with respect to the depth s, we obtain

$$\int \mathcal{W}(\mathbf{x},\mathbf{y})(s(\mathbf{y})-s(\mathbf{x}))d\mathbf{x} = 0 \quad \forall \mathbf{y}\in\Omega. \qquad (12)$$

By rearranging eq. (12) one immediately obtains that the minimum of $E_n[s]$ is the directional nonlocal-means filtering of s

$$s(\mathbf{y}) = \frac{1}{C^*(\mathbf{y})}\int \mathcal{W}(\mathbf{x},\mathbf{y})s(\mathbf{x})d\mathbf{x}. \qquad (13)$$

Remark 1 *Notice the similarity between eq.* (12) *and the upper bound to the nonparametric smoothness term derived through a Bayesian formulation in [20]. Indeed, the upper bound in [20] could be approximately derived by using the bilateral filter given in eq.* (5) *where also the spatial distance between pixels is taken into account and where correspondence is established via pixel-based matching. In contrast, in eq.* (12) *we use directional region matching, which yields more reliable correspondences, and avoid terms based on the spatial coordinates of pixels, which is equivalent to using a uniform probability density distribution in the Bayesian formulation. Finally, notice that our energy term is quadratic in the unknown depth map s and therefore it can be easily minimized.*

3. Shape Estimation: Depth from Defocus

The last term left to be defined in the energy minimization (1) is the data fidelity term. We consider the data term provided by a formulation of the problem of depth from defocus where there is no need for image restoration. Firstly, however, we need to introduce the notation and the image formation model.

Defocused images $I : \mathbb{Z}^2 \mapsto [0, \infty]$ have been successfully described with linear models of the type

$$I(\mathbf{y}) = \int_{\Omega \subset \mathbb{R}^2} k_\sigma(\mathbf{y}, \mathbf{x}) f(\mathbf{x}) d\mathbf{x} \tag{14}$$

where k_σ denotes the point spread function (PSF) of the camera and $f : \Omega \mapsto [0, \infty]$ is the sharp image of the scene. The PSF k_σ depends on the 3D surface $s : \mathbb{Z}^2 \mapsto [0, \infty]$ of the scene. The 2D coordinates $\mathbf{y} = [y_1\ y_2]^T$ lie on the sensor array, while the 2D coordinates $\mathbf{x} = [x_1\ x_2]^T$ parametrize points in 3D space. More specifically, the PSF is often approximated by a Gaussian kernel [9, 11]

$$\begin{aligned} k_\sigma(\mathbf{y}, \mathbf{x}) &\doteq \frac{1}{2\pi\sigma^2} e^{-\frac{\|\mathbf{y}-\mathbf{x}\|^2}{2\sigma^2}} \\ \sigma &\doteq \gamma \frac{Dv}{2} \left| \frac{1}{F} - \frac{1}{v} - \frac{1}{s(\mathbf{y})} \right|, \end{aligned} \tag{15}$$

where σ is the spread of the PSF and γ is a calibration parameter (the unit conversion of millimeters to pixels), D is the lens aperture, F is the focal length of the lens, and v is the spacing between the sensor and the camera lens. Other common choices are the Pillbox function

$$k_\sigma(\mathbf{y}, \mathbf{x}) \doteq \begin{cases} \frac{1}{\pi\sigma^2} & \|\mathbf{x} - \mathbf{y}\| < \sigma \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

where σ is defined as above. Both of these models ignore diffraction and other aberration effects and therefore hold only approximately. Nonetheless, such effects are relatively negligible in our data as the dimensions at play in our camera (*e.g.*, the pixel size) are sufficiently large. Our proposed method does not exploit one or the other choice. However, we find that the Pillbox function leads to a more computationally and memory efficient algorithm as it uses smaller supports for a given depth value. Notice that in general a calibration procedure to register the defocused images needs to be used, even if one employs telecentric optics [26] to eliminate scaling effects.

In shape from defocus, one is typically given two defocused images I_1 and I_2 obtained with different focus settings v_1 and v_2 respectively. This results in changes to the PSF k as shown in eqs. (15) and (16). The inference of s can be posed as the problem of matching the observations I_1 and I_2 to the defocused image model eq. (14). However, this requires the estimation of an additional unknown, the sharp image f. One way to avoid estimating f is to formulate the inference problem so that f is algebraically eliminated. This has been done in the literature with the so-called equifocal planar approximation, where the model (14) is locally approximated as a convolution and Fourier analysis allows to obtain a closed form solution. An alternative to such approximation is to match the observations to each other as it is done in stereopsis. Matching defocused images to each other has been done in the past in shape from defocus [10, 12, 11, 22]. The idea is to further blur with a kernel one image until it matches the other. We therefore consider the following approximate models

$$\begin{aligned} I_1(\mathbf{y}) &= \int k_{\sigma_1}(\mathbf{y}, \mathbf{x}) f(\mathbf{x}) d\mathbf{x} \simeq \int k_{\Delta\sigma}(\mathbf{y}, \bar{\mathbf{y}}) I_2(\bar{\mathbf{y}}) d\bar{\mathbf{y}} \\ &= \int k_{\Delta\sigma}(\mathbf{y}, \bar{\mathbf{y}}) \int k_{\sigma_2}(\bar{\mathbf{y}}, \mathbf{x}) f(\mathbf{x}) d\mathbf{x} d\bar{\mathbf{y}} \end{aligned} \tag{17}$$

$$\begin{aligned} I_2(\mathbf{y}) &= \int k_{\sigma_2}(\mathbf{y}, \mathbf{x}) f(\mathbf{x}) d\mathbf{x} \simeq \int k_{\Delta\sigma}(\mathbf{y}, \bar{\mathbf{y}}) I_1(\bar{\mathbf{y}}) d\bar{\mathbf{y}} \\ &= \int k_{\Delta\sigma}(\mathbf{y}, \bar{\mathbf{y}}) \int k_{\sigma_1}(\bar{\mathbf{y}}, \mathbf{x}) f(\mathbf{x}) d\mathbf{x} d\bar{\mathbf{y}} \end{aligned} \tag{18}$$

where eq. (17) holds for $\Xi \doteq \{\mathbf{y} : \sigma_1^2 > \sigma_2^2\}$ and eq. (18) holds in the complementary domain $\Xi_c \doteq \{\mathbf{y} : \sigma_2^2 > \sigma_1^2\}$. The relative spread $\Delta\sigma$ is defined as $\Delta\sigma \doteq \sqrt{\sigma_1^2 - \sigma_2^2}$ for all $\mathbf{y} \in \Xi$ and as $\Delta\sigma \doteq -\sqrt{\sigma_2^2 - \sigma_1^2}$ for all $\mathbf{y} \in \Xi_c$. To simplify the notation, we define

$$\begin{aligned} \hat{I}_{2,\Delta\sigma}(\mathbf{y}) &\doteq \int k_{\Delta\sigma}(\mathbf{y}, \bar{\mathbf{y}}) I_2(\bar{\mathbf{y}}) d\bar{\mathbf{y}} \\ \hat{I}_{1,\Delta\sigma}(\mathbf{y}) &\doteq \int k_{\Delta\sigma}(\mathbf{y}, \bar{\mathbf{y}}) I_1(\bar{\mathbf{y}}) d\bar{\mathbf{y}}. \end{aligned} \tag{19}$$

This allows us to write the following data term for the energy

$$\begin{aligned} E_{data}[s] =& \int_\Xi \Psi\left(\hat{I}_{2,\Delta\sigma}(\mathbf{y}) - I_1(\mathbf{y})\right) d\mathbf{y} \\ &+ \int_{\Xi_c} \Psi\left(\hat{I}_{1,\Delta\sigma}(\mathbf{y}) - I_2(\mathbf{y})\right) d\mathbf{y} \\ =& \int H(\Delta\sigma(\mathbf{y})) \Psi\left(\hat{I}_{2,\Delta\sigma}(\mathbf{y}) - I_1(\mathbf{y})\right) d\mathbf{y} \\ &+ \int (1 - H(\Delta\sigma(\mathbf{y}))) \Psi\left(\hat{I}_{1,\Delta\sigma}(\mathbf{y}) - I_2(\mathbf{y})\right) d\mathbf{y} \end{aligned} \tag{20}$$

where H denotes the Heaviside function, and Ψ is a robust norm. In our implementation we choose $\Psi(z) \doteq \sqrt{z^2 + \epsilon^2}$ with $\epsilon \doteq 10^{-3}$ and image intensities are in the range $[0, 255]$. The estimation of the surface s can then be obtained from the spread $\Delta\sigma$ via

$$s(\mathbf{y}) = \left(\frac{1}{F} - \frac{1}{v_2 - v_1} - \frac{1}{|v_2 - v_1|} \sqrt{1 + \frac{4\Delta\sigma|\Delta\sigma|}{\gamma^2 D^2} \frac{v_2 - v_1}{v_2 + v_1}} \right)^{-1}. \tag{21}$$

4. Iterated Linearization Scheme

The minimization (1) can be carried out in several ways. Because of numerical efficiency and the complexity of

Table 1. Numerical approximations for the total variation regularization scheme [5].

$$\begin{aligned}\nabla\cdot\left(\frac{\nabla s_{n,m}}{\|\nabla s_{n,m}\|}\right) \approx\ & |\nabla s_{n+\frac{1}{2},m}|(s_{n+1,m}-s_{n,m})\\ &-|\nabla s_{n-\frac{1}{2},m}|(s_{n,m}-s_{n-1,m})\\ &+|\nabla s_{n,m+\frac{1}{2}}|(s_{n,m+1}-s_{n,m})\\ &-|\nabla s_{n,m-\frac{1}{2}}|(s_{n,m}-s_{n,m-1})\end{aligned}$$

$$\begin{aligned}|\nabla s_{n+\frac{1}{2},m}| &\approx \sqrt{(s_{n+1,m}-s_{n,m})^2+\tfrac{1}{16}(s_{n+1,m+1}-s_{n+1,m-1}+s_{n,m+1}-s_{n,m-1})^2}\\ |\nabla s_{n-\frac{1}{2},m}| &\approx \sqrt{(s_{n,m}-s_{n-1,m})^2+\tfrac{1}{16}(s_{n-1,m+1}-s_{n-1,m-1}+s_{n,m+1}-s_{n,m-1})^2}\\ |\nabla s_{n,m+\frac{1}{2}}| &\approx \sqrt{(s_{n,m+1}-s_{n,m})^2+\tfrac{1}{16}(s_{n+1,m+1}-s_{n-1,m+1}+s_{n+1,m}-s_{n-1,m})^2}\\ |\nabla s_{n,m-\frac{1}{2}}| &\approx \sqrt{(s_{n,m}-s_{n,m-1})^2+\tfrac{1}{16}(s_{n+1,m-1}-s_{n-1,m-1}+s_{n+1,m}-s_{n-1,m})^2}.\end{aligned}$$

the neighborhood system, we choose to solve the Euler-Lagrange equations of the cost functional

$$\nabla E[s] \doteq \nabla E_{data}[s] + \alpha\nabla E_{tv}[s] + \beta\nabla E_n[s] = 0 \quad (22)$$

by iterative linearization [5]. The key idea is to describe the update to the depth map s as a small perturbation δ such that one can use the first-order approximation of the above equations

$$\nabla E[s+\delta] \approx \nabla E[s] + \langle \tfrac{\partial \nabla E[s]}{\partial s}, \delta\rangle = 0. \quad (23)$$

Then, once δ has been computed by inverting the linearised system, the depth map s is updated with $s+\delta$ and the step repeated until $\delta \approx 0$. To retain efficiency, the matrix $\frac{\partial \nabla E[s]}{\partial s}$ should satisfy the necessary conditions for convergence with the successive over-relaxation method [28]. Such conditions require that the relaxation parameter $0 < \omega < 2$ and that $\frac{\partial \nabla E[s]}{\partial s}$ be symmetric and positive-definite, which is typically not true. If we choose the relaxation parameter $\omega = 1$ successive over-relaxations reduces to Gauss-Seidel and convergence is guaranteed also when $\frac{\partial \nabla E[s]}{\partial s}$ is strictly diagonally dominant matrix, *i.e*., such that

$$\forall i : |e_{ii}| > \sum_{j\neq i} |e_{ij}| \qquad e_{ij} \doteq \left[\frac{\partial \nabla E[s]}{\partial s}\right]_{ij}. \quad (24)$$

When neither of these conditions are satisfied, one needs to resort to slower methods to solve linear systems, such as conjugate gradient on the least square formulation. One simple technique to help the convergence of the linearized system is to introduce an artificial term $\mu\delta$ with $\mu > 0$ in eq. (22) that penalizes large values of δ. In the first order approximation (23) this results in an identity matrix scaled by μ. Then, one can choose μ so that the resulting linear system is diagonally dominant. Otherwise, one could use other fast solvers such as Gaussian Belief Propagation provided that $\frac{\partial \nabla E[s]}{\partial s}$ is walk-summable [19, 16]. More details on the computation of the gradients are reported in the Appendix.

4.1. A Note on Pyramid Schemes

We also have implemented a coarse-to-fine (pyramid) scheme where the above equations are solved first on a down-sampled version of the input images and then the solution is up-sampled and used to initialize the next iteration. However, we find that this procedure has several problems: 1) it introduces a bias in the depth estimate towards large edges, and 2) the data term does not have useful matches for scales that are too low. The first issue might be due to the coarse resolution of the initial depth map inherited from the previous scale in the pyramid scheme. It seems that once a sharp edge is created, it is difficult for the algorithm to adjust its position at the higher scales. In the second issue the low high-frequency content in the down-sampled images seems to generate a plateau in the data fidelity term, *i.e*., there is a larger number of ambiguities in the solution. This is particularly evident in depth from defocus where the difference in frequency content of texture is used to estimate the depth map. For these reasons we currently use only 2 levels of the pyramid.

5. Experiments

Synthetic Data: In this section we demonstrate how the proposed method performs with different levels of noise in the input data and compare it to the bilateral filtering given in eq. (5) (which is comparable to using the nonparametric smoothness term in [20] at its best, *i.e*., where all correspondences are used). As one can see in Figure 2, the proposed method returns more reliable correspondences which then allow an accurate estimation of the edges. We synthetically generate defocused image pairs of a fronto-parallel plane occluded by a regular grid in the foreground. Then, we add 4 levels of Gaussian noise, namely, 0%, 1%, 2%, 5% of the maximum intensity. We also test the method for different number of neighbors used in the correspondences. This is shown in Figure 3 The test consists in keeping only the dominant components in the weight matrix for each neighborhood system. The weight matrix is then re-normalized with the remaining ones. We consider 5 cases: 2, 3, 4, 6, and 10 dominant components. It is evident that the number of correspondences is key to achieving high accuracy in the shape and position of the depth map. Finally, we assess the accuracy in the reconstruction of the depth map. For simplicity, we generate data with the Pillbox PSF and then use the same model in the matching term and focus on the estimation of the relative depth $\Delta\sigma$ as the depth map can be obtained via eq. (21). In general the PSF is not known unless one performs a calibration procedure. Furthermore, the matching term used in the proposed method is an approximation unless the PSF is a Gaussian and the depth maps are fronto-parallel planes. This results in distortions of the

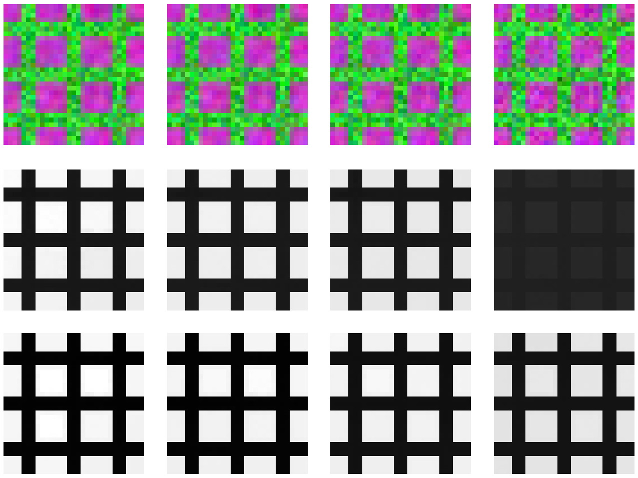

Figure 2. Comparison for different levels of noise in the input data. From left to right, each column shows experiments for additive noise in the input data with levels 0%, 1%, 2%, 5% of the maximum intensity. First row: one of the two input images. Second row: depth maps recovered with Bilateral filtering regularization. Third row: depth maps recovered with the proposed regularization. In all experiments only the 6 most significant weights were kept. The additive noise makes the depth estimation more difficult unless pixels move jointly.

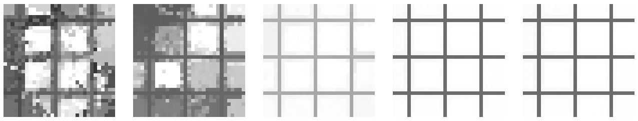

Figure 3. Comparison for different numbers of correspondences. Each column shows the depth map recovered with the proposed method for different sets of correspondences. From left to right, we keep only 2, 3, 4, 6, 10 dominant components in the weight matrix.

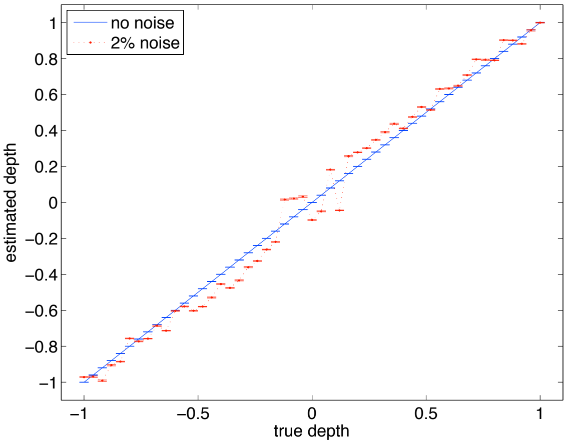

Figure 4. Estimated relative depth map (ordinate) versus ground truth (abscissa). We plot the mean and 3 times the standard deviation of the relative depth estimated at 51 planes with no noise (solid blue) and with 2% noise (dotted red).

depth map especially around the locations where the relative blur between the input data is approximately 0. We simulate planes at 51 depth locations and plot the mean and 3 times the standard deviation of the relative estimated depth $\Delta\sigma$ by using the proposed algorithm in Figure 4.

Real Data: We have tested our algorithms on real data that is publicly available [1, 2], where also specifications and settings of the hardware can be found, and on a data set that we have captured with a CANON EOS 5D SLR. In Figure 5 the first two rows show 3 publicly available data sets and a data set that we have captured (last column). Each set is made of two defocused images: In one image objects closer to the camera are in focus, and in the other image objects further away form the camera are in focus. The third and fourth rows show the resulting depth map and metallic-rendered surfaces obtained with the proposed method. Depth maps are encoded with brightness intensity values where dark intensities correspond to points far away from the camera and bright intensities correspond to points close to the camera. The metallic-rendered surfaces are used to illustrate the fine-details of the estimated surfaces.

In the data set that we have captured, we consider a scene with more elaborate objects containing thin structures. Also, to simulate a realistic scenario where a user captures two defocused images, the two images are captured by changing the focus setting of the lens while holding the camera in hand in two different time instants. This resulted in a small change in the viewpoint that needed adjustment. We registered the two frames by using an affine transformation and used a pyramid scheme to accelerate the convergence. Notice that, due to the non-planar 3D surface of the scene, the alignment is reasonable but not perfect. However, the method is quite robust to such small misalignments and still retrieves accurate edges and thin structures (see magnification of a detail in Figure 1). In the simulation we have used a MacBook 2.4GHz Core 2 Duo, with a (reasonably) optimized Matlab implementation of the iterated linearization methods. The running time for each simulation strongly depends on the amount of defocus in the input data. As a rule of thumb, more defocus requires more computational time. On average the simulations with the proposed method required about 10 minutes on 640×480 pixels images. Notice that the nonlocal-means neighborhood term increases the number of computations substantially not only during the evaluation of the weights for each correspondence, but also by decreasing the convergence rate of the successive over-relaxations algorithm (due to the larger neighborhood structures).

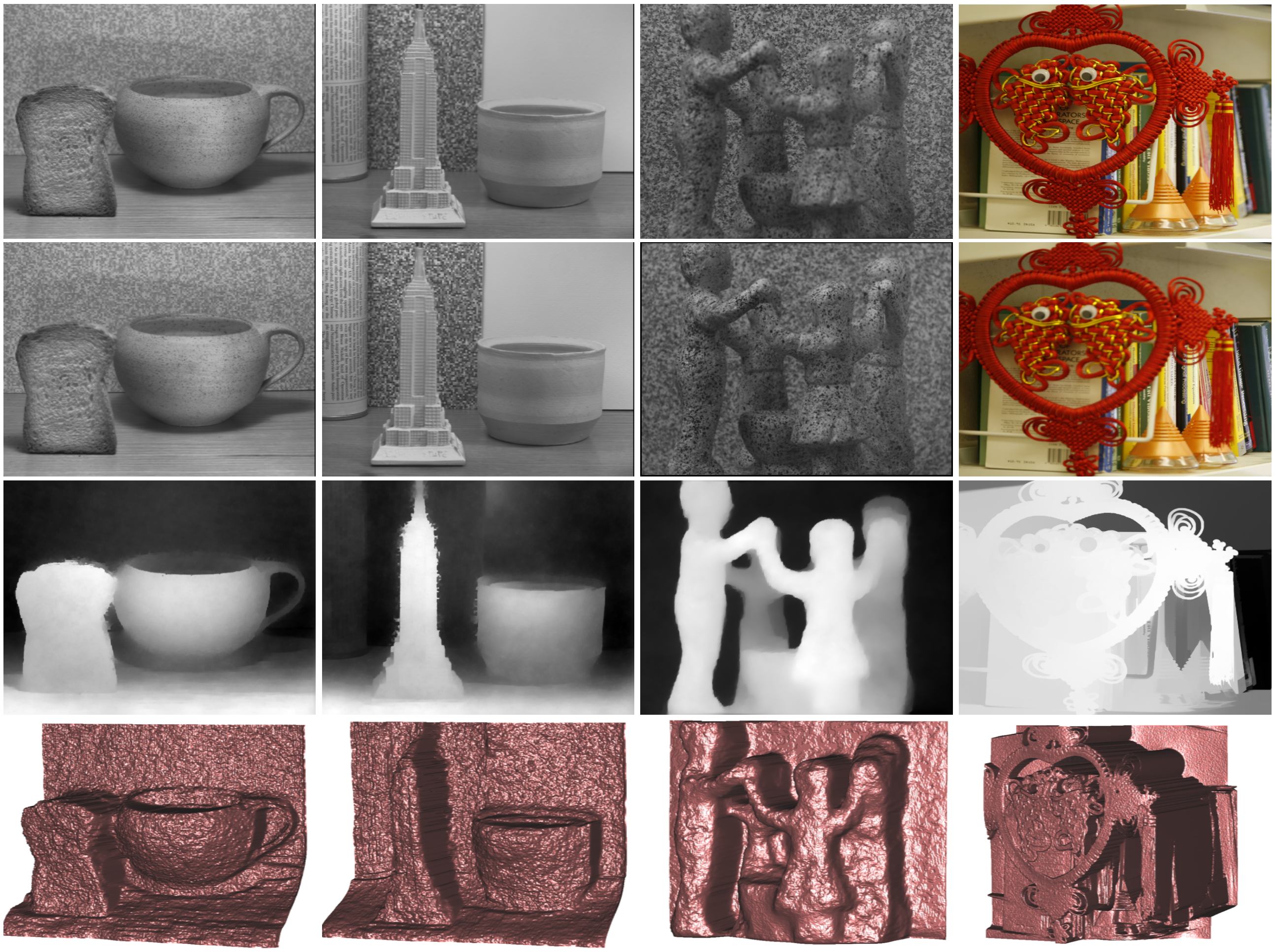

Figure 5. Comparison of the proposed methods on publicly available real data and a data set that we have captured (rightmost column). The first two rows show the two input images obtained with different focus settings (and scaled). The third and the fourth rows show two renderings of the resulting depth maps obtained the proposed iterative linearization method. Notice that our algorithm compares favorably with state-of-the-art methods (see [2] and [1]). Correspondence is much more accurate in color images than just grayscale values. Also, notice that in some data sets the texture of the objects can be easily confused with the texture in the background.

6. Limitations and Discussion

Determining which pixels share the same surface is by and large still an unsolved problem. Indeed by matching pixels with similar color we might also connect surfaces that are completely uncorrelated. Vice versa, the same surface might have regions with very different colors. In both cases the proposed procedure might introduce artifacts: averaging uncorrelated surfaces or creating incorrect edges. Nonetheless, the proposed regularization seems to be quite helpful in most scenarios in depth from defocus. The idea of establishing reliable correspondences by comparing elongated windows and then penalizing corresponding pixels with different depth is demonstrated in the precise estimation of boundaries of thin surfaces.

Appendix

Now, the computation of $\nabla E[s]$ amounts to the evaluation of three terms: $\nabla E_{data}[s]$, $\nabla E_{tv}[s]$, and $\nabla E_n[s]$. We derive them directly from the Euler-Lagrange equations:

$$\nabla E_{data}[s] \doteq \frac{\partial \Delta\sigma}{\partial s}(\mathbf{y}) \Bigg[H(\Delta\sigma(\mathbf{y}))\Psi'\Big(\hat{I}_{2,\Delta\sigma}(\mathbf{y}) - I_1(\mathbf{y})\Big)\frac{\partial \hat{I}_{2,\Delta\sigma}(\mathbf{y})}{\partial \Delta\sigma} + (1 - H(\Delta\sigma(\mathbf{y})))\Psi'\left(\hat{I}_{1,\Delta\sigma}(\mathbf{y}) - I_2(\mathbf{y})\right)\frac{\partial \hat{I}_{1,\Delta\sigma}(\mathbf{y})}{\partial \Delta\sigma}\Bigg], \tag{25}$$

$$\nabla E_{tv}[s] \doteq -\nabla \cdot \left(\frac{\nabla s(\mathbf{y})}{|\nabla s(\mathbf{y})|} \right), \tag{26}$$

and

$$\nabla E_n[s] \doteq \int \mathcal{W}(\mathbf{x}, \mathbf{y})(s(\mathbf{y}) - s(\mathbf{x}))d\mathbf{x}. \tag{27}$$

Notice that the Dirac delta terms in eq. (25) cancel each other and that the chosen robust norm yields $\Psi'(z) =$

$\frac{z}{\sqrt{z^2+\epsilon^2}}$. The second term $\frac{\partial \nabla E[s]}{\partial s}$ is a matrix and can be evaluated by defining

$$\langle \frac{\partial \nabla E[s]}{\partial s}, \delta \rangle \doteq \langle \frac{\partial \nabla E_{data}[s]}{\partial s}, \delta \rangle + \alpha \langle \frac{\partial \nabla E_{tv}[s]}{\partial s}, \delta \rangle + \beta \langle \frac{\partial \nabla E_{n}[s]}{\partial s}, \delta \rangle \tag{28}$$

where

$$\begin{aligned}\langle \frac{\partial \nabla E_{data}[s]}{\partial s}, \delta \rangle \approx & \left(\frac{\partial \Delta\sigma}{\partial s} \right)^2 \Big[H(\Delta\sigma)\Psi'\Big(\hat{I}_{2,\Delta\sigma} - I_1\Big)\left(\frac{\partial \hat{I}_{2,\Delta\sigma}}{\partial \Delta\sigma} \right)^2 \\ & + (1 - H(\Delta\sigma))\Psi'\left(\hat{I}_{1,\Delta\sigma} - I_2\right)\left(\frac{\partial \hat{I}_{1,\Delta\sigma}}{\partial \Delta\sigma} \right)^2 \Big] \delta,\end{aligned} \tag{29}$$

$$\langle \frac{\partial \nabla E_{tv}[s]}{\partial s}, \delta \rangle \approx -\nabla \cdot \left(\frac{\nabla \delta}{|\nabla s|} \right), \tag{30}$$

and

$$\langle \frac{\partial \nabla E_{n}[s]}{\partial s}, \delta \rangle = \text{diag}\left[\int \mathcal{W}(\mathbf{x},\mathbf{y}) d\mathbf{x} \right] - \mathcal{W}(\mathbf{y},\mathbf{x}). \tag{31}$$

Notice that in eq. (29) and eq. (30) we have ignored second order derivatives that appear as a result of the derivatives with respect to s and the (highly nonlinear) terms in the Dirac delta.

References

[1] http://www1.cs.columbia.edu/cave/software/softlib/raf.php. Rational Filters for Focus Analysis.

[2] http://www.eps.hw.ac.uk/˜pf21/pages/page4/page4.html. Shape from Defocus Code.

[3] S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *Proc. of the Intl. Conf. on Comp. Vision*, pages 489–495, 1999.

[4] T. Brox. *From Pixels to Regions: Partial Differential Equations in Image Analysis*. PhD thesis, Saarland University, Apr 2005.

[5] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert. High accuracy optical flow estimation based on a theory for warping. *European Conference on Computer Vision*, 4:25–36, May 2004.

[6] A. Buades, B. Coll, and J.-M. Morel. Nonlocal image and movie denoising. In *International Journal of Computer Vision*, volume 76(2), pages 123–139, 2007.

[7] T. Chan, P. Blongren, P. Mulet, and C. Wong. Total variation blind deconvolution. *IEEE Intl. Conf. on Image Processing*, 1997.

[8] T. F. Chan and J. Shen. *Image processing and analysis : variational, PDE, wavelet, and stochastic methods*. Society for Industrial and Applied Mathematics, Philadelphia, 2005.

[9] S. Chaudhuri and A. N. Rajagopalan. Depth from defocus: a real aperture imaging approach. *Springer-Verlag*, 1999.

[10] J. Ens and P. Lawrence. An investigation of methods for determining depth from focus. *IEEE Trans. Pattern Anal. Mach. Intell.*, 15:97–108, 1993.

[11] P. Favaro and S. Soatto. *3-d shape reconstruction and image restoration: exploiting defocus and motion-blur.* Springer-Verlag, 2006.

[12] P. Favaro, S. Soatto, L. A. Vese, and S. J. Osher. 3-d shape from anisotropic diffusion. *Proc. IEEE Computer Vision and Pattern Recognition*, I:179–186, 2003.

[13] S. W. Hasinoff and K. N. Kutulakos. Confocal stereo. *European Conf. on Computer Vision*, pages 620–634, 2006.

[14] A. Klaus, M. Sormann, and K. Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *Proceedings of Int. Conference on Pattern Recognition*, pages III: 15–18, 2006.

[15] J. Lee. Digital image smoothing and the sigma filter. *Comp. Vision, Graphics and Image Proc.*, 24(2):255–269, November 1983.

[16] D. M. Malioutov, J. K. Johnson, and A. S. Willsky. Walk-sums and belief propagation in gaussian graphical models. *Journal of Machine Learning Research*, 5, 2006.

[17] D. Nister, H. Stewenius, R. Yang, L. Wang, and Q. Yang. Stereo matching with color-weighted correlation, hierachical belief propagation and occlusion handling. In *Proc. IEEE Conf. on Comp. Vision and Pattern Recogn.*, pages II: 2347–2354, 2006.

[18] A. Pentland. A new sense for depth of field. *IEEE Trans. Pattern Anal. Mach. Intell.*, 9:523–531, 1987.

[19] O. Shental, D. Bickson, P. H. Siegel, J. K. Wolf, and D. Dolev. Gaussian belief propagation solver for systems of linear equations. *in IEEE Int. Symp. on Inform. Theory (ISIT)*, 2008.

[20] B. Smith, L. Zhang, and H. Jin. Stereo matching with non-parametric smoothness priors in feature space. In *Proc. IEEE Conf. on Comp. Vision and Pattern Recogn.*, pages 485–492, 2009.

[21] S. Smith and J. Brady. Susan: A new approach to low-level image-processing. *Int. J. of Computer Vision*, 23(1):45–78, May 1997.

[22] M. Subbarao and G. Surya. Appplication of spatial-domain convolution/deconvolution transform for determining distance from image defocus. In *SPIE*, 1992.

[23] H. Tao, H. Sawhney, and R. Kumar. Dynamic depth recovery from multiple synchronized video streams. In *Proc. IEEE Conf. on Comp. Vision and Pattern Recogn.*, pages I:118–124, 2001.

[24] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proc. of the Intl. Conf. on Comp. Vision*, pages 839–846, 1998.

[25] M. Watanabe and S. Nayar. Rational filters for passive depth from defocus. *Intl. J. of Comp. Vision*, 27(3):203–225, 1998.

[26] M. Watanabe and S. K. Nayar. Telecentric optics for focus analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(12):1360–1365, December 1997.

[27] L. P. Yaroslavsky. *Digital Picture Processing*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1985.

[28] D. M. Young. *Iterative Solution of Large Linear Systems*. Academic Press, New York, 1971.