Int J Comput Vis
DOI 10.1007/s11263-011-0476-5

Towards Unrestrained Depth Inference with Coherent Occlusion Filling

Arnav V. Bhavsar · A.N. Rajagopalan

Received: 18 October 2010 / Accepted: 9 June 2011

Abstract Traditional depth estimation methods typically exploit the effect of *either* the variations in internal parameters such as aperture and focus (as in depth from defocus), *or* variations in extrinsic parameters such as position and orientation of the camera (as in stereo). When operating off-the-shelf (OTS) cameras in a general setting, these parameters influence the depth of field (DOF) and field of view (FOV). While DOF mandates one to deal with defocus blur, a larger FOV necessitates camera motion during image acquisition. As a result, for unfettered operation of an OTS camera, it becomes inevitable to account for pixel motion as well as optical defocus blur in the captured images. We propose a depth estimation framework using calibrated images captured under general camera motion *and* lens parameter variations. Our formulation seeks to generalize the constrained areas of stereo and shape from defocus (SFD)/focus (SFF) by handling, in tandem, various effects such as focus variation, zoom, parallax and stereo occlusions, all under one roof. One of the associated challenges in such an unrestrained scenario is the problem of removing user-defined foreground occluders in the reference depth map and image (termed inpainting of depth and image). Inpainting is achieved by exploiting the cue from motion parallax to *discover* (in other images) the correspondence/color information missing in the reference image. Moreover, considering the fact that the observations could be differently blurred, it is important to ensure that the degree of defocus in the missing regions (in the reference image) is coherent with the local neighbours (*defocus inpainting*).

Keywords Defocus blur · Multi-view stereo · Depth from defocus · Belief propagation · Inpainting

1 Introduction

One of the most challenging tasks in computer vision is computing the depth of objects in the 3D world from their 2D images captured by a camera. Numerous computer vision techniques exploit a variety of cues to infer depth information. Typically, the cues may be classified as those embedded in the scene itself, such as texture, illumination, perspective etc, and those induced by the image capturing device, such as pixel-motion and blur/accommodation. Here, we concern ourselves with the latter category i.e., depth cues which are controllable and are functions of the camera parameters.

We believe that, in general, the depth estimation task need not be confined to some restrained camera parameter settings or configurations. One must be at liberty to flexibly operate a camera since some of the assumptions about the quality and content of acquired image may not be satisfied for OTS cameras.

In spite of the immense progress witnessed in the manufacturing of cameras, there still exists practical limits on the intrinsic parameters such as focal length f, working distance u and aperture radius $r = f/(2f_n)$ (where f_n denotes the f-number of the lens), that impact the depth-of-field (DOF) and field-of-view (FOV) (angle of view β). It is well-known that the DOF can be expressed as

$$DOF = \frac{2uf_n cf^2(u-f)}{f^4 - f_n^2 c^2 (u-f)^2} = \frac{2f^2 f_n c(\mu+1)}{f^2\mu^2 - f_n^2 c^2} \tag{1}$$

A.V. Bhavsar (✉) · A.N. Rajagopalan
Image Processing and Computer Vision Lab, Dept. of Electrical Engineering, Indian Institute of Technology, Madras, India
e-mail: arnav.bhavsar@gmail.com

A.N. Rajagopalan
e-mail: raju@ee.iitm.ac.in

Published online: 06 July 2011

and is inversely related to u, f and r. Here, c denotes the circle of confusion (which is typically fixed for a particular sensor size) and μ denotes the magnification. Similarly, the angle of view (or field of view) can be expressed as

$$\beta = 2 \arctan \frac{d(u-f)}{2uf} \tag{2}$$

which is also inversely related to f (and hence the magnification). Here d is the dimension of the sensor (usually taken as the larger side). The FOV depends on β and the aperture radius.

From these relations, it is possible to make some deductions that clearly reflect the effects that camera parameters have on the acquired images.

- Moving the camera closer to the objects, to acquire sufficient shape and image detail, reduces the field of view. Also, since the object is relatively close to the camera, it can lie beyond the range of the nearer DOF and will get defocused.
- Another way to acquire sufficient resolution by not going closer to the object physically is to increase the magnification. However, increasing the magnification yields a larger focal length, which will in turn reduce the FOV. This will also decrease the nearer DOF limit, increasing the possibility of the image getting defocused.
- To increase the FOV, one either has to increase the aperture size or reduce focal length. Reducing the focal length will reduce the magnification which will reduce the resolution. Increasing the aperture will keep the resolution constant but decrease the DOF leading to increased defocus.

In Table 1, we provide some practical values for a standard Olympus digital camera (Wrotniak 2003, 2006). When $u = 20$ cm, if the object is kept less than 16 cm or more than 23 cm from the camera, then its image will appear defocused. Moreover, for macro-imaging (typically used for operating in the range $u = 3$ to 20 cm), the DOF is even smaller with the FOV covering around 5 cm of the real world.

Thus, it is clear that in practical scenarios there are trade-offs among acquiring sufficient detail, content and image quality. Often one cannot avoid defocusing of images (although blurring is sometimes even desirable for visual effects) and a compromise on spatial resolution and content. There also arise situations where it is necessary to move the camera (e.g., to capture a larger field-of-view), by varying the extrinsic parameters viz. rotation and translation. Thus, it is necessary that the depth estimation framework should be able to handle unrestrained operation of the camera by considering parameter variations and the resultant image-level effects, such as focusing/defocusing, zooming in or out, motion parallax, occlusions etc.

Table 1 Examples of camera parameters and depth of field for Olympus C-5050 ($d = 7.2$ mm × 5.3 mm and $f = 14.2$ mm)

$u = 50$ cm		$u = 20$ cm	
$f/$	*DOF* (cm)	$f/$	*DOF* (cm)
2	48.6–51.5	2	19.1–21
4	47.2–53.1	4	18.2–22.1
5.6	46.2–54.5	5.6	17.6–23.1
8	44.7–56.7	8	16.8–22.8

In this work, we address the problem of formulating a general framework for inferring dense shape using observations acquired from a calibrated setup involving an unrestrained but controllable camera. Acknowledging the existence of practical effects of camera parameters, our framework elegantly couples the defocus blur and pixel motion, induced by freely operating the camera, since both carry important information about the shape of the 3D world. This essentially encompasses the so called low-level vision task of shape estimation exploiting the camera induced effects (and visual cues) of defocus blur and/or motion, whenever one or both of them are present in the observed images.

The depth cue in defocus blur has been exploited in works on depth from defocus (DFD) (Rajagopalan and Chaudhuri 1999; Favaro and Soatto 2005; Favaro et al. 2008) and shape from focus (SFF) (Nayar and Nakagawa 1994; Sahay and Rajagopalan 2008). However, such approaches impose strong restrictions on camera parameters and motion. We relax such constraints and handle the blurring under general camera parameter variation as well as general camera motion. Moreover, we also consider the pixel motion caused due to the variations in the intrinsic camera parameters (e.g., the zooming effect). Such a level of generalization in considering the blurring effect is not yet addressed in DFD (Favaro and Soatto 2005; Favaro et al. 2008), SFF (Sahay and Rajagopalan 2008) or related works.

Camera motion provides additional depth information (Scharstein and Szeliski 2002; Strecha et al. 2004; Seitz et al. 2006), but this has been exploited mainly for the pin-hole camera and not so much for real-aperture ones. The pin-hole model is not valid when one operates at distances beyond the DOF of the camera (e.g., close-up pictures, high zooming, macro-photography etc). Our approach considers defocus blurring in the process of finding correspondences. Moreover, camera motion brings with it a baggage of additional issues of occlusions and visibilities due to motion parallax. We account for the *stereo occlusion* effect which involves those points which are seen in the reference image but not in some of the other images.

We also address the occlusion problem where the points not seen in the reference view, (due to user-defined foreground objects) are visible in some of the other images (we term this as *view occlusion*). This problem can also be viewed as one of *inpainting* depth and image for finding missing background information in the reference view. Moreover, since we work with a practical real aperture camera, the estimation of missing depth must also consider the variation in blur across images due to the 3D nature of the scene. Removing view occlusions involves 'uncovering' the background by removing the occluding objects. Since, removing occluders will change the scene content, the depth map as well as the image must be estimated at the missing regions. We consider two cases here: (1) when the occluded missing regions in the reference image are due to objects which are part of the scene, and (2) when the occluders are part of the camera (e.g., local damage to the lens/imaging sensor).

The motion cue facilitates information missing in some images to be present in others. This includes the correspondence information for estimating depth as well as the color information for estimating image in the missing areas. Since our framework considers camera motion, it naturally exploits the motion cue to address the view occlusion/inpainting problem. Accounting for defocus blur in the image inpainting implies that despite the fact that observations have different degrees of blur, a pixel should be inpainted coherently according to the amount of blur in its neighbourhood. We term this as *defocus inpainting*, since it also considers defocus blur while computing the inpainted image. We also observe that *defocus inpainting* involves filling unknown information in one of the views (using information from others), which essentially tantamounts to *defocused novel view synthesis* (albeit, in this case, only for missing regions).

Note that the task of inpainting as discussed here is one of occlusion removal. Since disocclusion in this work essentially involves depth cues stemming from parallax and defocus blur, inpainting is a natural fall-out of our depth estimation process. Thus, our work also serves to highlight the tight coupling between the related problems of depth estimation and inpainting for an unconstrained real-aperture camera.

In summary, our generalized approach accounts for camera variations such as focus, zooming, parallax, and camera motion for depth estimation. The framework can handle stereo occlusions as well as remove view occlusions. We also perform defocus image inpainting which can be interpreted as a (restricted) version of the more general problem of defocused novel view synthesis.

1.1 Relation to Previous Work

There is a considerable amount of literature on calibrated shape estimation under defocus blur and camera motion, considered *individually*. These works which are typically categorized under the aegis of stereo, depth from defocus and shape from focus, are in fact, special restrictive cases of the general scenario that we consider; for example, stereo works with a pin hole model with a point-like aperture; the blur due to intrinsic parameter variations is not considered. DFD assumes that the camera is static and only the intrinsic lens parameters are varied. SFF enforces the intrinsic parameters to be constant and only allows for axial motion of the camera. SFF also ignores pixel motion by assuming the lens to be tele-centric. Thus, in these cases either only the pixel motion or the blur cue is used.

Few works have been reported wherein both blurring and motion effects (or cues) are taken into account in the same framework. In Myles and Lobo (1998), the authors have proposed a technique to compute the affine motion and the defocus blur simultaneously in images of planar scenes. The work has been extended to estimate defocus blur and arbitrary space-variant spatial shifts in Deschenes et al. (2004). Recently, the work in Seitz and Baker (2009) also handles global affine transform to model the zooming process in the depth from defocus problem. However, in these works, only the defocus blur is actually exploited as a cue for shape.

Some works (Subbarao et al. 1997; Frese and Gheta 2006) have motivated the problem of shape estimation from a cue combination point of view. The authors in Subbarao et al. (1997) carry out a sequential process where the shape is first estimated using local DFD and SFF techniques. This shape estimate is used to constrain the stereo correspondence problem and is improved by solving the stereo problem on pin-hole images. Another sequential approach is proposed in Frese and Gheta (2006) where a depth map computed with a contemporary stereo technique on pin hole images is further improved by an SFF technique on a focus series. In their case, the images in focus-series are not coincident (as is required in traditional SFF). The SFF algorithm is modified to take into account the pixel shift, the information about which is available from the stereo depth map. Apart from these works, some cooperative active depth estimation approaches have also been reported (Ahuja and Abbot 1993; Krotkov and Bajcsy 1993) that control the focus and vergence angles in an active manner to facilitate image acquisition suitable for depth from focus and depth from stereo. The depth maps obtained from each are improved by that obtained from the other. The motivation for these active approaches arises due to the claim that stereo-based and focus-based shape estimation act in a complementary manner when one considers aspects such as occlusions, depth of field, feature localization etc. Although these techniques utilize both the motion and the defocus cues for shape estimation, they sequentially combine the conventional techniques of stereo, SFF and DFD and hence work with image configurations that are tailored for these specific techniques.

Recently, an approach that integrates structure from motion with the depth from defocus problem, has been reported in Wohler et al. (2009). It involves a priori tracking of points and computation of blur at those tracked points. This is followed by a cost minimization that compares the projection of 3D points to the tracked image points and also compares the depth of the projected points to the a priori computed blur parameter at the tracked points. This technique computes a sparse structure. Also, the a priori blur parameter computation is quite heuristic. Moreover, this technique (and, in fact, all the above mentioned methods) do not exploit the relationship between blur, motion and depth in an integrated framework as in our case.

In recent years, some works have considered the interrelationship between the blur and the motion since both are related to the depth (Rajagopalan et al. 2004; Kim and Sikora 2007; Duo and Favaro 2008). In Duo and Favaro (2008), a novel approach is suggested to create a virtual stereo pair by using a special camera in which, in addition to the aperture size, the aperture position with respect to the lens can be varied laterally. This gives rise to both motion parallax and defocus blur in the observed image. However, this configuration is different and has limited freedom than the more general camera variations that we consider. Also, this requires a special camera setup, that may not always be feasible with conventional off-the shelf cameras. The authors in Kim and Sikora (2007) consider a depth estimation and image restoration problem similar to ours in a binocular stereo scenario with lateral motion. Here, multiple (around 10) differently focused images are captured from each view and a window to constraint the disparity estimation is computed using the blur disparity relationship. Although this method uses the blur motion (disparity) relationship, it only exploits it locally to define windows for stereo disparity estimation.

Closely related to our work are those in Rajagopalan et al. (2004), Sahay and Rajagopalan (2009b) which use the blur-disparity relationship in a global regularization framework to compute depth. Whereas the work of Rajagopalan et al. (2004) uses binocular stereo images with lateral camera translation and captures two differently focused images from each view, the authors in Sahay and Rajagopalan (2009b) follow axial camera translation with no intrinsic parameters variation (a commonly used setting in SFF). Thus, these methods involve very restricted camera configurations.

None of the above mentioned works consider handling stereo occlusions; especially, the problem of removing / inpainting the view occluders. The former is addressed in numerous stereo approaches, but not in the works which also consider defocus blur. The view occlusion (inpainting) problem has been mainly reported on a single image level for color and range images (Criminisi et al. 2003; Bhavsar and Rajagopalan 2008), individually. These are essentially based on some heuristics for establishing continuities in some sense by considering neighbourhood, edges, etc. Our occluder inpainting methodology uses multiple views and searches (in different views) for the actual information, which is lost in the reference image. Thus, our inpainting approach involves more of looking for plausibly correct information and less of heuristics. In fact, there is very little work on inpainting both images and depth. A recent work uses a stereo setting (with pin-hole cameras) to inpaint occluders present in the scene, in both image and the depth map (Wang et al. 2008). Unlike Wang et al. (2008) which uses a binocular setup, our approach uses multiple images and involves a different pixel mapping. Also, we address the cases of removing occluders in the scene as well as missing regions due to sensor/lens damage, unlike Wang et al. (2008) which only considers the former. Moreover, akin to traditional inpainting approaches, the method in Wang et al. (2008) does not account for camera defocus. Some approaches do consider the camera defocus while inpainting distortions in the images (Zhou and Lin 2007; Gu et al. 2009). However, in these works the images are acquired from a single view and address only image inpainting. The works in Sahay and Rajagopalan (2009a), Sahay and Rajagopalan (2010) considers both motion and blur in the restrictive setting of axial motion and no camera parameter variations. On the other hand, our approach involves defocus and pixel-motion under a very general setting, considers visibility and segmentation constraints, and is also computationally efficient.

The novelties of our work as compared to the above mentioned works can be summarized as follows. (1) Our image acquisition process offers more freedom so as not to be constrained by a strict DFD, SFF or a stereo-like setting. We consider general camera motion and parameter variations. (1a) For intrinsic parameters, we consider variations in the aperture, working distance and lens-to-image-plane distance. Our framework also accommodates the zooming process caused by variations in the latter. We also account for the situation where the scene does not completely lie in one part of the blur cone (the *near-far* focusing scenario). (1b) Unlike in most of the above works, we consider all 6 degrees of freedom for camera motion. An important consequence of this is in considering the effect of camera rotation on defocus blur variation. (2) We also address the important issue of occlusions that has not been addressed in other works. (2a) The view-occlusion case is, in fact, equivalent to depth and image inpainting which is another novel addition. We account for two kinds of occluders while considering motion as well as the blur cue. (3) We also use the vital cue from color image segmentation during the depth estimation and inpainting process; a cue that has been shown to be very successful in conventional stereo works (Yang et al. 2009).

We would like to add that this work is a comprehensive extension of our own related and previously published works (Bhavsar and Rajagopalan 2009, 2010). The work in

Bhavsar and Rajagopalan (2009) reports only depth estimation for camera translation (albeit along all the 3 axes) and with aperture variation. In Bhavsar and Rajagopalan (2010), we consider depth estimation under general camera motion and parameter variation, but inpainting only static missing regions (e.g. due to sensor damage). In this paper (1) we extend the inpainting section by also considering the case of removing occluders present in the scene. We note that this is a more general problem than the one considered in Bhavsar and Rajagopalan (2010). (2) We also include a discussion on special cases of our general framework and bring out a couple of non-conventional and important special camera configurations arising out of our general framework. (3) We provide a brief analysis on the inpainting problem relating camera motion to the extent of the missing regions that can be inpainted. (4) Finally, we provide extensive results with quantitative evaluation for performance evaluation.

2 Coupling Motion, Blur and Depth

We now discuss the relationship among motion, blur and depth. Without loss of generality, we work in camera centered coordinates with the initial camera center coincident with the origin of the Cartesian coordinate system. The optical axis for the initial camera position coincides with z-axis, and the x- and y-axis are parallel to the image plane axes.

We denote the hypothetical focused (pin-hole equivalent) and noiseless ideal image with respect to the initial camera position as I. The i^{th} observation g_i is modeled as warped and blurred version of I i.e.,

$$g_i(n_1, n_2) = \sum_{l_1, l_2} h_i(n_1, n_2, \sigma_i, \theta_{1_i}, \theta_{2_i}) \cdot I(\theta_{1_i}, \theta_{2_i}) + \eta_i(n_1, n_2), \quad i = 1, 2, \ldots, N. \tag{3}$$

In the above equation, the geometrically transformed coordinates of the pixel at (l_1, l_2) in the hypothetical reference view I, are denoted by $(\theta_{1_i}, \theta_{2_i})$, while the blur kernel around the pixel which operates on $I(\theta_{1_i}, \theta_{2_i})$ is expressed by $h_i(n_1, n_2, \sigma_i, \theta_{1_i}, \theta_{2_i})$.

In the following discussion, we refer to Fig. 1 which shows perspective projection and blurring for two different lens position and settings. Note, from Fig. 1, that the blur kernel is centered on the point of projection of the central ray from the 3D point P on to the image plane. Also, the blur kernel for any point will always be symmetric around this point as long as the lens is symmetric and is parallel to the image plane (a common arrangement in most cameras). Thus, the geometry clearly shows that the transformation of a particular 3D point P can be described by its projection on the image plane followed by blurring of this projected image point. At the image level, this implies that in (3), the observations g_i are formed by depth-dependent warping of the pixels of f, followed by their space-variant blurring.

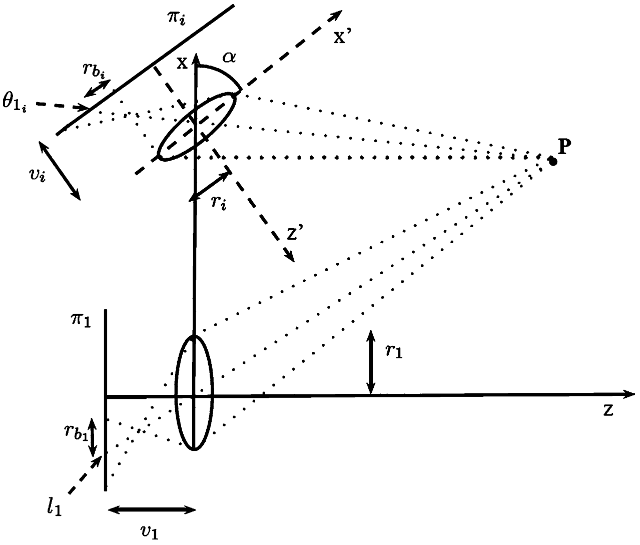

Fig. 1 Camera transformations

Hence, to describe the projection of a 3D particular point, we only require the principle ray that passes through the center of the lens, because the blur is always centered around the point of projection of this ray on the image plane. We shall now discuss the relationship of motion and blur with the 3D location of a point. In the following discussion, without loss of generality, we consider the reference camera to be placed at the world origin.

2.1 Motion-Depth Relationship

Let us denote the coordinates of a 3D point in space as (X, Y, Z), with respect to the reference camera position (or the world origin). The image pixel coordinates (l_1, l_2) in the first view corresponding to the perspective projection of this 3D point is expressed as

$$l_1 = \frac{v_1 X}{Z}, \qquad l_2 = \frac{v_1 Y}{Z} \tag{4}$$

where v_1 is the distance between the lens and the image plane for the first view.

Denoting v_i as the lens-to-image-plane distance in the i^{th} view, the camera translation along the x-, y- and z-axes by the elements of a 3×1 vector $[t_{x_i}\ t_{y_i}\ t_{z_i}]^T$, and the rotation parameters as the elements of a 3×3 matrix $R = [a_{p_1 p_2}]$, where $1 \leq p_1, p_2 \leq 3$, the 2D projection of the point (X, Y, Z) in the i^{th} view can be expressed as

$$\begin{aligned} \theta_{1_i} &= \frac{v_i a_{i_{11}} X + v_i a_{i_{12}} Y + v_i a_{i_{13}} Z + v_i t_{x_i}}{a_{i_{31}} X + a_{i_{32}} Y + a_{i_{33}} Z + v_2 t_{z_i}}, \\ \theta_{2_i} &= \frac{v_i a_{i_{21}} X + v_i a_{i_{22}} Y + v_i a_{i_{23}} Z + v_i t_{y_i}}{a_{i_{31}} X + a_{i_{32}} Y + a_{i_{33}} Z + v_2 t_{z_i}}. \end{aligned} \tag{5}$$

Eliminating X and Y, we can relate pixel coordinates in the reference and the i^{th} view in terms of the camera parameters and depth Z as

$$\theta_{1_i} = \frac{v_i a_{i_{11}} l_1 + v_i a_{i_{12}} l_2 + v_1 v_i a_{i_{13}} + v_1 v_i \frac{t_{x_i}}{Z}}{a_{i_{31}} l_1 + a_{i_{32}} l_2 + v_1 a_{i_{33}} + v_1 \frac{t_{z_i}}{Z}},$$
$$\theta_{2_i} = \frac{v_i a_{i_{21}} l_1 + v_i a_{i_{22}} l_2 + v_1 v_i a_{i_{23}} + v_1 v_i \frac{t_{y_i}}{Z}}{a_{i_{31}} l_1 + a_{i_{32}} l_2 + v_1 a_{i_{33}} + v_1 \frac{t_{z_i}}{Z}}. \tag{6}$$

2.2 Relating Blur with Depth

Having described the pixel motion, we now focus our attention on blurring. Our framework models the camera lens system with the thin lens model as shown in Fig. 1, which suggests a circularly symmetric point spread function (PSF) for the blur kernel. Without loss of generality, in this work, we use the Gaussian PSF model as it is a popular approximation owing to the effect of the central limit theorem for describing various optical aberrations (Pentland 1987; Rajagopalan and Chaudhuri 1999). A parametric blur model allows us to relate blur and depth analytically thus simplifying the calibration as compared to the case of arbitrary blur kernels. In general, the PSF need not have a parametric form. Our depth estimation/inpainting approach (elaborated in Sect. 3), works by computing the relative blur between images, i.e., for each point it computes the blur that, ideally, brings the i^{th} image close to the reference image. Such a computation of relative blur can be done for any known arbitrary relative blur kernel. Hence, our framework does not enforce any particular form or parametrization of the PSF. Some discussion on this issue is provided in Sect. 6.

The Gaussian blur kernel can be expressed as

$$h_i(n_1, n_2, \sigma_i, \theta_{1_i}, \theta_{2_i}) = \frac{1}{2\pi\sigma_i^2} \exp\left(-\frac{(n_1 - \theta_{1_i})^2 + (n_2 - \theta_{2_i})^2}{2\sigma_i^2}\right). \tag{7}$$

The depth dependent blur parameter $\sigma_i = \rho r_{b_i}$, where r_{bi} is the physical blur radius and ρ is the factor (camera constant) that converts physical units to pixels. We now derive the relationship of σ_i to depth Z of a 3D point under general camera parameter variation and motion.

Remark 1 The blur parameter σ_i for the i^{th} view, corresponding to a 3D point with coordinates (X, Y, Z) in the reference coordinate system, is related to the distance Z_i which is the distance between the lens plane in the i^{th} view and plane parallel to the lens plane and passing through that 3D point. This relationship can be stated as

$$\sigma_i = \rho r_i v_i \left(\frac{1}{u_i} - \frac{1}{Z_i}\right). \tag{8}$$

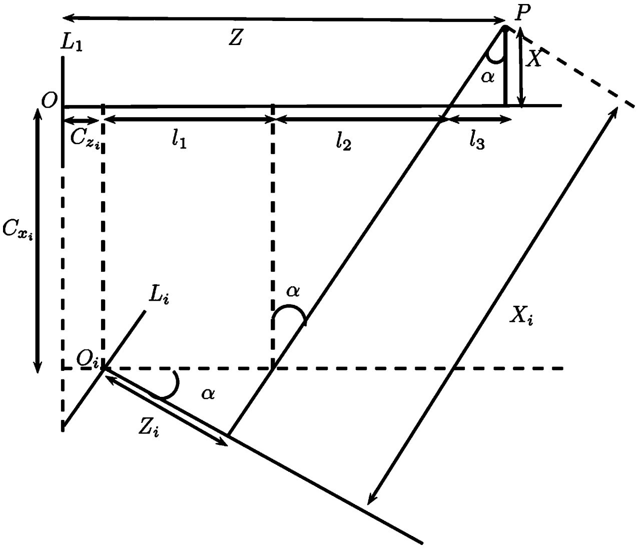

Fig. 2 A flatland representation of relative motion between rotated and translated coordinate systems

Here, r_i is the aperture radius in the i^{th} view, Z_i is the depth of point P with respect to the i^{th} camera, u_i signifies the focusing distance and v_i denotes the distance between the lens and the image plane.[1]

We now relate the depth Z_i with respect to the i^{th} view to the depth Z from the reference view. For, simplicity, we derive this in 2D and then generalize it to 3D. In Fig. 2, we show a flatland representation of the relative motion between two camera coordinate systems. The origin of the reference coordinate system, which coincides with the center of the lens L_1 is denoted by O. In the reference coordinate system, a 3D point P has coordinates (X, Z). Thus, the distance between the lens plane L_1 and the plane parallel to the lens plane L_1 passing through P is Z, which is nothing but the Z-coordinate of P. Since we considered the first view as the reference view (the choice being quite arbitrary), we have $Z_1 = Z$ $(i = 1)$ for the reference view.

The lens L_i in the i^{th} view $(i \neq 1)$ with center O_i is also shown in Fig. 2 which is rotated about the x-axis by an angle α_i with respect to L_1. The lens center O_i is also shifted relative to O, by C_{x_i} and C_{z_i} along the x- and z-axes, respectively. The depth of point P from lens L_i is denoted by Z_i. Since we wish to compute the depth Z in the reference view (center O), we must express Z_i in terms of Z. This relationship can be derived as follows. According to Fig. 2,

$$Z = C_{z_i} + l_1 + l_2 + L_3 \tag{9}$$

$$= C_{z_i} + \frac{Z_i}{\cos(\alpha)} + \frac{C_{x_i}\sin(\alpha)}{\cos(\alpha)} + \frac{X\sin(\alpha)}{\cos(\alpha)}, \tag{10}$$

$$Z_i = (Z - C_{z_i})\cos(\alpha) - (X + C_{x_i})\sin(\alpha). \tag{11}$$

[1] This remark, although well-known, is emphasized here because it is important in the context of defocus blur in the multiple-view scenario.

Similarly, one can express X_i, the x-coordinate of point P in the i^{th} view, in terms of X and Z as

$$X_i = \frac{Z_i \sin(\alpha)}{\cos(\alpha)} + \frac{C_{x_i} \cos(\alpha)}{\cos(\alpha)} + \frac{X}{\cos(\alpha)} \tag{12}$$

$$= (Z - C_{z_i}) \sin(\alpha) - (X + C_{x_i}) \cos(\alpha). \tag{13}$$

This can be written in matrix-vector form as

$$\underline{X}' = R_i(\underline{X} - \underline{C_i}) \tag{14}$$

where

$$R_i = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix}, \tag{15}$$

$$\underline{C_i} = \begin{bmatrix} -C_{x_i} & C_{z_i} \end{bmatrix}^T, \tag{16}$$

$$\underline{X} = \begin{bmatrix} X & Z \end{bmatrix}^T, \tag{17}$$

$$\underline{X_i} = \begin{bmatrix} X' & Z' \end{bmatrix}^T. \tag{18}$$

Here, R_i is the 2D rotation matrix and $\underline{C_i}$ is the vector signifying the shift in the center of projection. Conventionally, the vector $-R_i\underline{C_i}$ is denoted by $\underline{t_i}$, the translation of the camera. Thus, (14) can be written as

$$\underline{X_i} = R_i\underline{X} + \underline{t_i}. \tag{19}$$

The 3D counterpart of (19) involves a 3×3 R_i matrix and a 3D translation vector $\underline{t_i}$. The components $r_{i_{pq}}$ $p, q \in 1, 2, 3$ of the R_i matrix are functions of the rotation angles about the 3 axes, while the components of $\underline{t_i}$ denote the camera translations along these axes. Hence, in a 3D case, Z_i for a point P with reference coordinates (X, Y, Z) can be expressed as

$$Z_i = a_{i_{31}} X + a_{i_{32}} Y + a_{i_{33}} Z + t_{z_i}. \tag{20}$$

From (4), we can rewrite Z_i in terms of (l_1, l_2) as

$$Z_i = Z\left(\frac{a_{i_{31}} l_1}{v_1} + \frac{a_{i_{32}} l_2}{v_1} + a_{i_{33}}\right) + t_{z_i}. \tag{21}$$

From (8) and (21), we have the following.

Proposition 1 (Blur variation under general camera motion) *Under general camera motion, σ_i at a particular pixel (l_1, l_2) in the i^{th} view is related to the depth Z (in the reference coordinate system). This relationship of σ_i and Z in terms of known intrinsic and extrinsic camera parameters, can be expressed as* [2]

$$\sigma_i = \rho r_i v_i \left(\frac{1}{u_i} - \frac{1}{Z\left(\frac{a_{i_{31}} l_1}{v_1} + \frac{a_{i_{32}} l_2}{v_1} + a_{i_{33}}\right) + t_{z_i}}\right). \tag{22}$$

Note, from Fig. 1, that the blur kernel is centered on the point of projection of the central ray on to the image plane. Thus, according to the above model the transformed blur kernel is formed around the point $(\theta_{1i}(l_1), \theta_{2i}(l_2))$ in the i^{th} image. Hence, the position of the blur kernel is also warped in the i^{th} image.

2.3 Some Special Cases

Here, we provide a discussion on various special cases that emerge from the generalized imaging model described by (3), (6) and (22).

- The stereo case arises when the aperture $r_i = 0$ $(\forall i)$ in (22) but the motion parameters (especially, the translation components $t_{x_i}, t_{y_i}, t_{z_i}$) are not all zero in (6). Note that these will yield $\sigma_i = 0$ in (22), no matter what values the motion and the other intrinsic parameters are, since $r_i = 0$. Thus, due to r_i being 0, blur kernel h_i is an impulse located at the kernel center. Thus, the pure stereo case does not involve any blurring of the images and the only depth cue arises from pixel motion. Our framework allows $r_i \neq 0$ and thus the presence of varying blur across images. Also, the shift in the blur kernel is accounted for in (3), as the blur kernel operates at the shifted pixel location $(\theta_{1_i}, \theta_{2_i})$.
- The DFD scenario restricts the rotation matrix to be an identity matrix and the translation components to be all zero. However, at least one of the intrinsic parameters $(r_i, v_i$ or $u_i)$ in (22) is varied across images. Since there is no camera motion, only the blur acts as a cue for depth. Note that in this case the variation in v_i will also be responsible for co-ordinate scaling (see (6)). However, this scaling/magnification effect is usually neglected in DFD, by assuming the availability of (an expensive) telecentric lens (Watanabe and Nayar 1995) (on the image-side) while capturing the images. Our general model does not make any such assumption and allows for scaling of image coordinates. The shift in the kernel due to image scaling is inherently accounted for in (3).
- The SFF scenario enforces the intrinsic parameters to be constant (and non-zero) across images but allows for restricted camera motion (only the t_{z_i} translation component is active). The camera translation t_{z_i} will induce pixel shifts according to 6, but this is ignored in SFF approaches, again through the assumption of tele-centricity (Watanabe and Nayar 1995) (on the object-side) akin to the DFD scenario. If the effect of camera motion is ignored, then blur acts as the only cue. However, if telecentricity assumption is relaxed, the camera motion will play a role by inducing depth dependent pixel-shift in which case the SFF scenario turns out to be a special case of real-aperture axial stereo (Sahay and Rajagopalan 2010).

[2] This relationship is the generalization of the defocus blur equation to the multi-view scenario.

3 Depth Estimation Using Belief Propagation

In this section, we formulate the depth estimation problem in a MAP framework which we solve using belief propagation (BP) (Felzenszwalb and Huttenlocher 2004). Belief propagation has shown much promise in recent years for efficiently solving standard computer vision problems such as denoising and stereo. As compared to traditional combinatorial optimization problems such as simulated annealing or iterated conditional modes, the fast-BP algorithm is computationally more efficient and quick in convergence. Furthermore, although not considered in this paper, BP can be made highly parallel and is well-suited for implementation on GPUs.

The max-product BP computes the MAP estimates over a graph (Felzenszwalb and Huttenlocher 2004). For images, the graph is usually a grid-graph, with nodes being pixel locations. (For the following discussion on BP, we denote nodes by p, q and s, for conciseness.) The max-product rule works by passing messages $m^t_{pq}(f_q)$ at time t to a node q from its neighbouring node p of the graph as follows:

$$\begin{aligned} & m^t_{pq}(f_q) \\ & \quad = \min_{f_p}\left(D_p(f_p) + V(f_p, f_q) + \sum_{s\in\Omega(p)|q} m^{t-1}_{sp}(f_p)\right) \qquad (23) \end{aligned}$$

where $D_p(f_p)$ is the data cost at node p for accepting a label f_p, $V(f_p, f_q)$ is the prior cost between the neighbouring nodes p and q, and $s \in \Omega(p)|q$ denotes the set of nodes in the neighbourhood of p, not including q. The message vector m^t_{pq} is an L-dimensional vector, where L is the number of labels that each node can take. This message passing is iterated for each node until convergence. At convergence, the beliefs are computed as

$$b_q(f_q) = D_q(f_q) + \sum_{p\in N(q)} m_{pq}(f_q). \qquad (24)$$

The belief $b_q(f_q)$ at each node q is a L-dimensional vector. The MAP solution for the label at q is that f_q which maximizes $b_q(f_q)$.

In our framework, the data cost is formulated from the image generation process described in the previous section. The prior cost is chosen to constrain neighbouring nodes to favour similar labels. Moreover, since some portions in the reference image may not be visible in the i^{th} image, we modulate the data cost with a visibility term which is updated at each iteration. Furthermore, to improve the depth estimate at unreliable and occluded pixels, we use a cue from color-image segmentation and plane-fitting, inspired from recent works in stereo vision (Yang et al. 2009). We now explain our cost computation in more detail. BP does not have the notion of an initial estimate of the unknown variable. As we discuss later in the section, this necessitates an approximation to be made while handling defocus blur.

3.1 Data Cost

To define the data cost, we relate the image in the i^{th} view with that in the reference view. Without loss of generality, the first image is considered as the reference image ($i = 1$). For ease of explanation, we consider for now, that the reference image is modeled as a shifted and blurred version of the i^{th} image. (This will not hold good if the reference image is more blurred than the i^{th} image, a situation which we will discuss shortly.) Thus, the relationship between the reference image and the i^{th} image is given as

$$\begin{aligned} g_1(n_1, n_2) &= h_{ri}(\sigma_i, n_1, n_2) * g_i(n_1, n_2) \\ &= \sum_{l_1,l_2} h_{r_i}(\sigma_i, n_1 - \theta_{1_i}(l_1, l_2), n_2 - \theta_{2_i}(l_1, l_2)) \\ &\quad \cdot g_i(\theta_{1_i}(l_1, l_2), \theta_{2_i}(l_1, l_2)) \qquad (25) \end{aligned}$$

where h_{ri} signifies the relative Gaussian blur kernel corresponding to blur parameter (standard deviation) $\sqrt{\sigma_1^2 - \sigma_i^2}$. The symbol $*$ denotes convolution. The relative blur parameter $\sqrt{\sigma_1^2 - \sigma_i^2}$ can be related to Z by substituting (8) for σ_1 and σ_i. The data cost for a particular node in the i^{th} view is defined as

$$\begin{aligned} & E_{d_i}(n_1, n_2) \\ & \quad = |g_1(n_1, n_2) - h_{r_i}(\sigma_i, n_1, n_2) * g_i(n_1, n_2)|. \qquad (26) \end{aligned}$$

Due to general camera motion and variation in focusing distance, g_1 need not be more blurred than g_i at all pixels (the near-far focusing scenario). The data cost in (26) involves blurring g_i so as to yield an estimate of g_1. Hence, (26) will not be valid where g_i is more blurred than g_1 (as mentioned earlier). In such cases, only the magnitude of $\sigma_1^2 - \sigma_i^2$ is not sufficient for estimating depth, since two depth values on opposite sides of the $\sigma_1^2 - \sigma_i^2 = 0$ plane can yield equal magnitude of $\sigma_1^2 - \sigma_i^2$. To resolve this, we modify the data cost computation as follows. For a particular depth label, if $\sigma_1^2 - \sigma_i^2 \geq 0$, we use (26) to define the data cost at the node. If $\sigma_1^2 - \sigma_i^2 < 0$ we define the original data cost (viz. a counterpart of (26)), as

$$\begin{aligned} & E_{d_i}(n_1, n_2) \\ & \quad = |g_i(\theta_{1_i}, \theta_{2_i}) - h_{r_1}(\sigma_i, n_1, n_2) * g_1(n_1, n_2)|. \qquad (27) \end{aligned}$$

The convolution on the right hand side of (27) is defined as

$$\begin{aligned} & h_{r_1}(\sigma_i, n_1, n_2) * g_1(n_1, n_2) \\ & \quad = \sum_{l_1,l_2} h_{r_1}(\sigma_i, n_1 - l_1, n_2 - l_2) \cdot g_1(l_1, l_2). \qquad (28) \end{aligned}$$

In this case, since g_i is more defocused than g_1, we blur g_1 to yield an estimate of g_i.

At this point, we note that (26) and (27) involve a convolution of a shifted blur kernel and a shifted image. However, (3) does not involve a convolution, since it models the image generation with space-variant blur. Thus, the data costs defined in (26) and (27), assume local space-invariance which is an approximation to the actual image generation model of (3). Such an approximation is also known as the *equi-focal* approximation and is commonly used in DFD works (Favaro et al. 2008).

This approximation is required to make our data cost amenable to processing by the BP algorithm. The data cost in BP, for a particular label at a node, is defined as *the cost incurred by that node for accepting that particular label.* For applications such as de-noising or stereo disparity estimation, the data cost at a particular node involves only the label at that node and is independent of the labels of neighbouring nodes. However, for applications involving space-variant blur, this does not hold since the blur at the neighbouring nodes also influences the observation at a particular node. This requires the current depth estimates for the neighbouring nodes. However, BP does not entertain a notion of current label estimates and operates by simply finding the best label out of a set of labels that minimizes a cost. Defining the data cost at a node through a convolution (26), (i.e. using the equi-focal approximation), involves only the relative blur kernel values at the current node which in turn depends on the depth label at the current node. Thus the data cost is rendered independent of the labels of the neighbouring nodes, allowing it to be used within the BP framework.

3.2 Visibility

We incorporate the notion of visibility in the data term to handle occlusions. Here, we consider the case of stereo occlusions. The view occlusion case with its inpainting interpretation, warrants a separate discussion which is deferred to Sect. 4.

For handling stereo occlusions, we introduce a binary visibility function V_i. For a particular site (n_1, n_2) on the reference grid, $V_i(n_1, n_2)$ is 1 if the pixel at that site is visible in the i^{th} image and 0 if the pixel is occluded. The modulated data cost is given by

$$E_{d_i}(n_1, n_2) = V_i(n_1, n_2) \cdot |g_1(n_1, n_2) - h_{r_i}(\sigma_i, n_1, n_2) * g_i(n_1, n_2)|. \tag{29}$$

We begin by considering all pixels as visible. In each iteration, the visibility is computed by warping the current depth estimate in the i^{th} view. As observed in Drouin et al. (2005), Kang et al. (2001) for the stereo problem, computing visibility in each iteration independently, may not yield a convergent solution (Kang et al. 2001). Hence, we use geo-consistency definition (Drouin et al. 2005) to update visibility in a temporal manner as

$$V_i(n_1, n_2, t) = V_i^{\text{new}}(n_1, n_2, t) \cdot V_i(n_1, n_2, t-1) \tag{30}$$

where $V_i^{\text{new}}(n_1, n_2, t)$ is the visibility computed with the current disparity estimate and t denotes the iteration index.

The total data cost for a particular node considering all views is then computed as

$$E_d = \frac{1}{N_i} \sum_i E_{d_i} \tag{31}$$

where $i > 1$ and N_i is the total number of images (excluding the reference image) where the pixel is visible.

3.3 Prior Cost

The prior cost enforces a smooth solution that constrains the neighbouring nodes to have similar labels. (At a more fundamental level, a smoothness constraint on depth manifests from modeling the depth as a Markov random field having a joint Gibbs distribution (Li 1995).) Although, a smooth solution is preferred, we also wish to avoid over-smoothing of prominent discontinuities in the solution. Thus, the penalty for neighbouring labels being different cannot be arbitrarily high. We define the smoothness prior as a truncated absolute function which is given by

$$E_p(n_1, n_2, m_1, m_2) = \min(|Z(n_1, n_2) - Z(m_1, m_2)|, T) \tag{32}$$

where (n_1, n_2) and (m_1, m_2) are neighbouring nodes in a first-order neighbourhood and T is the truncation threshold to saturate the prior cost beyond a certain difference in the depth labels.

3.4 Incorporating Segmentation Cue

Modeling space-variant blurring as local convolution is generally a good approximation except at depth discontinuities, especially if the blurring differences are large across discontinuities. This factor, in addition to image noise and errors in computing occlusions, can make the depth estimation error prone. To mitigate such errors, we incorporate segmentation cue in our estimation (Yang et al. 2009), as explained below.

Prior to starting the estimation, we color-segment the reference image using the mean-shift algorithm (Dorin and Meer 2002). We also compute a reliability map to classify the pixels as reliable or not. To form this reliability map, we compute the top two labels which provide the two least data costs. Calling these costs as C_1 and C_2, we compute a confidence measure

$$C = \frac{|C_1 - C_2|}{C_2}. \tag{33}$$

This measure is similar to that defined in Yang et al. (2009). However, we note that in Yang et al. (2009) the costs C_1 and C_2 are computed using a correlation volume. In our case, the terms C_1 and C_2 are computed using the data cost itself, which also accounts for the blurring effect while computing the confidence. If the confidence measure is above a particular threshold c_f then we define that pixel to be a reliable pixel. After this initial processing, the actual estimation process follows.

The first BP iteration is run without using the segmentation cue. We then compute a plane-fitted depth map that uses the current estimate, the segmented image and the reliability map. The plane computation for each segment is carried out via the robust RANSAC (Fischler and Bolles 1981) approach, using only reliable pixels in that segment. The plane-fitted depth map is computed as follows. If the fraction of reliable pixels r_f in a segment is above a threshold, then the reliable pixels are assigned their own depth values and only the unreliable pixels are assigned the plane-fitted depth values. If this is not so, then all the pixels in the segment are assigned the plane-fitted depth values. If the segment itself is very small ($< s_f$ pixels), then all pixels are assigned the median of the current depth labels for that segment.

Once the plane-fitted depth map is computed as explained above, we feed it back into the iteration process to regularize the data term. Thus, the new data term is

$$\begin{aligned} E_{d_s}(n_1, n_2) &= E_d(n_1, n_2) \\ &\quad + w(n_1, n_2) \cdot |Z(n_1, n_2) - Z_p(n_1, n_2)| \end{aligned} \tag{34}$$

where Z_p denotes the plane-fitted depth map and the regularization weight w is binary (0 if the pixel is a reliable pixel and 1 if it is not). The second term in (34) regularizes the unreliable depth estimate such that these estimates do not deviate from the plane-fitted depth map. We use this data term in subsequent iterations after the first.

4 Handling View Occlusions: Depth and Image Inpainting

We are now ready to discuss the view occlusion problem. Many natural scenes contain unwanted foreground objects that occlude the objects of interest in the background. Often these objects are such that they would occlude, at least partially, a background object of interest no matter from which view one captures the scene. Information gathering can also be hampered due to defects in camera sensors and lenses. These include sensor contamination by dust and humidity while changing lenses (Zhou and Lin 2007), sensor damage from over-exposure to sunlight etc. Similarly, lens damage due to shocks as well as climatic effects, and occlusions due to lens depositions/attachments can also lead to image artifacts (Gu et al. 2009; Sahay and Rajagopalan 2009a). It is not surprising that numerous photography sites/blogs exist on the web that discuss these issues, thus highlighting the practical significance of this problem.

The above mentioned issues can be classified into two scenarios, both of which we address. (1) The case where it is desired to remove, from the reference depth map and the image, a user-defined object which is a part of the scene and occludes the background objects of interest. (2) The case where some portion of the image data is missing due to damage to sensor/lens and one desires to retrieve the depth and image, from the reference view, in the damaged areas.

Our approach to handle such view occlusions is essentially built upon the depth estimation framework described in Sect. 3. We realize that camera motion, which is an integral part of our framework, can be exploited to *discover*, in other views, the pixels occluded/missing in the reference view. Note that the view occlusion case is in fact quite complementary to that of stereo occlusions. Unlike in the latter case, where the points in the reference view are rendered invisible in other views due to motion parallax, in the view occlusion scenario, the points invisible in the reference image are rendered visible in other views, also due to motion parallax.

An example to demonstrate this effect is shown in Fig. 3 where the wires in the foreground serve as occluders, which we wish to remove. We observe that the portions which are occluded in the reference (left) image are uncovered in the right image due to the inherent parallax between the wires and the other background objects. Some such prominent portions which can be explicitly pointed out include the bars in the Pisa tower, neck of the bunny, and the dots on the yellow tree in Fig. 3. Thus, if such pixels can be found in more than one view then the correspondence (depth) and intensities can be found even for the pixels which are not visible in the reference view.

Given the images where the foreground/damaged regions to be removed are marked by the user, our approach com-

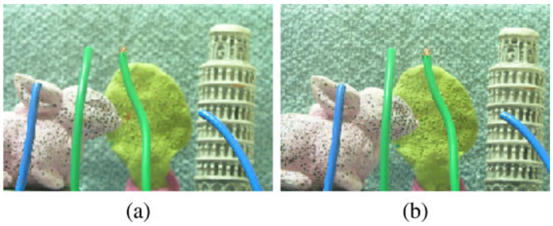

Fig. 3 (**a**, **b**) Two observations showing the *uncovering* of regions in the second image which are occluded in first image

putes the depth map along with the background depth labels estimated in the regions marked as occluded. Note that as described in Sect. 3.1, the correspondence search for occlusion removal also considers the defocus blur. Given the estimated depth map, we fill in the missing region in the (reference) color image. Importantly, since the observations are acquired using a real-aperture camera, the intensity assignment for the occluded pixels should also satisfy the defocus extent from that view to maintain visual coherence.

Removing occlusions caused by objects/device damage is essentially the same as filling in background information in missing regions in the images. In this sense, the view occlusion problem can also be interpreted as one of inpainting. In our approach, the information to be inpainted is actually available in the images due to the motion cue, which is exploited to *inpaint* both depth and image. Moreover, since filling in color information in missing regions in the image also accounts for the defocus blur, we refer to it as *defocus inpainting*. Thus, unlike traditional inpainting works, our approach is *cue-based*, which considers the motion and defocus cues for inpainting the occluded regions.

Intra-Scene Occluder Here, the task is to remove from an image, an object which is part of the scene. Typically, this is a foreground object occluding background objects in the scene. As the occluder is just another object in the scene, it also undergoes an apparent motion in the image like other background objects do. However, the motion of the occluder is different from that of the background objects. Hence, pixels occluded in the reference image may be observed (uncovered) in other images. Thus, even if correspondences with the reference image cannot be found, they can still be established across other images.

As mentioned earlier, the locations of the occluded pixels to be inpainted is usually assumed to be known and are a priori marked by the user. Although our approach involves multiple images, the user interaction is minimized by marking the occluder pixels to be inpainted *only in the reference image*. The occluder pixel locations in the images other than the reference are marked depending on the knowledge of those in the reference image. However, the motion of the occluder pixels across images depends on their depth values. Hence, we compute the depth map of the complete scene including the occluder using our proposed depth estimation approach. Given this depth map and the reference image where the occluder pixels are marked, we compute and mark the location of the occluder pixels in other images.

Missing Pixels Due to Sensor/Lens Damage While considering sensor/lens damage, the situation is somewhat simpler. Given that a single damaged camera is used to capture the images, all observations will be equally affected due to the sensor/lens damage; the locations of the missing pixels do not vary in the image coordinate system i.e., the motion of the missing pixels is zero. In case where multiple differently damaged cameras are used, the location of the missing regions will be different in each image. However, in both cases, the motion of the pixels to be inpainted will be independent of depth. Thus, in the case of sensor/lens damage, prior computation of depth of the complete-scene, as in the intra-scene occluder case, is not required.

Due to camera motion, the locations of pixels corresponding to scene points will change independent of the damaged pixels. Thus, the above argument of missing pixels in the reference image being visible in the other images, still holds and provides the cue for inpainting.

We now formally describe our approach for depth and image inpainting where we have the reference image with occluded/missing regions marked. Given the observations with the missing regions marked, the following methodology for depth and image inpainting is common for both the cases of intra-scene occluder as well as device damage.

4.1 Depth Inpainting

We denote the set of missing pixels as M. We begin by arranging the images in an (arbitrary) order $(g_1, g_2, \dots, g_N)$ with g_1 being the reference image. For pixels $\notin M$ in all images, we estimate the depth as described in Sect. 3. Thus, if there are no missing regions and no pixels belong to M, we effectively carry out depth estimation as in Sect. 3.

If a pixel $g_1(l_1, l_2) \notin M$ and $g_j(\theta_{1_i}, \theta_{2_i}) \in M$ for some $i > 1$ then the data cost between the reference view and the i^{th} view is not computed. In case, $g_j(\theta_{1_i}, \theta_{2_i}) \in M \ \forall i \neq 1$, then the pixel is left unlabeled.

If $g_1(l_1, l_2) \in M$, we look for observations at $(\theta_{1_i}, \theta_{2_i})$ and $(\theta_{1_j}, \theta_{2_j})$ for a depth label. If the observations $g_i(\theta_{1_i}, \theta_{2_i}) \notin M$ and $g_j(\theta_{1_j}, \theta_{2_j}) \notin M$, the matching cost between them is defined as

$$E_{d_i}(n_1, n_2) = V_{ij}(n_1, n_2) \cdot |g_i(\theta_{1_i}, \theta_{2_i}) - h_{r_{ij}}(\sigma_{ij}, n_1, n_2) * g_j(n_1, n_2)| \tag{35}$$

where $1 < i < j$, and

$$h_{r_{ij}}(\sigma_i, n_1, n_2) * g_j(n_1, n_2) = \sum_{l_1, l_2} h_{r_{ij}}(\sigma_i, n_1 - \theta_{1_j}, n_2 - \theta_{2_j}) \cdot g_j(\theta_{1_j}, \theta_{2_j}) \tag{36}$$

and

$$V_{ij}(n_1, n_2) = V_i(n_1, n_2) V_j(n_1, n_2). \tag{37}$$

Note that the variables in the above equation are with respect to the i^{th} and the j^{th} as opposed to the reference and the i^{th} image used in defining the data cost in Sect. 3.1. Here, $h_{r_{ij}}$

denotes the blur kernel corresponding to the blur parameter $\sqrt{\sigma_i^2 - \sigma_j^2}$. The *compound* visibility V_{ij} signifies that the data cost is not computed if a pixel is not observed in either the i^{th} or the j^{th} view. We also update this compound visibility similar to (30).

The corresponding data cost for a pixel and depth label, considering all views, is then computed by summing the matching costs as $E_d = \frac{1}{N_{ij}} \sum_i E_{d_i}$, where N_{ij} are the number of pairs of images g_i and g_j such that $g_i(\theta_{1_i}, \theta_{2_i}) \notin M$ and $g_j(\theta_{1_j}, \theta_{2_j}) \notin M$ and $V_{ij}(l_1, l_2) \neq 0$. Thus, the cost for a pixel missing in the reference image is computed by using those images in which the corresponding pixels are visible.

Equation (35) implicitly assumes that $g_j(\theta_{1_j}, \theta_{2_j})$ is blurred and compared with $g_i(\theta_{1_i}, \theta_{2_i})$. However, as in Sect. 3.1, this is assumed only for ease of explanation. The vice-versa case (near-far focusing) can be handled comfortably in a manner similar to that discussed in Sect. 3.1.

The above process can yield pixels which are not labeled (for which no correspondences are found). Moreover, as discussed in Sect. 3.4, some pixels can also be labeled incorrectly. We invoke the segmentation cue similar to that explained in Sect. 3.4 to mitigate such errors. Note, however, that color segmentation of damaged observations will yield segments corresponding to missing regions. For brevity, we denote a set of such segments by S_m. Each such segment will span largely different depth layers, thus disobeying the very premise for the use of segmentation, and cannot be used for computing the plane-fitted depth.

To address this issue, we assign the pixels in S_m to the *closest* segment $\notin S_m$. This essentially extends the segments neighbouring to those in S_m by including the pixels belonging to S_m. The *closeness* is determined by searching in eight directions around the pixel. Thus, after this operation every pixel that belonged to S_m will be assigned a new segment-label of that segment ($\notin S_m$) which is closest to that pixel. The intuitive idea is that most of such pixels which belong to the segment S_m corresponding to the missing region, actually would have been a part of the segments to which they are assigned after this operation, if the missing regions, which gave rise to S_m, had not been present. Thus, this operation computes the natural segmentation labeling which is necessary for using the segmentation cue.

The plane-fitted depth map is then computed using the reliable pixels in these extended segments. This plane-fitted depth map is fed back into the estimation process in the next iteration where the unlabeled pixels are now labeled because of the regularizer depending on the plane-fitted depth map. Further iterations improve the estimates.

4.2 Image Inpainting

Given the estimated depth map, we now wish to estimate the color labels for the missing pixels. We minimize a data cost using the BP algorithm, which compares the intensities of $g_i(\theta_{1_i}, \theta_{2_i})$, $i > 1$ with an intensity label, if $g_i(\theta_{1_i}, \theta_{2_i}) \notin M$. This data cost is defined as

$$E_d(n_1, n_2) = V_i(n_1, n_2) \cdot |L - h_{r_i}^p(\sigma_i, n_1, n_2) * g_i(n_1, n_2)| \tag{38}$$

where L is an intensity label, and the convolution is defined as in (25). We note that the image inpainting accounts for the blurring process. The kernel superscript p denotes that $h_{r_i}^p$ carries out a partial sum for only those pixels in its support which $\notin M$.

Here, a minor limitation is that g_1 should be more defocused than g_i for at least some $i > 1$ for the pixels to be inpainted. The above data cost is computed only for those g_is satisfying this condition, which however, is easily met, given sufficient images.

Lastly, there may be missing pixels in g_1 for whom it is possible that $g_i(\theta_{1_i}, \theta_{2_i}) \in M \ \forall i$. Such pixels are left unlabeled. The extent of these unlabeled regions depends on the original extent of the missing region and pixel motion. In our experiments, for most cases, the pixel motion, when all the images are considered, is sufficient to leave no missing region unlabeled. The maximum extent of such unlabeled regions, if they exist at all, is up to 2–3 pixels. Such small unlabeled regions can be filled by any inpainting algorithm; for instance, we use exemplar-based inpainting (Criminisi et al. 2003).

4.3 Comments on the Inpainting Ability with Respect to Extent of the Missing Regions

Since our inpainting approach is dependent primarily on the motion cue, the motion for a particular pixel should be such that it crosses the missing regions in one or more of the other images.

In the case of camera damage, where the missing regions are static, we can see that

$$\begin{aligned} &w_1 < |l_1 - \theta_{1_i}| \quad \text{or} \\ &w_2 < |l_2 - \theta_{2_i}| \end{aligned} \tag{39}$$

where w_1 and w_2 are the extent of the missing regions, starting from (l_1, l_2) in the directions given by the signs of $(\theta_{1_i} - l_1)$ and $(\theta_{2_i} - l_2)$ along the x and y image axes, respectively. Thus, for a missing pixel at (l_1, l_2) in the reference image, its corresponding warped pixel at $(\theta_{1_i}, \theta_{2_i})$ in the i^{th} image should satisfy the above relation to go beyond the missing regions. In fact, for depth estimation, the above condition must be satisfied for at least two images, while for image inpainting it will suffice if even any one image satisfies (39).

For the case of intra-scene occlusion, the foreground to be inpainted also changes its location across the observations depending on its own depth. However, given the depth map containing the foreground occluder (which we compute as a first step), we can know locations of the occluder pixels in the i^{th} image. Thus, the ideal condition for the motion to serve as an inpainting cue can be written as

$$\begin{aligned}
&|l_1 - \theta^o_{1i}(l_1)| + w_{1_p} < |l_1 - \theta^b_{1i}(l_1)| \quad \text{or} \\
&|l_1 - \theta^o_{1i}(l_1)| - w_{1_n} > |l_1 - \theta^b_{1i}(l_1)| \quad \text{or} \\
&|l_2 - \theta^o_{2i}(l_2)| + w_{2_p} < |l_2 - \theta^b_{2i}(l_2)| \quad \text{or} \\
&|l_2 - \theta^o_{2i}(l_2)| - w_{2_n} > |l_2 - \theta^b_{2i}(l_2)|
\end{aligned} \tag{40}$$

where w_{1_p} and w_{2_p} are the extent of the missing regions in the i^{th} image, starting from $(\theta^o_{1_i}(l_1), \theta^o_{2_i}(l_2))$ in the directions given by the signs of $(\theta^o_{1_i}(l_1) - l_1)$ and $(\theta^o_{2_i}(l_2) - l_2)$ along the x and y image axes, respectively. Similarly, w_{1_n} and w_{2_n} are the extent of the missing regions, in the directions given by the signs of $(l_1 - \theta^o_{1_i}(l_1))$ and $(l_2 - \theta^o_{2_i}(l_2))$, respectively. The superscripts o and b denote that the warped pixels are those for the occluder and the background. If the background pixels hidden in the reference image, satisfy the above condition, then they will undergo enough shift to come out of the missing region and will be uncovered in the i^{th} image.

We now provide some insights based on the above analysis, which can serve as important guidelines for inpainting under general camera motion.

- The warped pixel coordinates θ_{1_i} and θ_{2_i} depend on the camera motion, the distance between the image plane and lens, and the depth of the scene. For the sensor/lens damage scenario, there is no constraint on the type of camera motion (rotation and or translation). The missing pixels are static (have zero motion) with respect to the camera coordinate system while the scene pixels will almost always have a different (non-zero) motion no matter how the camera moves. The only constraint is that the amount of camera motion should be sufficient enough so as to potentially satisfy the above conditions.
- For the occlusion removal case, the inpainting essentially depends on the *parallax* between the foreground and the background. Thus, there must be a translational component in the camera motion. A purely rotating camera will make θ_{1_i} and θ_{2_i} independent of the foreground and background depths, thus effectively providing no relative motion between foreground and background.
- In the restrictive case of an axial translation (Sahay and Rajagopalan 2010), the motion-cue for inpainting is effectively absent near the image center due to small pixel motion. Hence, the above bounds are not satisfied. On the other hand, a general motion scenario allows the motion to act as an inpainting cue without any restrictions on the image positions.
- In general, the bounds derived in this section require knowledge of scene depth to compute θ_{1_i} and θ_{2_i}, which is unknown in the first place. However, they can still be used to derive worst-case bounds for maximum depths, bounds for average depths etc., given the knowledge of the depth ranges that we operate upon. Thus, one can indeed use these bounds to deduce the required motion so that the captured observations do contain sufficient information for inpainting.

5 Experimental Results

We provide validation on several synthetic and real images. The synthetic experiments were carried out on the Middlebury stereo database (Scharstein and Szeliski 2002) by creating warped and blurred observations from a focused image and depth map, using realistic camera parameters. For the real results, the observations were captured in our laboratory with Olympus C-5050 camera. The images were captured with the camera either in a normal mode or in a super-macro mode (in which the camera can focus on very short distances). The focal length of the camera was of the order of 1 to 2 cm. The distance range in the scene was 20 to 50 cm in the normal mode, and 3 to 15 cm in the super-macro mode, within which we varied the focusing distance. The f-number is varied between F/8–F/4. The camera translation and rotation was of the order of 5–20 mm and 5–15 degrees, respectively, with respect to the reference image. In all experiments, threshold T, in the prior cost, is chosen as half of the maximum depth label. The depth quantization is 0.5.

The intrinsic parameters of the Olympus C-5050 camera which can be varied are the aperture radius r, the focusing distance u and focal length f (which also controls the zoom). For this camera, the values of r and u are available while operating the camera whereas the value of f is available from the *exif* data stored in the images. For moving the camera, we used a calibrated translational and rotational table.

In Sect. 5.1, we first show the results for the generalized depth estimation described in Sect. 3 to validate our primary claims about a general framework. In addition to various general cases of intrinsic and extrinsic parameter variation, we also provide results for some non-conventional scenarios. Results for the view occlusion problem which involve depth inpainting as well as the defocus image inpainting are provided in Sect. 5.2.

Our experiments involve various arbitrary combinations of intrinsic and extrinsic parameters. In each of the Sects. 5.1 and 5.2, we typically begin with simpler cases involving variations in two parameters viz. camera translation along an axis and aperture variation, and then move on to more involved scenarios involving general camera motion, aperture

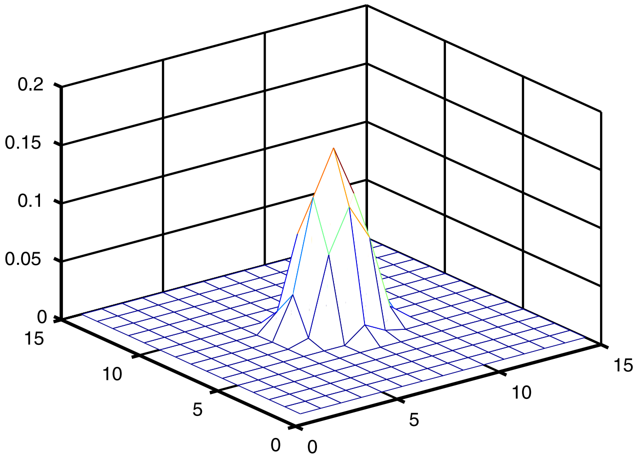

Fig. 4 A blur kernel estimated at a fronto-parallel region in n real image using two images with different aperture settings

variations and focusing distance. The idea is to show that our method performs consistently well regardless of the nature of variations in intrinsic and extrinsic parameters.

We provide quantitative evaluation for our synthetic experiments by computing the mean absolute error (MAE) with ground-truth. To evaluate the performance of our approach for real scenes we also provide quantitative evaluation for some of the experiments. The real experiments which are evaluated are chosen based on ease of ground-truth measurements in the scene for at least 4–5 different depths. Otherwise, these experiments and scenes have no special attributes as compared to others.

Before getting into the depth estimation and inpainting experiments, we show an example of a real relative blur PSF estimated for the Olympus C5050Z camera which we used in our real experiments. The PSF was estimated at a fronto-parallel region using a pair of real images with no relative motion (Fig. 4). We used the method of Paramanand and Rajagopalan (2010), which does not assume any additional constraints. We see that the circular symmetry suggested by the thin lens model is indeed satisfied. Moreover, we can observe a decay in the magnitude of the kernel which peaks at the center. The *real* blur kernel can, in fact, be well-approximated by a Gaussian. The choice of a parametric Gaussian model for the blur kernel is apt, not just due to the simplicity it offers in relating blur with absolute depth or its physical justification via the central limit theorem, but also due to the resemblance it bears to a real blur kernel.

5.1 Results for Depth Estimation

Synthetic Experiment (t_x, R_x, r) In Fig. 5 we show results for synthetic experiments on the *teddy* scene and the *plastic* scene. These experiments involve camera translation, rotation and aperture variation. For each scene, one of the images and the ground-truth disparity map is acquired from the Middlebury stereo database (Scharstein and Szeliski 2002). The ground-truth depth maps are computed from the disparity maps. Using one view, the ground-truth depth map, and realistic camera parameters (of the order of few centimeters), we create another view which is warped and differently blurred as compared to the first view. The translation, rotation and the camera parameters were chosen such that the maximum pixel motion was ≈ 30 pixels. The maximum relative blur parameter $\sigma_r \approx 2$. The observations for the *teddy* and *plastic* scenes are shown in Figs. 5(a, c) and Figs. 5(b, d), respectively. It is important to note that the image generation followed the space-variant model of (3), whereas the estimation is carried out using the convolution approximation discussed in Sect. 3.1. The estimated depth maps for the two scenes are shown in Figs. 5(g, h). Observe that the depth estimates are very close to the ground-truth (5(e, f)) and also show good discontinuity localization. The

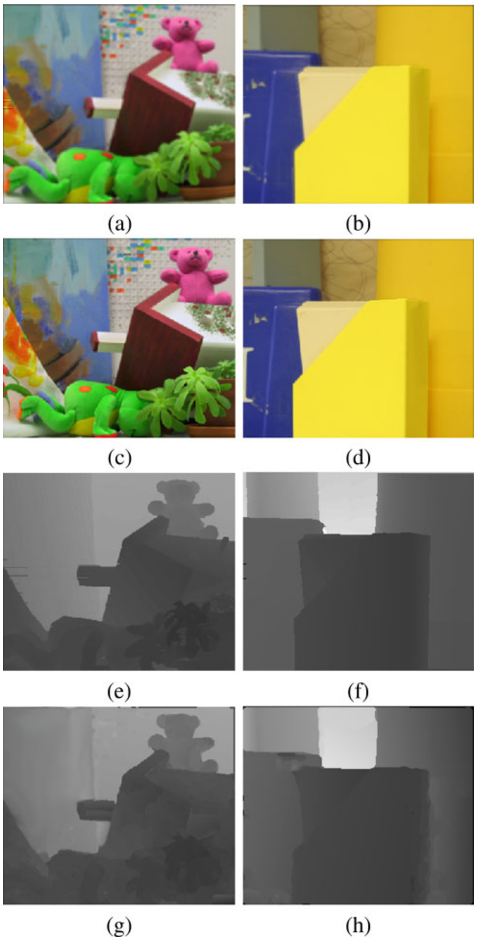

Fig. 5 Synthetic experiments: (**a**, **c**) and (**b**, **d**) Binocular image pairs with translation, rotation, aperture variation for *teddy* and *plastic* scenes, respectively. (**e**, **f**) Ground-truth depth maps, and (**g**, **h**) their respective estimated depth maps. MAE in depth estimation for the *teddy* and *plastic* scenes is 0.34 cm and 0.46 cm, respectively. The depth range for the scenes are 5–16 cm and 6–20 cm, respectively

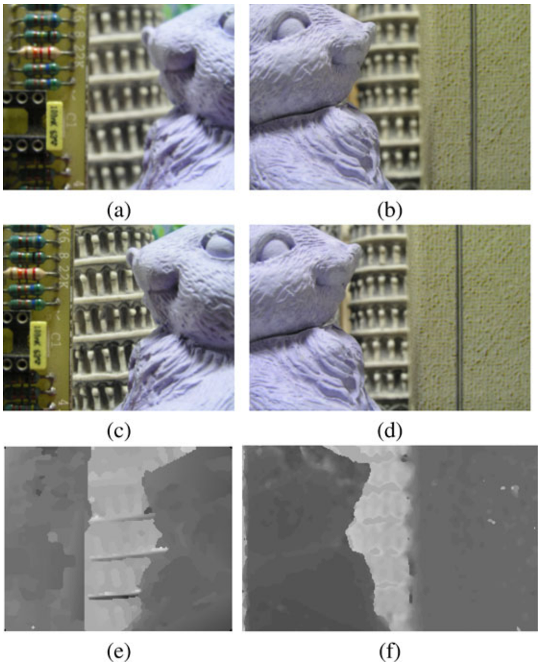

Fig. 6 Real experiments: (**a**, **c**) and (**b**, **d**) Translated image pairs with aperture variation for two different scenes. (**e**, **f**) Estimated depth maps

complicated shapes in the *teddy* depth map are quite well-preserved.

Binocular stereo real experiments (t_x, r) In Fig. 6 we demonstrate results on real data for two real scenes. We show both the results in a single figure since both examples involve binocular images (two observations shown in Fig. 6(a, c) and Fig. 6(b, d)) with t_x translation and variation in the aperture. In these experiments, we used the camera in the so called super-macro mode, since the scene depth range was less than 15 cm. The respective depth maps from the reference views of Figs. 6(a) and 6(b) are shown in Figs. 6(e) and 6(f). In addition to shape and discontinuity preservation, we note that the output is quite good even on somewhat low textured regions (e.g. the cardboard in the scene in the bottom row). Moreover, in this experiment, since the objects are kept quite close to the camera, the algorithm is also able to capture relatively fine variations (e.g. on the Pisa tower-model in both the scenes and on the circuit board in the example in the left column).

Multi-view Stereo Real Experiment (t_x, r) Next, we show an experiment using multiple images captured with a translating camera, varying in aperture. This real example involves a scene where all the foreground objects are low-textured wooden statues/models. (Figs. 7(a–c)). The camera is focused on the Ashoka pillar model (the leftmost model in the images). Note especially, the heavy blurring over the Ganesha idol and the background region. The estimated depth map for the reference view (Fig. 7(a)), is shown in Fig. 7(d). The algorithm is able to capture the depth variations with good shape localization. For instance, note the curves and the gaps in the Ashoka pillar model and the subtle sawtooth patterns in the hemisphere around the Buddha's head. In the Ganesha idol, the fact that the belly, trunk and the shade (over the head) are in front of the legs and the ears, manifests in the depth map too. A novel-view rendering of this scene is shown in Fig. 7(e). We also measured the actual distances (d_m) from the camera for some regions on the objects. We find that our estimated distances (d_e) (averaged for around 10 points in these regions) agree well with the measured distances. These values are provided in Table 2.

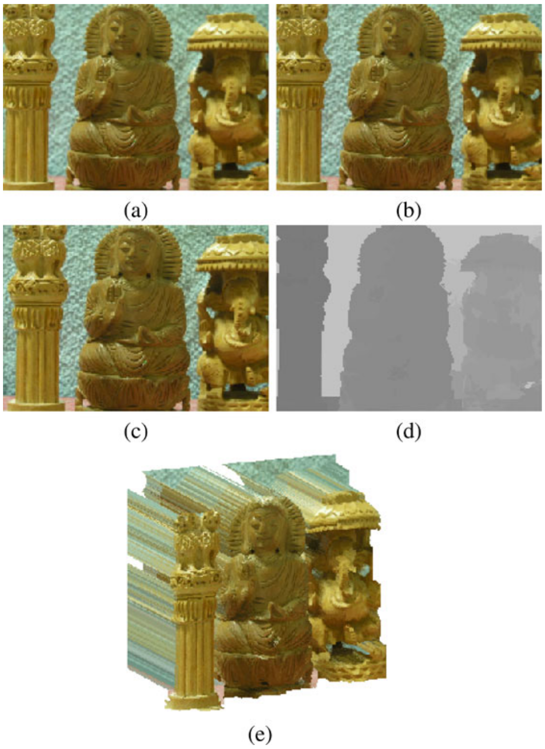

Fig. 7 Real experiment: (**a**, **b**, **c**) Translated observations with aperture variation. (**d**) Estimated depth map. (**e**) Novel-view rendering of the scene

Table 2 Comparison of the estimated and ground-truth distances for the scene shown in Fig. 7

Objects/regions	d_m (cm)	d_e (cm)
Ashoka pillar model	24	24.6
Buddha idol (central region)	30	27.6
Ganesha idol (central region)	32	31.15
Background (top region)	40	38.6

Multi-view Stereo Real Experiment (t_x, t_z, r) In Fig. 8 we provide a result on real data involving multiple observations with translation and variation in the aperture. Figures 8(a–c) show three out of the four observations that involve transla-

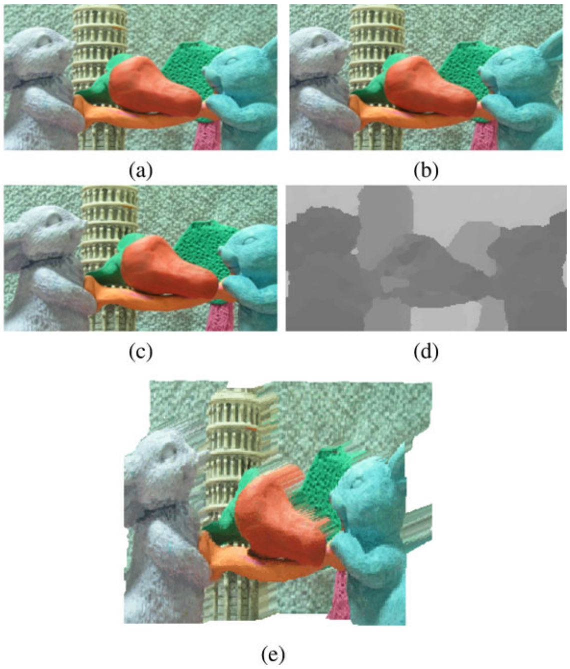

Fig. 8 Real experiment: (**a**, **b**, **c**) Observations with t_x and t_z translation and aperture variation. (**d**) Estimated depth map. (**e**) Texture-mapped novel-view rendering of the scene

tion in the t_x and t_z direction. Note that the image in Fig. 8(c) appears to be somewhat magnified than the other observations because of the t_z (axial) translational component towards the scene while acquiring this observation. The camera was focused on the Pisa tower model. The depth map from the reference view of Fig. 8(a) is shown in Fig. 8(d). The result re-assures that the approach captures all the important depth variations and achieves good discontinuity localization. The scene-objects in this experiment were also kept at similar distances as in the previous one. We can see that the gray-level depth representation which uses the same scale-factor as the previous experiment shows similar gray-levels, thus, indicating a correct estimation. A rendered novel view for this scene is shown in Fig. 8(e) by texture-mapping the reference observation onto the depth map.

Multi-view Stereo Real Experiment (t_x, R_x, r) In Figs. 9(a–c), we show three out of five observations involving translation, rotation and aperture variation. In this experiment, the camera was focused on the blue background. One can clearly make out the large relative motion and blur variation over the regions in the front. Our depth map output for the view in Fig. 9(a) is shown in Fig. 9(d). In this experiment too, the depth ranges are similar to those in the evaluated experiment of Fig. 7 and the result clearly shows the depth estimation fidelity. We can also see that the discontinuities are consistently preserved. Figure 9(e) shows a novel-view rendering of the scene, which brings out the relative depth differences between the objects.

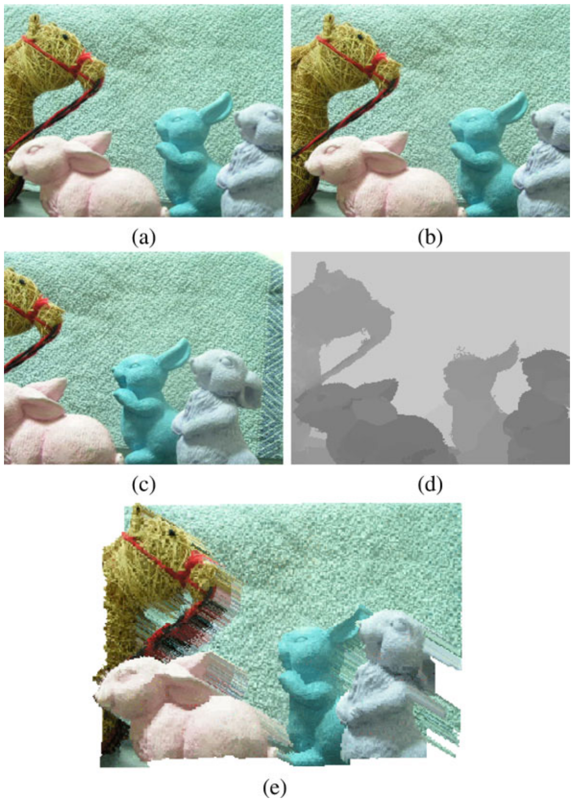

Fig. 9 Real experiment: (**a**, **b**, **c**) Observations with translation, rotation and aperture variation. (**d**) Estimated depth map. (**e**) A novel-view

5.1.1 Depth Estimation Results for Non-conventional Scenarios

In addition to the results for the various cases shown above, we now discuss two unconventional scenarios, arising in our general framework. In obtaining the results for these situations, we did not enforce the segmentation cue, since we wish to highlight the stand-alone ability of the inherent cues for depth estimation under these special camera configurations.

Depth Estimation with Pure Camera Rotation The first of these is the case of pure rotational camera motion. The images captured under pure camera rotation for a pin-hole camera model are geometrically related through a 8-parameter homography. For this special case, the pixel motion does not depend on the depth of the corresponding 3D points. Pure rotational motion provides no depth cue for the pin-hole camera (Hartley and Zisserman 2003). However, if the camera has a non-zero aperture, then all is not lost. Equation (22) for pure camera rotation yields

$$\sigma_i = \rho r_i v_i \left(\frac{1}{f} - \frac{1}{v_i} - \frac{1}{Z(\frac{a_{i_{31}} l_1}{v_1} + \frac{a_{i_{32}} l_2}{v_1} + a_{i_{33}})}\right). \qquad (41)$$

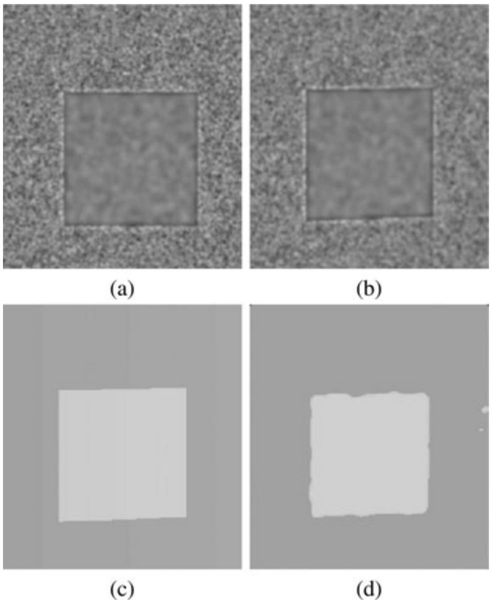

Fig. 10 Synthetic experiment: (**a**, **b**) Observations acquired from a purely rotating real aperture camera. (**c**) Ground-truth depth map. (**d**) Estimated depth map. MAE is 0.22 cm where the depth range is 8 to 10 cm)

Clearly, even though the camera parameters are kept constant, one can note that the blur parameter σ_i for the i^{th} view for the pixel at coordinates (l_1, l_2) in the reference view, is different than σ_1, the blur parameter in the reference view. Not only is the relative blur parameter $\sigma_r = \sqrt{\sigma_i^2 - \sigma_1^2}$ nonzero, but it is also a function of the depth Z. Hence, for a real-aperture camera, even pure camera rotation provides a depth cue through the relative blur induced by camera rotation.

Figure 10 shows the results for a synthetic experiment for depth estimation under pure camera rotation. For simplicity, the ground-truth depth was chosen to be a two-layer cake (Fig. 10(c)). A random-dot image was rotated and blurred to create the observations (Figs. 10(a, b)), with the reference image being the latter. The rotation was 2° while the depth varied from 8 to 10 cm. The camera parameters were realistically chosen so that the order of r_i was a few millimeters and that of u_i and v_i was a few centimeters. As the camera parameters were constant across the observations, the blur variation was only due to camera rotation.

We observe the depth variation is captured satisfactorily (Fig. 10(d)). The errors at the discontinuities are due to convolution approximation and owing to the fact that we did not enforce the segmentation constraint for this experiment.

A real result for the case of pure camera rotation about the vertical axis is shown in Fig. 11. Note the blur variation in the observations (Figs. 11(a, b)) induced by the camera rotation (visible more clearly over the foreground). The distance of the object from the camera was 3–5 cm and the camera was rotated by about 2°. The estimated depth map is shown in Fig. 11(c). Again, except for the errors at the discontinuities, shapes of the objects in the scene are captured fairly well with blur serving as the only depth cue.

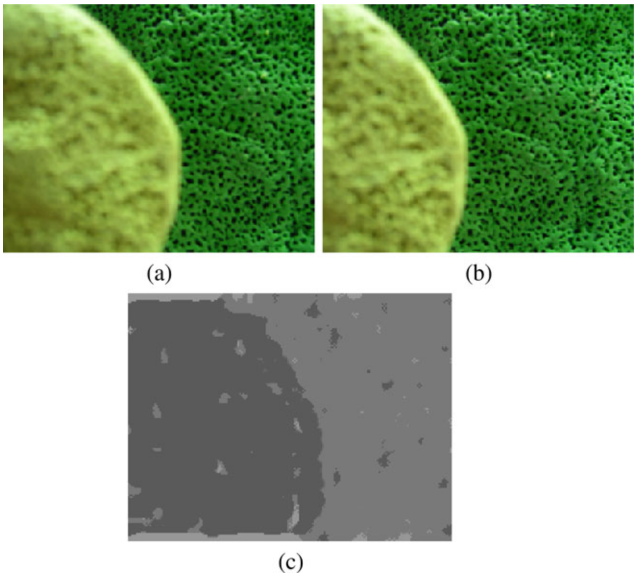

Fig. 11 Real experiment: (**a**, **b**) Observations for a real aperture camera undergoing pure rotation. (**c**) Estimated depth map

Resolving DFD Ambiguity with Stereo Yet another unconventional situation arises when we exploit the ability of our framework to handle camera motion to resolve the ambiguity in depth estimation that can occur with just the defocus cue. In DFD, it is common to capture two images with different camera settings. In such a case, one can face a situation where two different values of Z yield equal values of the relative blur parameter. This, in turn, makes depth estimation ambiguous.

The square of the relative blur parameter σ_r, under two internal camera settings (r_1, v_1, u_1) and (r_2, v_2, u_2) in a DFD scenario can be given as

$$\sigma_2^2 - \sigma_1^2 = \left(\rho r_2 v_2\left(\frac{1}{u_2} - \frac{1}{Z}\right)\right)^2 - \left(\rho r_1 v_1\left(\frac{1}{u_1} - \frac{1}{Z}\right)\right)^2. \quad (42)$$

The plot in Fig. 12 shows an example of the variation in $\sigma_2^2 - \sigma_1^2$ as a function of depth (blue curve), where Z_{c_1} and Z_{c_2} denote abscissa values of the zero crossings. Note that $Z = Z_p$, $\sigma_2^2 - \sigma_1^2$ has a stationary point, which can be easily computed by taking the first derivative of 42 with respect to Z and setting it to 0. Thus, Z_p can be expressed as

$$Z_p = \frac{(v_2^2 - v_1^2)}{(\frac{v_2^2}{u_2} - \frac{v_1^2}{u_1})}. \quad (43)$$

If for all the world objects, the depth Z is restricted such that $Z \geq Z_p$ or $Z \leq Z_p$, then there is a one-one corre-

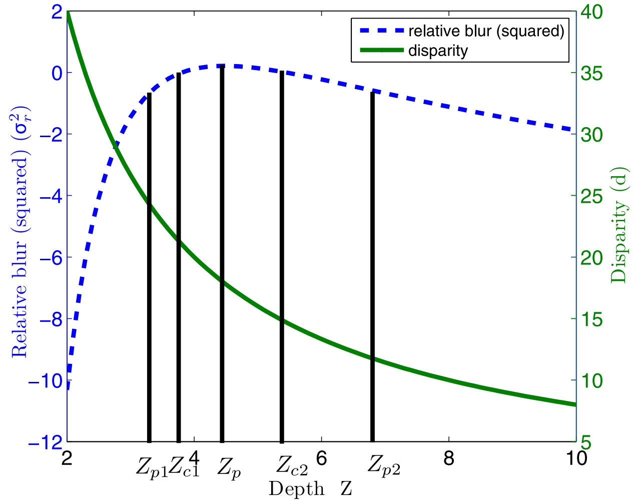

Fig. 12 Ambiguity in DFD

spondence between the depth value Z and the relative blur $\sigma_2^2 - \sigma_1^2$. Otherwise, one can have two depth values (e.g. Z_{p_1} and Z_{p_2}) for which the magnitude and sign of quantity $\sigma_2^2 - \sigma_1^2$ are equal. Thus, there is no way of distinguishing between depths Z_{p_1} and Z_{p_2}.

However, if now the real aperture camera also undergoes motion (importantly, translation), in addition to the parameter variation, while capturing the two images, then one could resolve such an ambiguity. For simplicity, considering pure camera translation t_x in the x-direction and assuming $v_1 = v_i = v$, the pixel motion (or in this case, disparity d_z) is given as

$$d_z = \frac{v t_x}{Z}. \qquad (44)$$

The function d_z is not quadratic with respect to Z and is bijective. It is shown by the monotonically decreasing curve in Fig. 12. Hence, every Z corresponds to a unique disparity d_z. Thus, with the observations captured with a translating real aperture camera, ideally our framework should select the depth which solves both (42) and (44) since only such a depth will minimize the energy with respect to the relative blur $\sigma_2^2 - \sigma_1^2$ as well as pixel disparity d_z. We thus have only one value of depth that satisfies both these constraints no matter what depth range one considers. Thus, our framework, allows one to operate even over those distances where using only the DFD approach can yield incorrect depth estimates.

A synthetic example is shown in Fig. 13 to provide a proof of concept for the case where the DFD ambiguity can be resolved using the stereo cue. We selected camera parameters such that the extremum of $\sigma_r^2(Z)$ lies at a depth of 5.7 cm. Our focused image was a random dot pattern. First, we choose our ground truth depth variation to be a ramp varying between 6.6 and 9.6 cm (Fig. 13(a)). In this range, the variation of $\sigma_r^2(Z)$ is monotonic. Hence, our approach operated for the special case of DFD (with no relative motion between the images), results in correct depth estimates (Fig. 13(b)). We next choose the depth variation as a ramp between 4 to 7 cm (Fig. 13(c)). In this case, using only a DFD pair as observations yields gross errors in the resultant depth map (Fig. 13(d)). Indeed, the depth variation does not resemble the true ramp-like variation at all. Now, for the same depth range shown in Fig. 13(c), we also induced a camera translation in the t_x (horizontal) direction while synthesizing the observations and used our algorithm such that it considers both blur and motion. Inclusion of the stereo cue disambiguates the depth variation and produces a correct output as shown in Fig. 13(e).

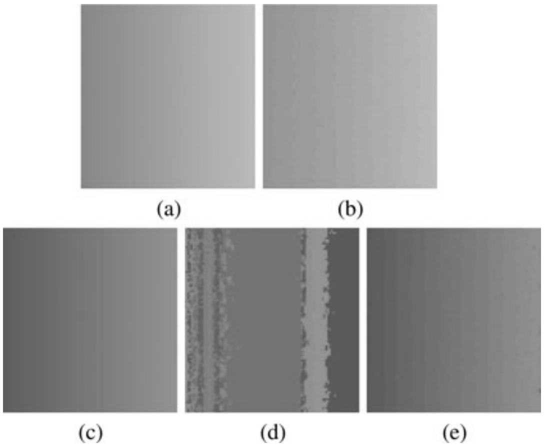

Fig. 13 Synthetic experiment: (**a**, **b**) Ground-truth and estimated depth maps, respectively, using only DFD when the depth values lie in a monotonic portion of the function $\sigma_r^2(Z)$. (**c**, **d**) Ground-truth and estimated depth maps, respectively, using only DFD when the depth values lie in a non-monotonic portion of the function $\sigma_r^2(Z)$. (**e**) Estimated depth map for the depth range in (**c**) when camera is also translated. MAE between (**a**) and (**b**) (with no ambiguity in DFD) is 0.24 cm (depth range is 6.6 cm to 9.6 cm). MAE between (**c**) and (**d**) (where DFD suffers from an inherent ambiguity) is 0.91 cm (depth range is 4 cm to 7 cm). MAE between (**c**) and (**e**) is 0.11 cm (where the stereo cue neutralizes the DFD ambiguity over the same depth range of 4 cm to 7 cm)

A real result for this phenomenon is shown in Fig. 14, where $Z_p = 36.6$ cm. In Fig. 14(c), we see that the depth estimation using only the DFD technique is correct when the scene depth range for the observations (Figs. 14(a, b)) lies in the monotonic part of $\sigma_r^2(Z)$ (more precisely, the depth range is 24–34 cm which is completely below Z_p). The depth variations represented as grey levels specify the depth ordering of the scene objects.

On the contrary, for a scene lying between 27–39 cm and thus containing Z_p (corresponding to the DFD observations in Figs. 14(d, e)), the DFD output (Fig. 14(f)) is quite erroneous. Note that the gray scale representation of some portions of the horse and the complete background are actually ordered in reverse to what they actually should have been.

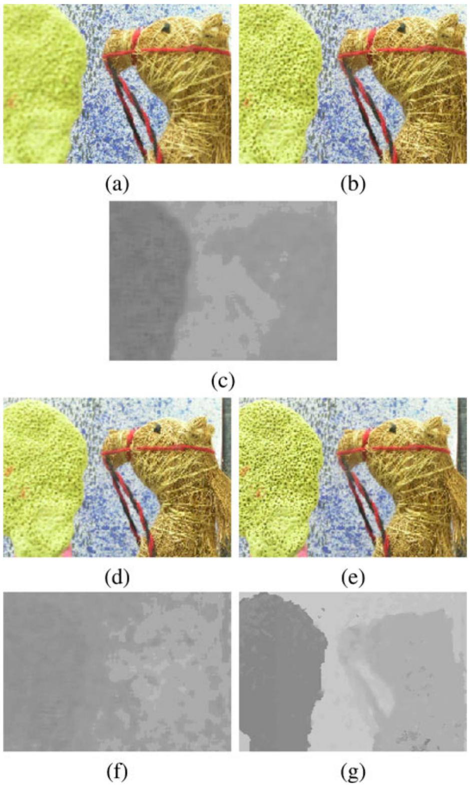

Fig. 14 Real experiment: (**a**, **b**) Observations for DFD with the scene depth range lying in a monotonic portion of the function $\sigma_r^2(Z)$. (**c**) Estimated depth map using only DFD. (**d**, **e**) Observations for DFD when the scene depth range contains Z_p. (**f**) Estimated depth map using only DFD. (**g**) Estimated depth map with both blur and stereo cues. It uses the observation (**d**) and another with the same camera settings as used in (**e**) but also involving a relative translation along the horizontal axis

The shape of the objects also have many distortions due to erroneous depth assignments.

The picture changes (even in the literal sense!) when we use the stereo cue along with the blur cue in this ambiguous depth range. The same scene and the same internal camera settings were used, as in the case for capturing Figs. 14(d, e), except that the second image (not shown here) was captured with the camera translated along the horizontal axis by about 1 cm relative to its position in Fig. 14(e). Note that the DFD depth estimation is disambiguated with the inclusion of stereo (Fig. 14(g)). The depth variations and localization are correctly captured now.

Thus, complimentary to the previous scenario where the blur cue can used to disambiguate the depth estimation under pure camera rotation, in this case, it is the motion cue which is helpful when the defocus cue fails. These special cases thus clearly further highlight the importance of a general framework for the depth estimation task.

5.2 Results for the View Occlusion/Inpainting Problem

In this subsection, we depict results for removing view occlusions/inpainting. We first cover the case of intra-scene occluder, where we remove foreground occluding objects from the scene. We then show experiments for the sensor damage case. In all the experiments, the occluded/missing regions are about 15–30 pixels wide.

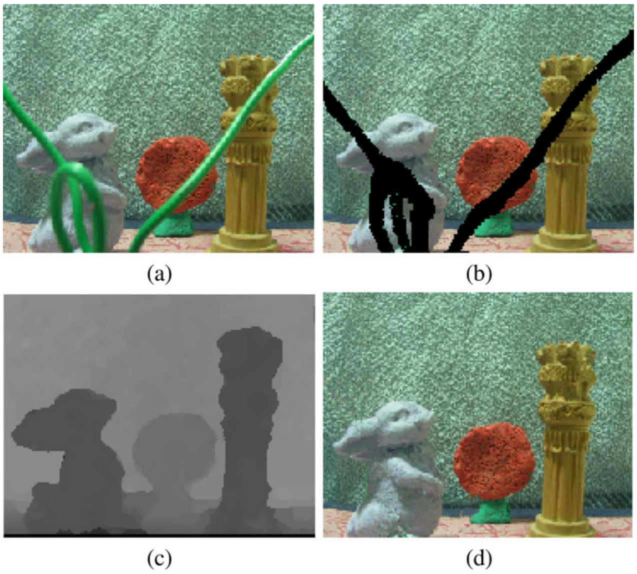

Fig. 15 Real experiment: (**a**) Reference observation where (**b**) the occluder is masked. (**c**) Inpainted depth map. (**d**) Inpainted image

5.2.1 *Results for Intra-Scene Occlusion Removal*

Multi-view Stereo Real Experiments (t_x, r) We next provide real results for our inpainting approach which involves removing an occluder present in the scene. Figures 15 and 16 show two such examples which involve camera translation and variation in aperture. The images with the largest aperture was chosen as reference images in both the examples (Figs. 15(a) and 16(a)). The wires in the foreground are the occluders which we desire to remove from the scene. Note that in the example in Fig. 16, one such occluder completely masks the green stem of the left-most yellow tree in the reference observation Fig. 16(a) and hence in the resultant depth map with the occluders (Fig. 16(d)). This green stem is visible in some of the other observations (e.g. in Fig. 16(b)). The occluders are shown masked in Figs. 15(b) and 16(c). The respective occluder-free depth map outputs are shown in Figs. 15(c) and 16(e). Note that there is barely any trace of the foreground wires. In particular, observe that the discontinuities where the wire crosses the objects do not show any visible signs of distortion. The smoother regions also do not show any artifacts. The resultant inpainted images are shown in Figs. 15(d) and 16(f). Here too, one can notice that texture with its inherent blurring on the rabbit, clay shapes and the background has been inpainted quite coherently with respect to the neighbourhood region. The details on the Ashoka pillar are also retrieved in Fig. 15(d). Interestingly, the green stem of the yellow tree (which is to-

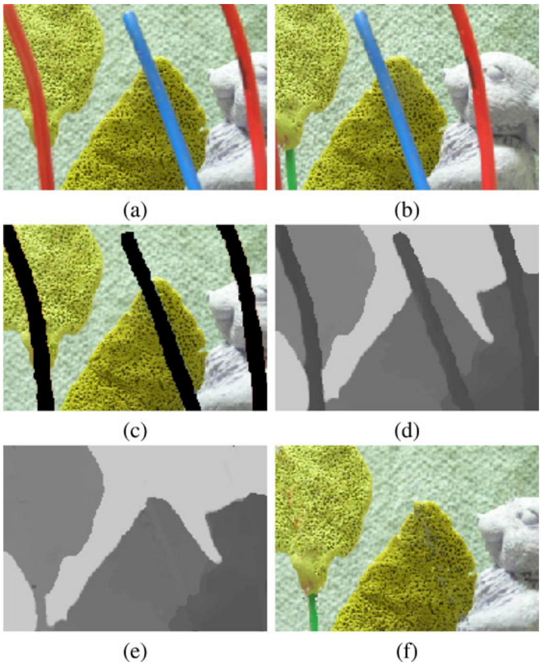

Fig. 16 (Color online) Real experiment: (**a**, **b**) Two out of four observations. (**c**) Occluders masked in the reference observation. Estimated depth map (**d**) with the occluder present and (**e**) with the occluder removed. (**f**) Inpainted image

tally occluded in the reference view) is also recovered completely in both the inpainted depth (Fig. 16(e)) and image (Fig. 16(f)). A small amount of jaggedness in the green stem in the inpainted image can be discounted, considering the fact that it was completely occluded in the reference image. Recovering a completely occluded object is simply impossible for single image-based inpainting methods and thus this result underlines the importance of motion-based occlusion removal.

Multi-view Stereo Real Experiment (t_x, r, u) Next, we show results for an experiment that uses 5 observations two of which are shown in Figs. 17(a) and 17(b). The images are captured from a camera translating on a plane while undergoing a change in aperture and focusing distance. Note that in Fig. 17(a) the camera focuses on blue rabbit while in Fig. 17(b), the focus is on the yellow tree. Figure 17(c) shows the reference observation with the occluders masked. The complete estimated depth map with the occluders is shown in Fig. 17(d). Note how the occluders are indeed shown closest to the camera (the darkest in shade) and are also localized quite well; this is important for their correct warping and subsequent use in inpainting. The inpainted depth map without the occluders is shown in Fig. 17(e) while Fig. 17(f) shows the inpainted image. Again, we observe that the occluders have been successfully removed. The discontinuities and the depth values are correctly inpainted in the depth map. The image inpainting also demonstrates a plausible estimation of texture, details and blur in the occluded regions. Akin to the experiment of Fig. 7, in this experiment too we compared the actual measured depth values to the estimated ones and found that they agree quite closely (see Table 3).

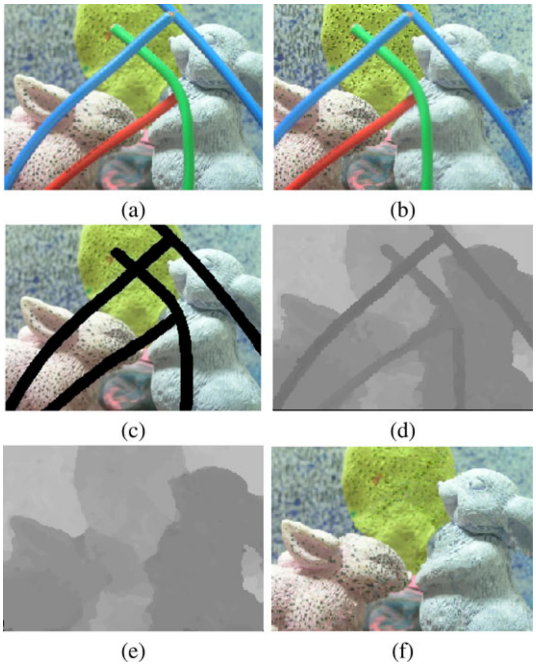

Fig. 17 (Color online) Real experiment: (**a**, **b**) Two out of five observations. (**c**) Occluders masked in the reference observation. Estimated depth map (**a**) with the occluder present and (**b**) with the occluder removed. (**f**) Inpainted image

Table 3 Comparison of the estimated and ground-truth distances for the scene shown in Fig. 17

Objects/regions	d_m (cm)	d_e (cm)
Leftmost blue occluder	23	23.7
Blue rabbit (central region)	27	27.2
Pink rabbit (face region)	29	29.6
Yellow tree (central region)	32	31.8
Background (central region)	38	37.4

Multi-view Stereo Real Experiment (t_x, R_x, r) Our next inpainting experiment uses a camera undergoing translation, rotation and aperture variations. Two of the four images used in the experiment are shown in Figs. 18(a, b). The reference image with the masked occluders are depicted in Fig. 18(c). Figures 18(d, e) respectively show the complete depth map (with the occluders) and the inpainted depth map with the occluders successfully inpainted. In this experiment, the actual depth of the objects from the camera was similar to that in that of Fig. 17. Hence, we can deduce, on the basis of result (which uses the same scale factor in gray-scale rep-

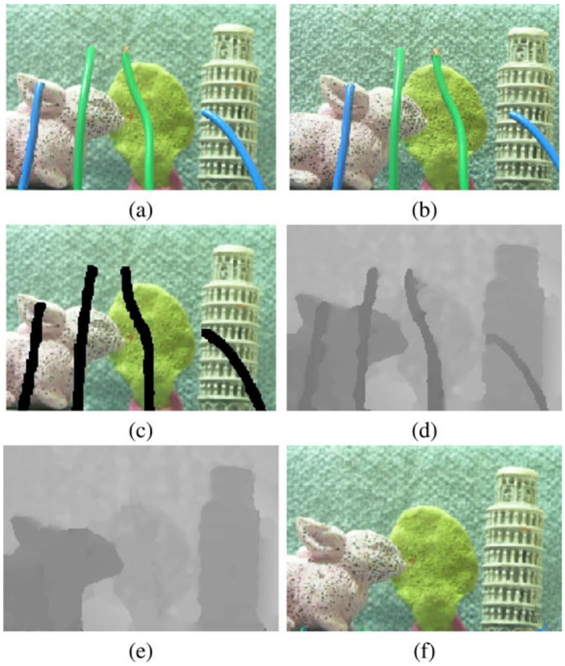

Fig. 18 (Color online) Real experiment: (**a**, **b**) Two out of four observations. (**c**) Occluders masked in the reference observation. Estimated depth map (**d**) with the occluder present and (**e**) with the occluder removed. (**f**) Inpainted image

resentation as in Fig. 17), that the depth estimation is fairly accurate. The inpainted image is shown in Fig. 18(f). Note especially the salient aspects in the inpainted regions such as the structural details in the Pisa tower. Also, texture on the yellow tree, pink bunny and the background are all recovered without any visible artifacts.

5.2.2 Results for Inpainting Lens/Sensor Damage

We now show results for inpainting the sensor damage where we scratch the observations to emulate sensor damage. For simplicity, we consider the case of a single damaged camera used for capturing multiple images. However, as discussed earlier, the same framework can be used to handle the multiple damaged camera case too. Since, we have the actual observations without scratches, to demonstrate the efficacy of the depth inpainting in the real examples, we compare the depth estimates with the scratched observations with those estimated using the undamaged observations.

Multi-view Stereo Synthetic Experiment (t_x, R_x, r) In Fig. 19 we show a synthetic example involving translation, rotation and aperture variation. Note that the scratched observations Figs. 19(a, b) are also space-variantly blurred. The observations were synthesized as in the earlier example of Fig. 5. We note that the inpainted depth map Fig. 19(d) is quite close to the ground truth of Fig. 19(c). The effect of sensor damage is barely visible. Similarly, the inpainted image Fig. 19(f) is visually identical to the ground-truth blurred image Fig. 19(e). Note that the filled-in regions in the image are coherently defocused as their neighbourhood even though the observations have varying amounts of blur.

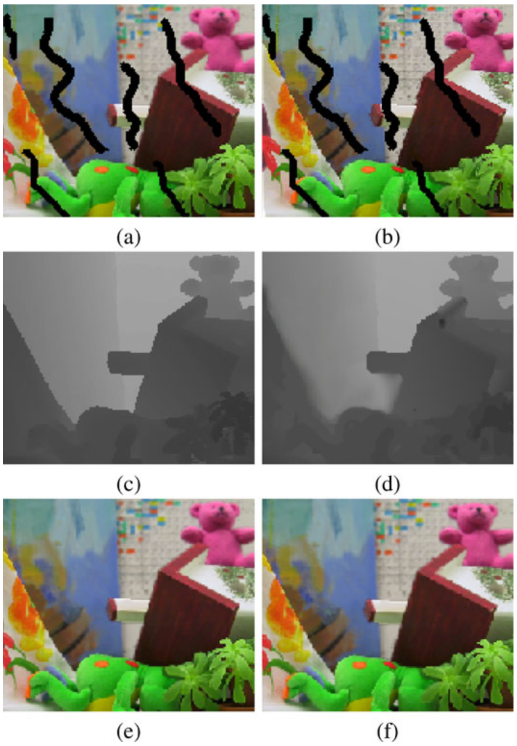

Fig. 19 Synthetic experiment: (**a**, **b**) Two of the four observations used in the experiment. (**c**, **d**) Ground-truth and estimated depth, respectively. (**e**, **f**) Original image and inpainted image, respectively. MAE for depth estimation is 0.33 cm over the depth range 5–16 cm. MAE for the inpainted image at the masked pixels is 6.29 intensity levels

Multi-view Stereo Real Experiment (t_x, R_x, r) We next provide a real result for inpainting involving translation, rotation and aperture variation. Figures 20(a, b) display two of the four damaged observations. Observe the difference in blurring, especially on the background and the farther objects such as the green tree and Pisa tower. Our depth map output is shown in Fig. 20(d). Note, again, that the depth map is cleanly inpainted and in addition, shows good discontinuity preservation and a plausible depth variation. Since we also have the undamaged observations we can compare the depth map in Fig. 20(d) with that estimated using the undamaged images (Fig. 20(c)). There is some difference at the places where the damage crosses the discontinuities, but that is hardly noticeable and there is no unnaturalness in the inpainted depth map. Thus, the accuracy of the depth inpainting can be appreciated. In comparing the original unscratched image (Fig. 20(e)) and the inpainted image (Fig. 20(f)), we find that the scratches are indeed filled without hampering the intricate intensity variations on the ob-

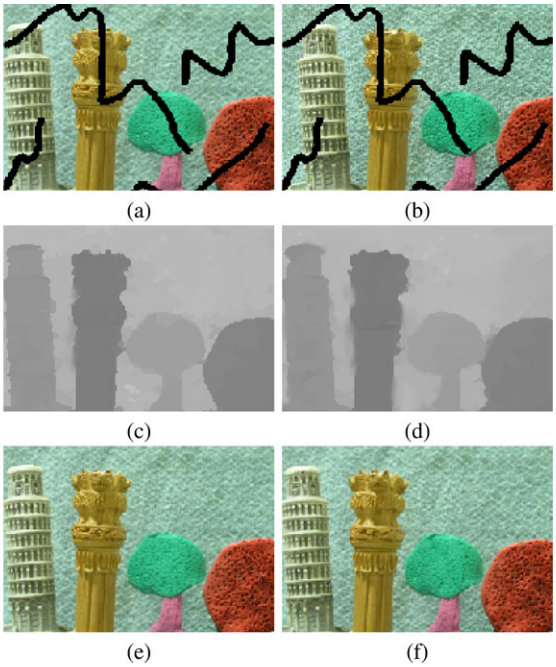

Fig. 20 Real experiment: (**a**, **b**) Two observations out of four with translation, rotation and variations in aperture. Estimated depth map from (**c**) non-damaged observations and (**d**) damaged observations. (**e**) Original unscratched image. (**f**) Inpainted image. MAE between the depth estimates in (**c**) and (**d**) is 0.5 cm overall and 0.67 cm over the scratched regions, where the depth range was 24–40 cm. MAE between the original image and the inpainted image over the scratched regions is 9.57 intensity levels

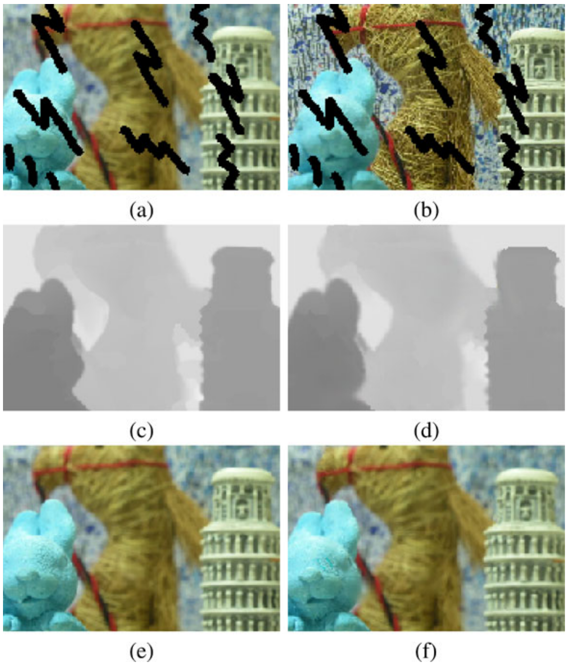

Fig. 21 Real experiment: (**a**, **b**) Two out of four damaged observations involving translation and variations in aperture and focusing distance. Estimated depth map from (**c**) undamaged observations and (**d**) damaged observations. (**e**) Original undamaged image. (**f**) Inpainted image. MAE between the depth estimates in (**c**) and (**d**) is 0.43 cm overall and 0.52 cm over the scratched regions, where the depth range was 25–45 cm. MAE between the original image and the inpainted image over the scratched regions is 6.72 intensity levels

jects (e.g., the pattern on the Pisa tower and the Ashoka pillar, and the dotted texture on the trees and the background).

Multi-view Stereo Real Experiment (t_x, r, u) Finally, we show a real result with four images involving camera translation, and variation in aperture and focusing distance. One can note the near-far focusing effects in the three observations which are shown in Figs. 21(a, b). Figure 21(d) shows the estimated depth map having well-defined shapes, even with a large amount of defocus in the observations for the foreground and background objects. The depth variation is also captured with good fidelity, not only across close objects (such as the bunny and the Pisa tower model), but also within each object (e.g. the ears of the bunny, which are slightly behind the face). In addition, the depth estimation in the damaged regions is also quite flawless. Again, we notice that the depth map estimated with undamaged observations (Fig. 21(c)) has little difference with that estimated using damaged observations, thus highlighting the correctness of depth inpainting. With largely different blur in the observations, the image inpainting result (Fig. 21(f)) also emphasizes the ability of our approach to coherently inpaint the damaged image regions, maintaining visually correct defocusing. The inpainted image also compares well with the actual undamaged observation (Fig. 21(e)).

6 Discussions

This paper describes one of the first comprehensive efforts of its kind towards generalizing the depth estimation problem. To our knowledge, there is no work in literature which considers the level of generalization considered in this paper. Hence, we are unable to provide comparisons with existing approaches as they are quite restrictive. Nevertheless, we believe that our results are qualitatively comparable to what is typically achieved in state-of-the-art depth estimation approaches. The primary goal of the paper was to develop a general framework for depth estimation and address the view occlusion problem within such a framework. Although, we have discussed, at length, various interesting situations that arise in the process of generalizing the task of depth estimation when using a real-aperture camera, we would like to highlight some issues that hold potential for future research in this direction.

- As mentioned in Sect. 2.2, ideally, the defocus blur kernel need not be constrained to have a parametric form. In the worst case, it may not even be circularly symmetric and can have an arbitrary shape. However, unlike parametric kernels, it is difficult to analytically relate such an arbitrary kernel with depth.

One way to map an arbitrary blur kernel to depth is to use a priori calibration to relate estimated blur kernels with their respective depth values and then pick the depth label corresponding to the kernel that minimizes the cost (Ens and Lawrence 1993). For a thin lens model, the kernels at different depths can be related through scaling of the PSF (Sorel 2007). In this case, the absolute depth at any point can be computed if one a priori knows (through calibration) the kernel-depth correspondence at some reference point.

- We discussed some bounds and gave some insights for the view occlusion problem which can provide the user with helpful guidelines. However, a detailed analysis (e.g. Chan and Kang 2006) of such a motion-based inpainting problem may provide a deeper understanding.
- The occlusions in our framework is only related to camera motion. There exists some works which consider *partial occlusion* due to defocus. However, for noticeable partial occlusion, the foreground objects should be extremely close to the camera and the background should be considerably farther. We note that over the (more natural) depth ranges that we operate, we have negligible partial occlusion and it is safe to ignore it. But it would play an important role if, in some special situations, one needs to considers more extreme depth ranges.
- We relate the relative blur between the images to the scene depth. This is in fact an approximation to the actual space-variant imaging model (as discussed in Sect. 3.1). An efficient approach which considers the true space-variant nature of the problem can be explored.
- Generalizing depth estimation further should consider even aspects such as different degrees of exposure and motion blur.

7 Conclusions

We proposed a depth estimation framework that intertwines the camera parameters, motion and blur cues. Our framework handles various camera effects such as parallax, zooming, occlusions and focusing variations. We also address the problem of removing user-defined view-occlusions from the depth map as well as the image. Our occlusion removal (inpainting) approach exploits depth cues resulting from motion and defocus blur and is hence closely coupled to the depth estimation framework. We formulated our depth estimation and occlusion removal approaches in an efficient BP framework with visibility handling and segmentation constraint. Our results sufficiently validate our claims, both qualitatively as well as quantitatively.

References

Ahuja, N., & Abbot, A. L. (1993). Active stereo: Integrating disparity, vergence, focus, aperture and calibration for surface estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *15*(10), 1007–1029.

Bhavsar, A. V., & Rajagopalan, A. N. (2008). Range map with missing data—joint resolution enhancement and inpainting. In *Indian conference on computer vision, graphics and image processing (ICVGIP 2008)* (pp. 359–365).

Bhavsar, A. V., & Rajagopalan, A. N. (2009). Depth estimation with a practical camera. In *British machine vision conference (BMVC 2009)*.

Bhavsar, A. V., & Rajagopalan, A. N. (2010). Depth estimation and inpainting with an unconstrained camera. In *British machine vision conference (BMVC 2010)*.

Chan, T., & Kang, S. (2006). Error analysis for image inpainting. *Journal of Mathematical Imaging and Vision*, *26*(1), 85–103.

Criminisi, A., Perez, P., & Toyama, K. (2003). Object removal by exemplar-based inpainting. In *Proc. IEEE computer society conference on computer vision and pattern recognition (CVPR 2003)* (pp. 721–728).

Deschenes, F., Ziou, D., & Fuchs, P. (2004). A unified approach for a simultaneous and cooperative estimation of defocus blur and spatial shifts. *Image and Vision Computing*, *22*(1), 35–57.

Dorin, C., & Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(5), 603–619.

Drouin, M., Trudeau, M., & Roy, S. (2005). Geo-consistency for wide multi-camera stereo. In *IEEE conference on computer vision and pattern recognition (CVPR 2005)* (Vol. 1, pp. 351–358).

Duo, Q., & Favaro, P. (2008). Off-axis aperture camera: 3d shape reconstruction and image restoration. In *IEEE conference on computer vision and pattern recognition (CVPR 2008)* (pp. 1–7).

Ens, J., & Lawrence, P. (1993). An investigation of methods for determining depth from focus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *15*(2), 97–108.

Favaro, P., & Soatto, S. (2005). A geometric approach to shape from defocus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *27*(3), 406–417.

Favaro, P., Soatto, S., Burger, M., & Osher, S. (2008). Shape from defocus via diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*(3), 518–531.

Felzenszwalb, P., & Huttenlocher, D. (2004). Efficient belief propagation for early vision. In *IEEE computer society conference on computer vision and pattern recognition (CVPR 2004)* (Vol. 1, pp. 261–268).

Fischler, M. A., & Bolles, R. C. (1981). Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, *24*, 381–395.

Frese, C., & Gheta, I. (2006). Robust depth estimation by fusion of stereo and focus series acquired with a camera array. In *IEEE international conference on multisensor fusion and integration for intelligent systems* (pp. 243–248).

Gu, J., Ramamoorthi, R., Belhumeur, P., & Nayar, S. (2009). Removing image artifacts due to dirty camera lenses and thin occluders. In *SIGGRAPH Asia '09: ACM SIGGRAPH Asia 2009 papers* (pp. 1–10).

Hartley, R., & Zisserman, A. (2003). *Multiple view geometry in computer vision*. Cambridge: Cambridge University Press.

Kang, S., Szeliski, R., & Chai, J. (2001). *Handling occlusions in dense multi-view stereo* (Tech. rep.). Microsoft Technical Report MSR-TR-2001-80.

Kim, J., & Sikora, T. (2007). Confocal disparity estimation and recovery of pinhole image for real aperture stereo camera systems.

In *IEEE international conference image processing (ICIP 2007)* (Vol. V, pp. 229–232).

Krotkov, E., & Bajcsy, R. (1993). Active vision for reliable ranging: Cooperating focus, stereo and vergence. *International Journal of Computer Vision*, *11*(2), 187–203.

Li, S. (1995). *Markov random field modeling in computer vision*. Tokyo: Springer.

Myles, Z., & Lobo, N. V. (1998). Recovering affine motion and defocus blur simultaneously. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *20*(6), 652–658.

Nayar, S. K., & Nakagawa, Y. (1994). Shape from focus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *16*(8), 824–831.

Paramanand, C., & Rajagopalan, A. N. (2010). Unscented transformation for depth from motion-blur in videos. In *IEEE workshop on three dimensional information extraction for video analysis and mining*.

Pentland, A. (1987). A new sense for depth of field. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *9*(4), 523–531.

Rajagopalan, A. N., & Chaudhuri, S. (1999). *Depth from defocus: A real aperture imaging approach*. New York: Springer.

Rajagopalan, A. N., Chaudhuri, S., & Mudenagudi, U. (2004). Depth estimation and image restoration using defocused stereo pairs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(11), 1521–1525.

Sahay, R., & Rajagopalan, A. N. (2008). A model-based approach to shape from focus. In *Proceedings of the 3rd international conference on computer vision theory and applications (VISAPP 2008)* (pp. 243–250).

Sahay, R., & Rajagopalan, A. N. (2009a). Inpainting in shape from focus: Taking a cue from motion parallax. In *British machine vision conference (BMVC 2009)*.

Sahay, R., & Rajagopalan, A. N. (2009b). Real aperture axial stereo: Solving for correspondences in blur. In *DAGM-Symposium 2009* (pp. 362–371).

Sahay, R., & Rajagopalan, A. N. (2010). Joint image and depth completion in shape-from-focus: Taking a cue from parallax. *Journal of Optical Society of America. A*, *27*(5), 1203–1213.

Scharstein, D., & Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, *47*(1), 7–42.

Seitz, S., & Baker, S. (2009). Filter flow. In *Proc. international conference on computer vision (ICCV 2009)*.

Seitz, S., Curless, B., Diebel, J., Scharstein, D., & Szeliski, R. (2006). A comparison and evaluation of multi-view stereo reconstruction algorithms. In *IEEE conference on computer vision and pattern recognition (CVPR 2006)* (Vol. 1, pp. 519–526).

Sorel, M. (2007). *Multichannel blind restoration of images with space-variant degradations*. PhD Thesis, Charles University.

Strecha, C., Van Gool, L., & Fransens, R. (2004). Wide-baseline stereo from multiple views: a probabilistic account. In *IEEE conference on computer vision and pattern recognition (CVPR 2004)* (Vol. 2, pp. 552–559).

Subbarao, M., Yuan, T., & Tyan, J. (1997). Integration of defocus and focus analysis with stereo for 3d shape recovery. *Proceedings of SPIE*, *3204*, 11–23.

Wang, L., Jin, H., Yang, R., & Gong, M. (2008). Stereoscopic inpainting: Joint color and depth completion from stereo images. In *Proc. IEEE computer society conference on computer vision and pattern recognition (CVPR 2008)* (pp. 1–8).

Watanabe, M., & Nayar, S. (1995). Telecentric optics for computational vision. In *European conference on computer vision (ECCV 1995)* (pp. 439–451).

Wohler, C., d'Angelo, P., Kruger, L., Kuhl, A., & Grob, H. M. (2009). Monocular 3d scene reconstruction at absolute scale. *ISPRS Journal of Photogrammetry and Remote Sensing*, *64*(6), 529–540.

Wrotniak, J. A. (2003). Depth of field tables for olympus c-30x0z, c-40x0z, and c-5050z cameras. http://www.wrotniak.net/photo/tech/dof-c5050.html.

Wrotniak, J. A. (2006). Depth of field tables for the olympus c-5050z, c-4040z, and c-3040z digital cameras. http://www.digitaldiver.net/lib_docs/oly_dof.html.

Yang, Q., Wang, L., Yang, R., Stewenius, H., & Nister, D. (2009). Stereo matching with color-weighted correlation, heiarchical belief propagation, and occlusion handling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *31*(3), 492–504.

Zhou, C., & Lin, S. (2007). Removal of image artifacts due to sensor dust. In *Proc. IEEE computer society conference on computer vision and pattern recognition (CVPR 2007)* (pp. 1–8).