Vis Comput
DOI 10.1007/s00371-014-1050-2

ORIGINAL ARTICLE

Fast depth from defocus from focal stacks

Stephen W. Bailey · Jose I. Echevarria ·
Bobby Bodenheimer · Diego Gutierrez

Abstract We present a new depth from defocus method based on the assumption that a per pixel blur estimate (related with the circle of confusion), while ambiguous for a single image, behaves in a consistent way when applied over a focal stack of two or more images. This allows us to fit a simple analytical description of the circle of confusion to the different per pixel measures to obtain approximate depth values up to a scale. Our results are comparable to previous work while offering a faster and flexible pipeline.

Keywords Depth from defocus · Shape from defocus

1 Introduction

Among single view depth cues, focus blur is one of the strongest, allowing a human observer to instantly understand the order in which objects are arranged along the z axis in a scene. Such cues have been extensively studied to estimate depth from single viewpoint monocular systems [7]. The acquisition system is simple: from a fixed point of view, several images are taken, changing the focal distance consecutively for each shot. This set of images is usually called a focal stack, and depending on the number of images in it, different approaches to estimate depth can be taken. When the number of images is high, a *shape from focus* [28] approach aims to detect the focal distance with maximal sharpness for each pixel, obtaining a robust first estimate that can be further refined.

With a small number of images in the focal stack (as low as two), that approach is not feasible. *Shape from defocus* [30] techniques use the information contained in the blurred pixels based on the idea of the circle of confusion, which relates the focal position of the lens and the distance from a point to the camera with the resulting size of the out-of-focus blur circle in an image.

Estimating the degree of blur for a pixel in a single image is difficult and prone to ambiguities. However, we propose the hypothesis that those ambiguities are possible to disambiguate by applying and analyzing the evolution of the blur estimates for each single pixel through the whole focal stack. This process allows us to fit an analytical description of the circle of confusion to the different estimates, obtaining actual depth values up to a scale for each pixel. Our results demonstrate that this hypothesis holds, providing reconstructions comparable to those found in previous work, and making the following contributions:

- We show that single image blur estimates can behave in a robust way when applied over a focal stack, with the potential to estimate accurate depth values up to a scale.
- A fast and flexible method, with components that can be easily improved independently as respective state of the art advances.
- A novel normalized convolution scheme with an edge-preserving kernel to remove noise from the blur estimates.

S. W. Bailey
University of California at Berkeley, Berkeley, USA
e-mail: stephen.w.bailey@berkeley.edu

J. I. Echevarria (✉)
Universidad de Zaragoza, Zaragoza, Spain
e-mail: jiecheva@unizar.es

B. Bodenheimer
Vanderbilt University, Nashville, USA
e-mail: bobby.bodenheimer@vanderbilt.edu

D. Gutierrez
Universidad de Zaragoza, Zaragoza, Spain
e-mail: diegog@unizar.es

Journal: 371 MS: 1050 TYPESET DISK LE CP Disp.:2014/11/17 Pages: 11 Layout: Large

– A novel global error metric that allows the comparison of depth maps with similar global shapes but local misalignments of features.

2 Related work

There is a vast amount of literature on the topic of estimating depth and shape based on monocular focus cues; we comment on the main approaches and how they relate to ours. First, we discuss active methods that make use of additional hardware or setups to control the defocus blur. Next, we discuss passive methods that depend on whether the information comes from focused or defocused areas.

Active methods Levin et al. [15] use coded apertures that modify the blur patterns captured by the sensor. Moreno-Noguer et al. [20] project a dotted pattern over the scene during capture. In the depth from diffusion approach [32], an optical diffuser is placed near the object being photographed. Lin et al. [17] combine a single-shot focal sweep and coded sensor readouts to recover full resolution depth and all-in-focus images. Our approach does not need any additional or specialized hardware, so it can be used with regular off-the-shelf cameras or mobile devices like smartphones and tablets.

Passive methods: shape from focus These methods start computing a focus measure [24] for each pixel of each image in the focal stack. A rough depth map can then be easily built assigning to each of its pixels the position in the focal stack for which the focus measure of that pixel is maximal. As the resolution of the resulting depth map in the z axis depends critically on the number of images in the focal stack, this approach usually employs a large number of them (several tens). Improved results have been obtained when focus measures are filtered [18,22,27] or smoother surfaces fitted to the previously estimated depth map [28]. Our method uses fewer images and the resolution in the z axis is independent of the number of them.

Passive methods: shape from defocus In this approach, the goal is to estimate the blur radius for each pixel, which varies according to its distance from the camera and focus plane. Since the focus position during capture is usually known, a depth map can be recovered [23]. This approach significantly reduces the number of images needed for the focal stack, ranging from a single image to a few of them.

Approaches using only a single image [1,3,4,21,33,34] make use of complex focus measures and filters to obtain good results in many scenarios. However, they are not able to disambiguate cases where the blur cannot be known to come from the object being in front of or behind the focus plane (see Fig. 2). Cao et al. [5] solves this ambiguity through user input.

Using two or more images, Watanabe and Nayar [30] proposed an efficient set of broadband rational operators, invariant to texture, that produces accurate, dense depth maps. However, those sets of filters are not easy to customize. Favaro et al. [8] model defocus blur as a diffusion process based on the heat equation, then they reconstruct the depth map of the scene estimating the forward diffusion needed to go from a focused pixel to its blurred version. Our algorithm is not based on the heat diffusion model but on heuristics that are faster to compute. Favaro [6] imposes constraints for the reconstructed surfaces based on the similarity of their colors. The results presented there show great details, but as acknowledged by the author, color cannot be considered a robust feature to determine surface boundaries. Li et al. [16] use shading information to refine depth from defocus results in an iterative method.

Hasinoff and Kutulakos [9] proposed a method that uses variable aperture sizes along with focal distances for detailed results. However, such an approach needs the aperture size to be controllable and they use hundreds of images for each depth map.

Our work follows a shape from defocus approach with a reduced focal stack of at least two images. We use simple but robust per-pixel blur estimates, coupled with high-quality image filtering to remove noise and increase robustness. We analyze the evolution of the blur at each pixel through the focal stack by fitting it to an analytical model for the blur size, which returns the distance of the object from the camera up to a scale.

3 Background

The circle of confusion is the resulting blur circle captured by the camera when light rays from a point source out of the focal plane pass through a lens with a finite aperture [11]. The diameter c of this circle depends on the aperture size A, focal length f, the focal distance S_1, and the distance S_2 between the point source and the lens (see Fig. 1). Keeping the aperture size, focal length, and distance between the lens and the point source constant, the diameter of the circle of confusion can be controlled by varying the focal position using the following relation when the focal position S_1 is finite:

$$c = c(S_1) = A \frac{|S_2 - S_1|}{S_2} \frac{f}{S_1 - f} \tag{1}$$

and when the focal position S_1 is infinite

$$c = \frac{fA}{S_2}. \tag{2}$$

As shown in Fig. 2, the relation between the focal position S_1 and c is non-linear. The behavior of Eq. 1 is not symmetric

Journal: 371 MS: 1050 TYPESET DISK LE CP Disp.:2014/11/17 Pages: 11 Layout: Large

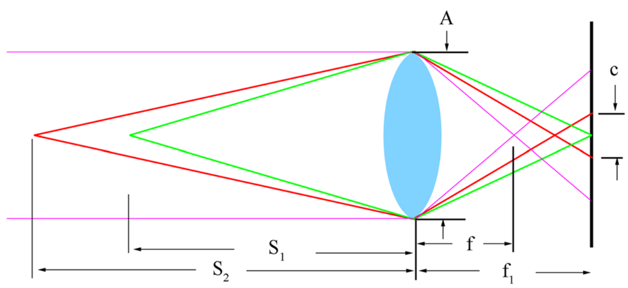

Fig. 1 Diagram showing image formation on the sensor when points are located on the focal plane (*green*), or out of it (*red* and *pink*)

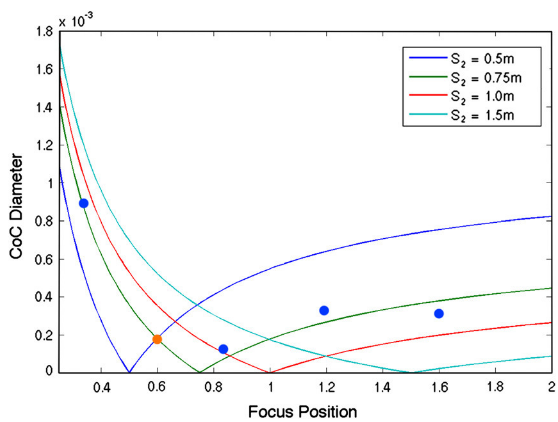

Fig. 2 Circle of confusion (CoC) diameter vs. focus position of the lens for points located at different distances from the camera S_2 (axis units in meters). *Plots* show how points become focused (smaller CoC) as the focal distance gets closer to their actual positions. It can be seen how different combinations of focal and object distances produce intersecting CoC plots, so a CoC measure from a single shot (*orange dot*) is not enough to disambiguate the actual position of the object (potentially at $S_2 = 0.5$ or $S_2 = 0.75$ for the depicted case). *Blue dots* show estimations from additional focus positions that, even without being perfectly accurate, have the potential to be fitted to the CoC function that returns the actual object position $S_2 = 0.75$ (shown by the *green line*) when its output is zero

around the distance of the focal plane (S_2), and approaches infinity for objects in front of the focal plane (making them disappear from the captured image) and asymptotically approaches the value given by Eq. 2 for objects behind it.

Our goal is to obtain the distance of each object S_2 for each pixel in the image. But, as seen from Eq. 1 and Fig. 2, even knowing all parameters A, S_1, c and f, there is ambiguity when recovering the position of S_2 with respect to the focus position S_1. So, instead of just using one estimate for c, the method described in this paper is based on the assumption that additional $n \geq 2$ estimates of c, c_i, $1 \leq i \leq n$, for different known focal distances S_1, S_1^i, will allow us to determine the single S_2 value that makes Eq. 1 optimally approximate all the measures obtained.

4 Algorithm

Our shape from defocus algorithm starts with a series of images that capture the same stationary scene but vary the focal position of the lens, a *focal stack*. For each image in the focal stack, we compute an estimate of the amount of blur using a two-step process. First, a focus measure is applied to each pixel of each image in the stack. This procedure generates reliable blur estimates near edges. We next determine which blur estimates are unreliable or invalid, and extrapolate them based on the existing irregularly sampled estimates in each image. For this step, we propose a novel combination of normalized convolution [13] with an edge-preserving filter for its kernel.

With blur estimates for each pixel in each image, we proceed to estimate per-pixel depth values fitting our blur estimates to the analytical function for the circle of confusion. We construct a least squares error minimization problem to fit the estimates to that function. Minimizing this problem gives the optimal depth for a point in the scene.

4.1 Focal stack

The input to our algorithm is a set of n images where $n \geq 2$. In our tests, we use 2 or 3 images. Each image captures the same stationary scene from the same viewpoint. The only difference between each image is the focal distance of the lens when the image is captured. Thus, each point in the object space will have varying circles of confusion in each image of the focal stack. Additionally, the focal position S_1^i of the lens when the image is captured is saved, where i is the ith image in the focal stack. While this information can be obtained easily from different sources (EXIF data, APIs to access digital cameras or physical dials on the lenses), in its absence a rough estimate of the focal distances based on the location of the objects in focus may suffice (Fig. 9).

In this paper, we assume that the images are perfectly registered to avoid misalignments due to the magnification that occurs when the focal plane changes. This can be achieved using telecentric optics [30] or image processing algorithms [6,9,29].

4.2 Local blur estimation

Our first step is to apply a focus measure that will give a rough estimate of the defocus blur for each pixel and thus an estimation of its circle of confusion. Several different measures have been proposed previously [24]. In our case, Hu and De Haan's [12] provided enough robustness and consistency to track the evolution of blur over the focal stack.

Given user defined parameters σ_a and σ_b, representing the blur radii of two Gaussian functions with $\sigma_a < \sigma_b$, the local blur estimation algorithm is applied to the focal stack. The

Journal: 371 MS: 1050 TYPESET DISK LE CP Disp.:2014/11/17 Pages: 11 Layout: Large

algorithm estimates a radius of the Gaussian blur kernel σ for each signal in each image in the focal stack. Note that σ_a and σ_b are chosen a priori and for the algorithm to work well $\sigma_a, \sigma_b \gg \sigma$. We empirically chose $\sigma_a = 4$ and $\sigma_b = 7$ for images of size 720×480. For the one-dimensional case, the radius of the Gaussian blur kernel, σ, is estimated as follows:

$$\sigma(x) \approx \frac{\sigma_a \cdot \sigma_b}{(\sigma_b - \sigma_a) \cdot r_{\max}(x) + \sigma_b} \tag{3}$$

with

$$r_{\max}(x) = \frac{I(x) - I_a(x)}{I_a(x) - I_b(x)} \tag{4}$$

where x is the offset into the image, and $I(x)$ is the input image; $I_a(x)$ and $I_b(x)$ are $I_b(x)$ are blurred versions of $I(x)$ using the blur kernels σ_a and σ_b, respectively. For 2-D images, isotropic 2D Gaussian kernels are used. We work with luminance values from the captured RGB images.

Because this algorithm depends on the presence of edges (discontinuities in the luminance), regions of the image far from edges or significant changes in signal intensities need to be estimated by other means. Consider a region of the image that is sufficiently far from an edge; for example, around $3\sigma_a$ from an edge, the intensities of the original image $I(x)$ and the blurred images $I_a(x)$ and $I_b(x)$ will be close to each other because the intensities in a neighborhood around x in the original image I are similar. This similarity causes the difference ratio maximum $r_{\max}(x)$ from Eq. 4 to go to zero if the numerator approaches zero or to infinity if the denominator approaches zero. If $r_{\max}(x)$ approaches zero, then from Eq. 3 the estimated blur radius approaches σ_a, and if $r_{\max}(x)$ approaches infinity, then the estimate approaches zero. Figure 3 shows an example of the blur maps obtained with this method.

It is important to note that similar to other single image blur measures, the method in [12] is not able to disambiguate an out-of-focus edge from a blurred texture. However, since we are using several images taken with different focus settings, our algorithm will seamlessly deal with their relative changes in blur during the optimization step (Sect. 4.4).

4.3 Noise filtering and data interpolation

Because of the assumption that $\sigma_a, \sigma_b \gg \sigma$, the above algorithm does not perform well in regions of the image far from edges where $\sigma \to \sigma_a$. Moreover, for constructing our depth map, we assume that discontinuities in depth correspond to discontinuities in the edge signals of an image, but the converse does not hold since they can come from discontinuities due to changes in texture, lighting, etc. The local blur estimation algorithm performs better over such discontinuities, but leaves uniform regions with less accurate estimations.

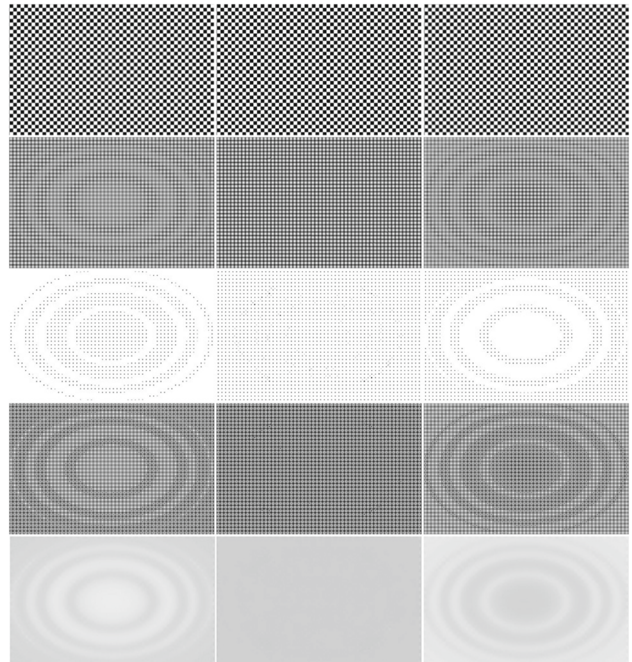

Fig. 3 From *top to bottom*, the different steps of our algorithm: Input focal stack consisting of three images (*left to right*) from a synthetic dataset (more details in Sect. 5.1). Initial blur estimations. Confidence maps from Eq. 6. Masked blur maps after Eq. 7. Refined blur maps after the application of normalized convolution. It can be seen how we are able to produce smooth and consistent blur maps to be used as the input for our fitting step. Final reconstruction for this example is shown in Sect. 5

Thus, we need a way of reducing noise by interpolating data to those areas. A straightforward approach to filter noise is to process pixels along with their neighbors over a small window. However, choosing the right window size is a problem on its own [14,19] as large windows can remove detail in the final results. So, we propose a novel combination of normalized convolution [13] with an edge-preserving filter for its kernel.

We use normalized convolution since this method is well suited for interpolating irregularly sampled data. Normalized convolution works by separating the data and the operator into a signal part $H(x)$ and a certainty part $C(x)$. Missing data is given a certainty value of 0, and trusted data a value of 1. Using $H(x)$ and $C(x)$ along with filter kernel $g(x)$ to interpolate, normalized convolution is applied as follows:

$$\bar{H}(x) = \frac{H(x) * g(x)}{C(x) * g(x)} \tag{5}$$

where $\bar{H}(x)$ is the resulting data with interpolated values for the missing data.

As the first step, we categorize good blur radius estimates and poor ones, which we then mark as missing data. Poor estimates will correspond to estimates for discrete signals

Journal: 371 MS: 1050 TYPESET DISK LE CP Disp.:2014/11/17 Pages: 11 Layout: Large

in the input image that are sufficiently far from detectable edges, and can be identified by their values being close to σ_a. Thus, we define good estimates as any blur estimate σ contained in the interval $[0, \sigma_a - \delta)$ and invalid estimates are contained in the interval $[\sigma_a - \delta, \sigma_a]$ where $\delta > 0$. In our experiments, we found that a value of $0.15\sigma_a$ worked well for δ. The confidence values for normalized convolution are then generated as follows:

$$C(x) = \begin{cases} 1 & \text{if } \sigma(x) < \sigma_a - \delta \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where $\sigma(x)$ is from Eq. 3. Figure 3 shows the confidence maps for the sparse blur map generated from the prior stage of pipeline. Similarly, the discrete input signal for normalized convolution is generated as follows:

$$H(x) = \begin{cases} \sigma(x) & \text{if } \sigma(x) < \sigma_a - \delta \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

With the resulting confidence values and input data, we only need to select a filter kernel $g(x)$ to use with normalized convolution.

Since we have estimates for discrete signals near edges in the image and need to interpolate signals far from edges, we want to use an edge-preserving filter. A filter with this property ensures that discontinuities between estimates that are caused by discontinuities in intensity in the original input signal are preserved, while spatially close regions with similar intensities will be interpolated based on valid nearby estimates that share similar intensities in the original image from the focal stack. There are several filters that have this property including the joint bilateral filter [25] and the guided image filter [10]. We use the guided image filter because of its efficiency and proven effectiveness [2]. In the absence of better guides, we use the original color images from the focal stack as the guides for the corresponding blur maps. With this filter as the kernel, we apply normalized convolution as described in Eq. 5. We use this technique to generate refined blur estimates for each image in the focal stack. The size of the spatial kernel for the guided image filter needs to be large enough to create an estimation of the Gaussian blur radius for every discrete signal in the image. Therefore, sparser maps require larger spatial kernels. The guided image filter has two parameters, the radius of the window and a value ϵ related to edge and detail preservation. Experimentally, we found that a window radius of between 15 and 30, and ϵ of 7.5e−3 works well for our focal stacks. The end result is a set of n maps, $\bar{H}_i(x)$, that estimate the radius of the Gaussian blur kernel in image i of the focal stack. Since the circle of confusion can be modeled as a Gaussian blur, these maps can be used to estimate the diameter of the circle of confusion for each pixel in each image of the focal stack. Figure 3 shows the output of the normalized convolution for each image in the focal stack.

4.4 Fit to the analytical circle of confusion function

Through the previous steps, each image I_i in the focal stack of size n is accompanied by the focal distance of the shot S_1^i. We can then estimate actual depth information. We first show how to do this for one pixel and its n circle of confusion estimations.

Given Eq. 1 for the circle of confusion, every variable is currently known or estimated except for S_2, the unknown depth. Solving for S_2 using only one estimate for the circle of confusion is not possible because of the ambiguity shown in Fig. 2; otherwise, there will be two possible values for S_2, as shown in the following equation:

$$S_2 = \frac{S_1}{\pm\frac{c(S_1 - f)}{Af} - 1}. \tag{8}$$

To find a unique S_2, a system of non-linear equations is constructed where we attempt to solve for S_2 that satisfies all of the equations. Each equation solves for depth given the circle of confusion estimates c_i for one image of the focal stack:

$$S_2 = \frac{S_1^i}{\pm\frac{c_i(S_1^i - f)}{Af} - 1} \quad \text{for all } i = 1, .., n \tag{9}$$

Since these equations are not, in almost all cases, satisfied simultaneously, we use a least squares method to minimize the error where we want to reduce the error in measured value for the circle of confusion. Thus, we obtain the following function to minimize:

$$\sum_{i=1}^{n} \left(c_i - A\frac{|S_2 - S_1^i|}{S_2}\frac{f}{S_1 - f} \right)^2 \tag{10}$$

This equation leads to a single-variable non-linear function whose minimizer is the best depth estimation for the given blur estimates. The resulting optimization problem is tractable using a variety of methods [26]. In our implementation, we use quadratic interpolation with the number of iterations fixed at four. This single-variable optimization problem can then be extended to estimate depth for each discrete pixel in the image. The result is a depth map that can be expressed as:

$$D(x) = \min \left[\sum_{i=1}^{n} \left(c_i(x) - A\frac{|S_2 - S_1^i|}{S_2}\frac{f}{S_1 - f} \right)^2 \right] \text{ for } S_2$$

To make our optimization run quickly, we assume bounds on the range of values that S_2 can have for each pixel. In

Journal: 371 MS: 1050 TYPESET DISK LE CP Disp.:2014/11/17 Pages: 11 Layout: Large

particular, we assume that the depth of at every point in the scene lies between the nearest focal length and the farthest focal length of all the images in the focal stack [30]. Note that this assumption is only necessary for fast optimization; methods that have an unbounded range exist [26].

However, because of this assumption every blur estimate needs to be scaled to ensure there are local minimizers of Eq. 10 that lie somewhere within the assumed range of depth. As shown in Appendix A, to ensure that there is a minimizer on the interval between the closest and farthest focal distances, an upper bound on the blur estimates c_i must be imposed. This bound is given by

$$\frac{Af}{S_1^j - f} = r \geq 2c. \tag{11}$$

Furthermore, we know that all blur estimates generated from normalized convolution are between 0 and σ_a. Thus, some positive scalar s can be defined as follows:

$$s \leq \frac{Af}{2\sigma_a(S_1^n - f)} \tag{12}$$

where S_1^n is the largest focal distance in the stack. Multiplying each blur estimate by s ensures that Eq. 11 is satisfied for all blur estimates, which implies that under normal conditions, there will be at least one local minimizer for Eq. 10 between the nearest and farthest focal distances. Figure 5 shows the final depth map for the focal stack from Fig. 3.

5 Results

In the following, we test our algorithm with synthetic scenes. Next, we run it over real scenes from previous work to allow visual comparisons between methods. Our algorithm can run in linear time. The C++ implementation of our algorithm takes less than 10s to generate the final depth map for 640 × 480 inputs on an Intel Core i7 2620M @ 2.7 GHz.

5.1 Synthetic scenes

To validate the accuracy of our algorithm, we generated synthetic focal stacks similar to those in prior work [8,18]. In particular, we used the slope, sinusoidal and wave objects as shown in Fig. 4.

To create the synthetic focal stacks, we start from an in-focus image and its depth map. Using Eq. 1, we are able to estimate the amount of blur c to be applied to each pixel of the image. We assume that the depth map ranges between 0.45 and 0.7 m, and the lens parameters are $f = 30$ mm and f-number $N = 2.5$. We then obtain three different images for each focal stack, with focal distances set to $S_1^1 = 0.4$ m, $S_1^2 = 0.6$ m and $S_1^3 = 1.0$ m (the resulting focal stack for the wave example can be found in Fig. 3).

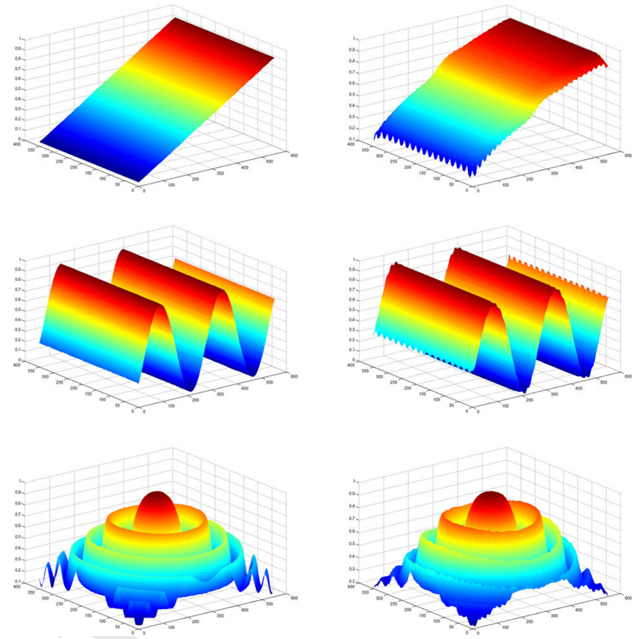

Fig. 4 3D Visualizations of the original depth maps (*left*) and our estimated depth maps (*right*). As can be seen, the global shape of the object is reconstructed in a recognizable way in all cases

Figure 4 shows the results of running our algorithms over these focal stacks, compared against the ground truth data. As can be seen, the global shape of the object is properly captured, but there are also noticeable local errors at different scales. Standard error metrics are thus difficult to apply because of their aggregation of these local error measures. Thus, we propose a novel error metric that favors the global shape comparing relativity between original and estimated depth values.

5.2 Global and local error metrics

We start choosing a reference pixel in the original depth map and mark (with 1) all pixels in the map that are greater than or equal to the depth value at that pixel. All other pixels remain unmarked (with 0). We repeat this process for the estimated depth map using the same reference pixel, as seen in Fig. 6. We then compute a similarity map by comparing per-pixel values in both previous maps, obtaining final values of 1 only for matching pixel values. An accuracy value for the reference pixel is computed by taking the sum of all values in the similarity map and dividing it by the total number of pixels of the map. So values closer to 1 are more accurate than the ones closer to 0. This process is repeated for each pixel in the depth maps to obtain accuracy maps as seen on the right in Fig. 5.

Journal: 371 MS: 1050 TYPESET DISK LE CP Disp.:2014/11/17 Pages: 11 Layout: Large

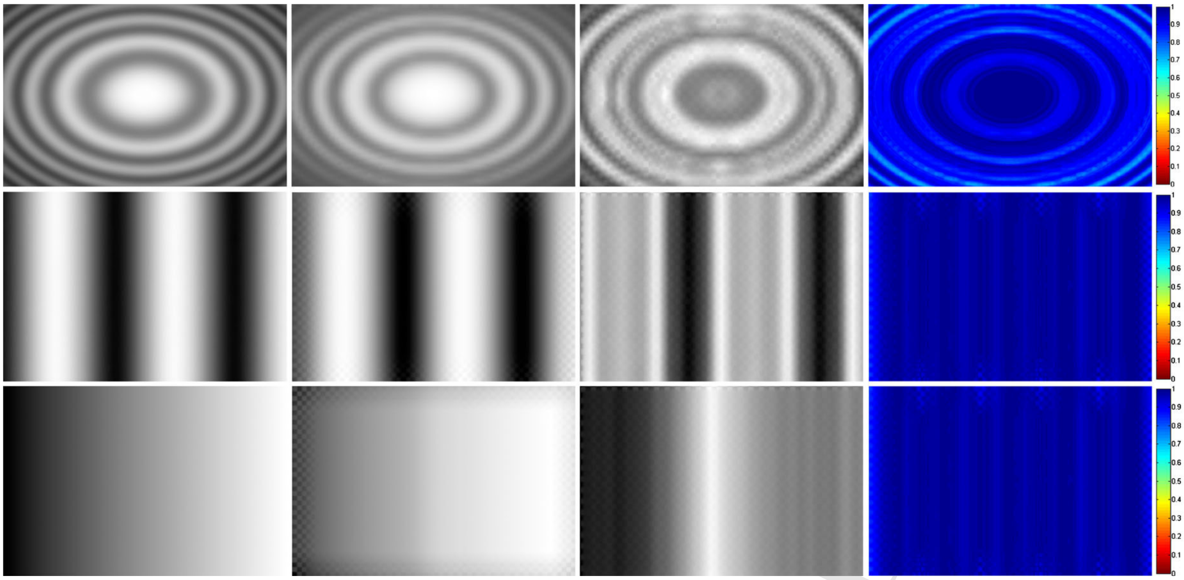

Fig. 5 Comparison of original depth maps (*on the left*) with our estimations (*middle left*). Local error from the curve fitting step (*middle right*) where the errors ranged between a magnitude of 10^{-9} and 10^{-8} (*black and white*, respectively, for better visualization), and our global accuracy metric (*right*). In this last case, a value of one means a perfect match. Our local and global accuracy metrics clearly show that while local errors may occur, the reconstructed global shape of the object has a good resemblance with the ground truth one, as appreciated also in Fig. 4

In addition to our global accuracy metric, we can also obtain per-pixel error maps from the optimization step. Such maps show the squared error obtained when fitting Eq. 1 to the estimated blur values for one pixel through the focal stack to obtain its final depth value. Examples of these maps can be found in Fig. 5 (middle right).

Looking at the blur estimates used for the optimization reveals that small blurs were over-estimated while large blurs were under-estimated. These inaccuracies caused the algorithm to compress the depth estimates such that the range of estimated depths is smaller than the actual range. However, since blur estimate errors are consistent across the entire image, the depth estimates are still accurate relative to each other, and so the global shape captures the main features of the ground truth.

5.3 Real scenes

We also tested our algorithm with real scenes. We again used examples from prior work [6,8,30] to allow direct visual comparisons with our results. In these examples, the number of images for each focal stack is two. As can be seen in Fig. 7, we obtain plausible reconstructions comparing favorably with both Watanabe and Nayar [30] and Favaro [8], even though our depth maps look blurrier due to the filtering explained in Sect. 4.3. Our work presents an interesting tradeoff between accuracy and speed, as it is significantly faster than the 10 min reported in [6]

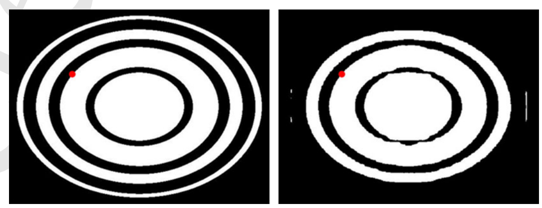

Fig. 6 Example of estimating the global accuracy of a pixel (marked in *red*) for the wave object from Fig. 4. Pixels with depth values greater or equal to it are marked in *white*, while the rest keep unmarked (*black*). This is done for both the ground truth depth map (*left*) and the estimated depth map (*right*). A similarity measure for that pixel is then computed by marking with one all the pixels with matching values and dividing that number by the total size of the map

Additional examples from real scenes can be found in Fig. 8. The first two rows show plausible reconstructions for different stuffed toys. The bottom row shows a difficult case for our algorithm. Given the asymptotic behavior of the circle of confusion function (Fig. 2), objects from a certain distance show small differences in blur. Since our blur estimations are not in real scale, this translates into either unrelated distant points recovered into the same background plane, or inaccurate and different depth values for neighboring pixels. This happens usually in outdoor scenes, so our algorithm is better suited for close-range scenes.

6 Conclusions

In this paper, we have presented an algorithm that estimates depth from a focal stack of images. This algorithm uses

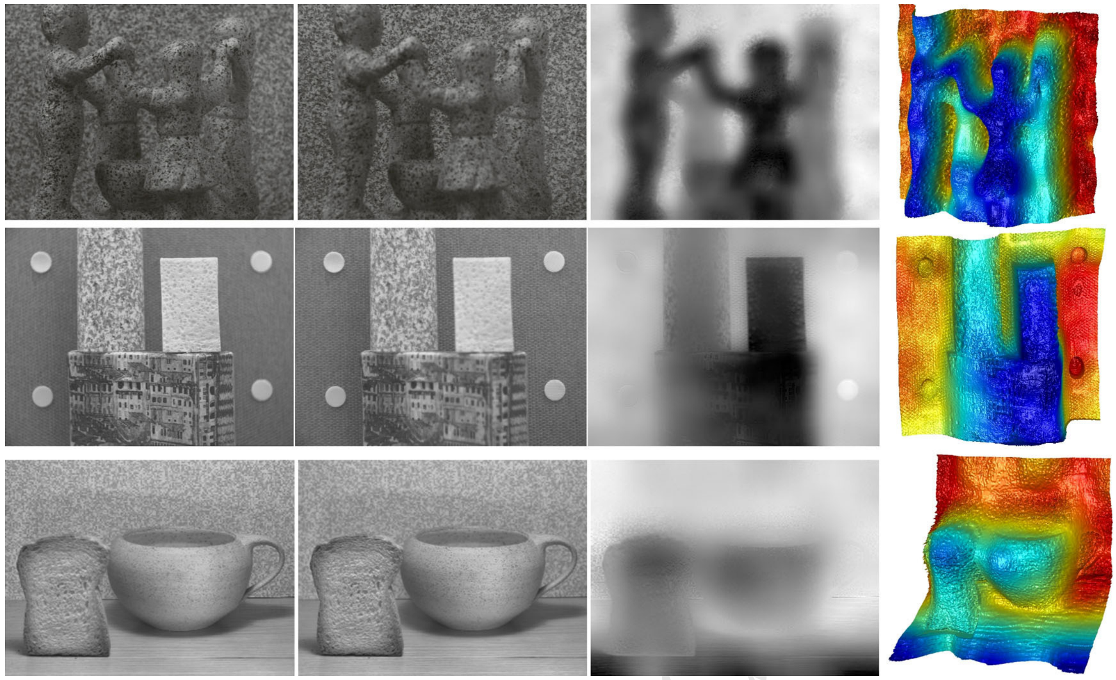

Fig. 7 Close focus (*left*), Far focus (*middle left*), our estimated depth map (*middle right*), and its corresponding 3D visualization (*right*). *Colors* and *shading* added for a better visualization

Fig. 8 Close focus (*left*), Far focus (*middle left*), our estimated depth map (*middle right*), and its corresponding 3D visualization (*right*). *Colors* and *shading* added for a better visualization. The estimated depth map for the top scene used parameters $f = 24$ mm, $f/8$, close focal distance of 0.35 m, and far focal distance of 0.75 m. The estimated depth map for the middle scene used parameters $f = 26$ mm, $f/8$, close focal distance of 0.4 m, and far focal distance of 0.8 m. The estimated depth map for the bottom scene used parameters $f = 18$ mm, $f/8$, close focal distance of 0.5 m, and far focal distance of 15 m

a least squares optimization to obtain depth values from a set of pixel measurements up to a scale. We have shown that it compares well with prior work but runs significantly faster.

As mentioned previously, our algorithm possesses some limitations. The focus measure we employed [12] has difficulties in estimating large blur radii, producing an undesired flattening of the estimated depth map. It would be interest-

Journal: 371 MS: 1050 TYPESET DISK LE CP Disp.:2014/11/17 Pages: 11 Layout: Large

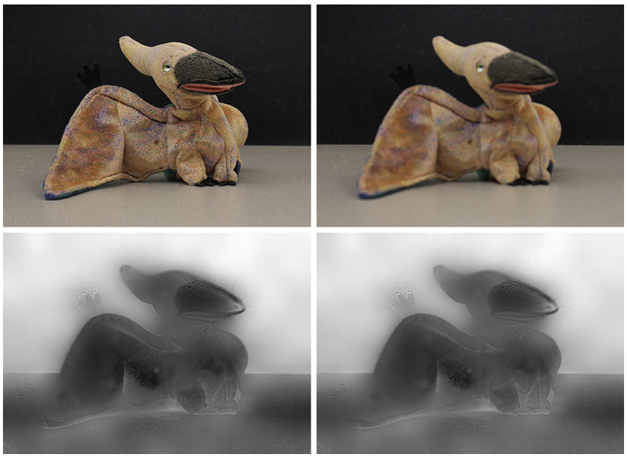

Fig. 9 Comparison between accurate and estimated focus positions. *Top* Input images captured with focal distances of 0.4 m (*left*), and 0.8 m (*right*). *Bottom left* estimated depth map using those focal distances. *Bottom right* results using estimates of 0.3 and 1.0 m, respectively. As can be seen, our algorithm can handle small inaccuracies robustly

ing to test other measures included in Pertuz et al. [24] to see their effect. In Fig. 9, we show that our algorithm can robustly handle small inaccuracies in focal distances, and it would be interesting to analyze the effect of these inaccuracies in future work. Also, the guided filter [10] used as the kernel for the normalized convolution shows texture-copy artifacts sometimes, given the suboptimal use of the color images as the guides for the filter. However, it is not clear what could be a good guide for this tasks, with possible choices like intrinsic images [31] being ill-posed problems that may introduce their own artifacts. Finally, while our current optimization step is already using interpolated blur data that took into account the confidence of each sample, it could be interesting to combine those confidence values to place additional constraints during this step.

We believe our method presents an interesting tradeoff between accuracy and speed when compared with previous works. The modularity of our approach makes it straightforward to study alternatives to the chosen algorithms at each step, so it can greatly benefit from separate advances that occur in the future.

Acknowledgments The authors thank T. S. Choi and Paolo Favaro for sharing their data sets. This work has been supported by the European Union through the projects GOLEM (grant agreement no.: 251415) and VERVE (grant agreement no.: 288914), as well as by the Gobierno de Aragon through the TAMA project. This material is based upon work supported by the National Science Foundation under Grant Nos. 0705863 and 1116988.

Appendix A: Least squares function analysis

In this appendix, we show how to cast the depth estimation problem as an optimization problem. Consider the optimization problem for a single signal with n blur estimates, and each c_i is captured with a focal position S_1^i. Let

$$g_i(x) = \left(c_i - A \frac{|x - S_1^i|}{x} \frac{f}{S_1 - f} \right)^2 \tag{13}$$

The function $g_i(x)$ has a critical point at S_1^i because the derivative at S_1^i of $g_i(x)$ does not exist due to the term $|x - S_1^i|$ in the function. Furthermore, if the blur estimate c_i is less than the circle of confusion size

$$c = \frac{f^2}{N(S_1^i - f)} \tag{14}$$

for a depth x at infinity, then the function will have two local minimizers, as shown in Fig. 10, at the points $g(x) = 0$ where

$$x = \frac{S_1^i A f}{A f - c_i (S_1^i - f)} \tag{15}$$

and

$$x = \frac{S_1^i A f}{A f + c_i (S_1^i - f)}. \tag{16}$$

However, if $c_i = 0$ then the function will have one minimizer at $x = S_1^i$, and similarly if x is larger than the circle of confusion size for a depth at infinity, then $g_i(x)$ will have only one minimizer somewhere within the interval $(0, S_1^i)$.

For the purposes of optimization, we assume that

$$0 < c_i < \frac{f^2}{N(S_1^i - f)}. \tag{17}$$

This assumption introduces the restriction that the depth of a signal in the focal stack cannot be too close to the lens.

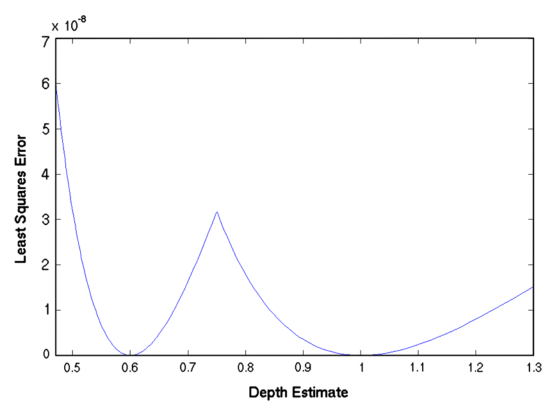

Fig. 10 Plot of $g_i(x)$ showing a local maximizer at the point $S_1^i = 0.75$ m and two local minimizers on either size of the maximizer

A further restriction for the depth x is that $S_1^1 < x < S_1^n$ where $0 < S_1^1 < S_1^2 < \cdots < S_1^n$. This restriction limits the depth of any point in the focal stack to be between the closest focal position of the lens and the farthest focal position.

With these assumptions, we can now look at the least squares optimization equation

$$z(x) = \sum_{i=1}^{n} g_i(x). \tag{18}$$

Because each $g_i'(x)$ is undefined at $x = S_1^i$ for all $i = 1, \ldots, n$, the function $z(x)$ has critical points at $S_1^1, \ldots, S_1^n$. Furthermore, $z(x)$ is continuous everywhere else for $x > 0$ because the functions $g_i(x)$ are continuous where $x > 0$ and $x \neq S_1^i$. Because $g_i(x)$ has a local maximizer at S_1^i, this point may be a local maximizer for $z(x)$. This gives us $n-1$ intervals on which $z(x)$ is continuous for $S_1^1 < x < S_1^n$, and these intervals are $(S_1^1, S_1^2), (S_1^2, S_1^3), \ldots, (S_1^{n-1}, S_1^n)$. These open intervals may or may not contain a local minimizer, and if an interval does contain a local minimizer, it might be the global minimizer of $z(x)$ on the interval (S_1^1, S_1^n).

Under certain conditions, $z(x)$ is convex within the interval (S_1^1, S_1^{i+1}) for all $i = 1, \ldots, n-1$. Note that $g_j(x)$ is convex within the open interval for all $j = 1, \ldots, n$. To see this, assume that Eq. 11 holds and that the focus position of the lens is always greater than the focal length f of the lens so that $r > 0$. We also assume that

$$S_1^n \leq \frac{3rS_1^j}{2c_j}. \tag{19}$$

If $x < S_1^j$ then the absolute value term $|x - S_1^j|$ in $g_i(x)$ becomes $-x + S_1^j$. From this, we know that

$$rS_1^j \geq 2c_j x \tag{20}$$

from Relation 11 and because x and S_1^j are positive. Rearranging the relation, we get

$$-2c_j S_1^j + rS_1^j \geq 0. \tag{21}$$

Since $x < S_1^j$, $2rx < 2rS_1^j$ and $2rS_1^j - 2rx > 0$. Therefore,

$$\begin{aligned} -2c_j x + 3rS_1^j - 2rx &= -2c_j x + rS_1^j + (2rS_1^j - 2rx) \\ &\geq 2rS_1^j - 2rx \\ &> 0 \end{aligned} \tag{22}$$

Furthermore, since $x > 0$, $r > 0$, and $S_1^j > 0$, we know that

$$\frac{2rS_1^j}{x^4} > 0. \tag{23}$$

Therefore, we know that

$$g_j''(x) = \frac{2rS_1^j(-2c_j x + 3rS_1^j - 2rx)}{x^4} > 0 \tag{24}$$

for $0 < x < S_1^j$.

If $x > S_1^j$, then

$$x < S_1^n \leq \frac{3rS_1^j}{2c_j} \tag{25}$$

from Eq. 19 and that $x < S_1^n$. Since $c_j > 0$, we can multiply the relation by $2c_j$ to get

$$3rS_1^j > 2c_j x. \tag{26}$$

From relation (11), we can say that

$$2r - 2c_j \geq 4c_j - 2c_j = 2c_j. \tag{27}$$

Therefore,

$$3rS_1^j > x(2r - 2c_j) \geq x(2c_j). \tag{28}$$

Distributing x in the above relation, we get

$$3rS_1^j > 2rx - 2c_j x \tag{29}$$

Rearranging the terms, we get

$$2c_j x + 3rS_1^j - 2rx > 0. \tag{30}$$

Multiplying by the left-hand side of (23), we get

$$g_j''(x) = \frac{2rS_1^j(2c_j x + 3rS_1^j - 2rx)}{x^4} > 0 \tag{31}$$

for $S_1^j < x < S_1^n$.

As shown above, the second derivative of $g_j(x)$ is always positive on the interval (S_1^1, S_1^n) except at the point S_1^j for all $j = 1, \ldots, n$. Since $z(x)$ is the summation of all $g_j(x)$, $z(x)$ is also convex on the interval except at the points $S_1^1, S_1^2, \ldots, S_1^n$. Therefore, $z(x)$ is convex in the intervals (S_1^i, S_1^{i+1}) for all $i = 1, 2, \ldots, n-1$. As a consequence, if S_1^i, and S_1^{i+1} are local maximizers, then there is some local minimizer within the open interval (S_1^1, S_1^n). From this, a global minimizer can be identified which gives the best depth estimate for the given signal on the interval (S_1^1, S_1^n). Figure 11 shows an example of $z(x)$ with the local maximizers and minimizers.

Journal: 371 MS: 1050 TYPESET DISK LE CP Disp.:2014/11/17 Pages: 11 Layout: Large

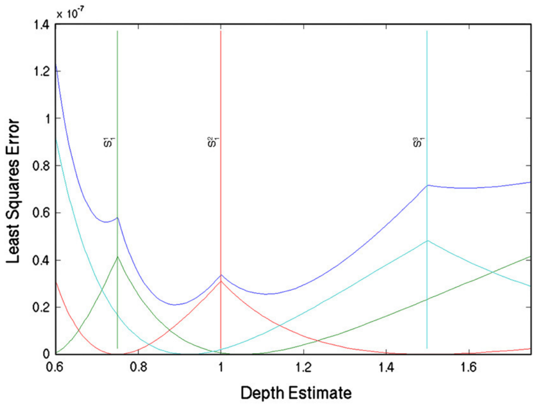

Fig. 11 Plot of $z(x)$ shown in show *dark blue* with $g_1(x)$, $g_2(x)$, and $g_3(x)$ shown in *red*, *light blue*, and *green*, respectively. This shows $z(x)$ with local maximizers at $S_1^1 = 0.75$, $S_1^2 = 1$, and $S_1^3 = 1.5$ and local minimizers in the intervals (S_1^1, S_1^2) and (S_1^2, S_1^3)

References

1. Bae, S., Durand, F.: Defocus magnification. Comput. Graph. Forum **26**(3), 571–579 (2007)
2. Bauszat, P., Eisemann, M., Magnor, M.: Guided image filtering for interactive high-quality global illumination. Comput. Graph. Forum **30**(4), 1361–1368 (2011)
3. Calderero, F., Caselles, V.: Recovering relative depth from low-level features without explicit t-junction detection and interpretation. Int. J. Comput. Vis. 1–31 (2013)
4. Cao, Y., Fang, S., Wang, F.: Single image multi-focusing based on local blur estimation. In: Image and graphics (ICIG), 2011 Sixth International Conference on, pp. 168–175 (2011)
5. Cao, Y., Fang, S., Wang, Z.: Digital multi-focusing from a single photograph taken with an uncalibrated conventional camera. Image Process. IEEE Trans. **22**(9), 3703–3714 (2013). doi:10.1109/TIP.2013.2270086
6. Favaro, P.: Recovering thin structures via nonlocal-means regularization with application to depth from defocus. In: Computer vision and pattern recognition (CVPR), 2010 IEEE Conference on, pp. 1133–1140 (2010)
7. Favaro, P., Soatto, S.: 3-D Shape Estimation and Image Restoration: Exploiting Defocus and Motion-Blur. Springer-Verlag New York Inc, Secaucus (2006)
8. Favaro, P., Soatto, S., Burger, M., Osher, S.J.: Shape from defocus via diffusion. Pattern Anal. Mach. Intel. IEEE Trans. **30**(3), 518–531 (2008)
9. Hasinoff, S.W., Kutulakos, K.N.: Confocal stereo. Int. J. Comput. Vis. **81**(1), 82–104 (2009)
10. He, K., Sun, J., Tang, X.: Guided image filtering. In: Proceedings of the 11th European conference on Computer vision: Part I. ECCV'10, pp. 1–14. Springer, Berlin, Heidelberg (2010)
11. Hecht, E.: Optics, 3rd edn. Addison-Wesley (1997)
12. Hu, H., De Haan, G.: Adaptive image restoration based on local robust blur estimation. In: Proceedings of the 9th international conference on Advanced concepts for intelligent vision systems. ACIVS'07, pp. 461–472. Springer, Berlin, Heidelberg (2007)
13. Knutsson, H., Westin, C.F.: Normalized and differential convolution: Methods for interpolation and filtering of incomplete and uncertain data. In: Proceedings of Computer vision and pattern recognition ('93), pp. 515–523. New York City, USA (1993)
14. Lee, I.H., Shim, S.O., Choi, T.S.: Improving focus measurement via variable window shape on surface radiance distribution for 3d shape reconstruction. Optics Lasers Eng. **51**(5), 520–526 (2013)
15. Levin, A., Fergus, R., Durand, F., Freeman, W.: Image and depth from a conventional camera with a coded aperture. ACM Transactions on Graphics, SIGGRAPH 2007 Conference Proceedings, San Diego, CA (2007)
16. Li, C., Su, S., Matsushita, Y., Zhou, K., Lin, S.: Bayesian depth-from-defocus with shading constraints. In: Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, pp. 217–224 (2013). doi:10.1109/CVPR.2013.35
17. Lin, X., Suo, J., Wetzstein, G., Dai, Q., Raskar, R.: Coded focal stack photography. In: IEEE International Conference on Computational photography (2013)
18. Mahmood, M.T., Choi, T.S.: Nonlinear approach for enhancement of image focus volume in shape from focus. Image Process. IEEE Trans. **21**(5), 2866–2873 (2012)
19. Malik, A.: Selection of window size for focus measure processing. In: Imaging systems and techniques (IST), 2010 IEEE International Conference on, pp. 431–435 (2010)
20. Moreno-Noguer, F., Belhumeur, P.N., Nayar, S.K.: Active refocusing of images and videos. In: ACM SIGGRAPH 2007 papers, SIGGRAPH '07. ACM, New York, NY, USA (2007)
21. Namboodiri, V., Chaudhuri, S.: Recovery of relative depth from a single observation using an uncalibrated (real-aperture) camera. In: Computer vision and pattern recognition, 2008. CVPR 2008. IEEE Conference on, pp. 1–6 (2008)
22. Nayar, S., Nakagawa, Y.: Shape from focus. Pattern Anal. Mach. Intel. IEEE Trans. **16**(8), 824–831 (1994)
23. Pentland, A.P.: A new sense for depth of field. Pattern Anal. Mach. Intel. IEEE Trans. PAMI **9**(4), 523–531 (1987)
24. Pertuz, S., Puig, D., Garcia, M.A.: Analysis of focus measure operators for shape-from-focus. Pattern Recognit. **46**(5), 1415–1432 (2013)
25. Petschnigg, G., Szeliski, R., Agrawala, M., Cohen, M., Hoppe, H., Toyama, K.: Digital photography with flash and no-flash image pairs. ACM SIGGRAPH 2004 Papers. SIGGRAPH '04, pp. 664–672. ACM, New York, NY, USA (2004)
26. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical Recipes: The Art of Scientific Computing, 3rd edn. Cambridge University Press (2007)
27. Shim, S.O., Choi, T.S.: A fast and robust depth estimation method for 3d cameras. In: Consumer Electronics (ICCE), 2012 IEEE International Conference on, pp. 321–322 (2012)
28. Subbarao, M., Choi, T.: Accurate recovery of three-dimensional shape from image focus. Pattern Anal. Mach. Intel. IEEE Trans. **17**(3), 266–274 (1995)
29. Vaquero, D., Gelfand, N., Tico, M., Pulli, K., Turk, M.: Generalized autofocus. In: IEEE Workshop on Applications of Computer Vision (WACV'11). Kona, Hawaii (2011)
30. Watanabe, M., Nayar, S.: Rational filters for passive depth from defocus. Int. J. Comput. Vis. **27**(3), 203–225 (1998)
31. Zhao, Q., Tan, P., Dai, Q., Shen, L., Wu, E., Lin, S.: A closed-form solution to retinex with nonlocal texture constraints. Pattern Anal. Mach. Intel. IEEE Trans. **34**(7), 1437–1444 (2012)
32. Zhou, C., Cossairt, O., Nayar, S.: Depth from diffusion. In: IEEE Conference on Computer vision and pattern recognition (CVPR) (2010)
33. Zhuo, S., Sim, T.: On the recovery of depth from a single defocused image. In: X. Jiang, N. Petkov (eds.) Computer Analysis of Images and Patterns, Lecture Notes in Computer Science, vol. 5702, pp. 889–897. Springer, Berlin Heidelberg (2009). doi:10.1007/978-3-642-03767-2_108. URL http://dx.doi.org/10.1007/978-3-642-03767-2_108
34. Zhuo, S., Sim, T.: Defocus map estimation from a single image. Pattern Recognit. **44**(9), 1852–1858 (2011)

Journal: **371** MS: **1050** ☐ TYPESET ☐ DISK ☐ LE ☐ CP Disp.:**2014/11/17** Pages: **11** Layout: **Large**